Neelabh Kashyap

Novel Resource-Efficient Algorithms for State Estimation in the Future Grid

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo March 16, 2012

Thesis supervisor:

D.Sc. (Tech.) Stefan Werner

Thesis instructor:

M.Sc. (Tech.) Taneli Riihonen

Aalto University
School of Electrical Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Neelabh Kashyap

Title: Novel Resource-Efficient Algorithms for State Estimation in the Future Grid

Date: March 16, 2012 Language: English Number of pages:7+66

Department of Signal Processing and Acoustics

Professorship: Signal Processing Code: S-88

Supervisor: D.Sc. (Tech.) Stefan Werner

Instructor: M.Sc. (Tech.) Taneli Riihonen

The purpose of this study is to examine the challenges and opportunities presented by the evolution of the legacy grid into a smart grid within a framework relevant to signal processing research. The focus here is on the application of statistical signal processing techniques to design new algorithms for state estimation which is a key function in the supervisory control and planning of power grids. To begin with, two resource-efficient forecasting-aided state estimation algorithms are developed which can combine measurements from both the traditional measurement devices as well as the newer measuring devices known as phasor measurement units and operate on them optimally in order to arrive at an estimate of the system state. The ability of the algorithms to track the evolution of the state vector in time is verified using a computer simulation and their statistical performance with respect to the root mean-squared error is studied. Since concentrating the state estimation function at a single point to monitor a large interconnection leads to huge communication and computational overhead, a more feasible approach is to distribute the state estimation function throughout the interconnection. An *on-demand*, distributed estimation scheme which features event-triggered communication is developed herein to reduce the communication and computational overhead associated with distributed estimation in large systems. This technique is derived from a cutting-edge signal processing paradigm known as set-membership adaptive filtering. The performance of the new algorithm is studied using computer simulations and comparisons are drawn to existing adaptive filtering methods like the recursive least-squares method.

Keywords: Smart grid, State estimation, Power systems, Kalman filtering, Set-membership adaptive filtering

"We can't solve problems by using the same kind of thinking we used when we created them."

– Albert Einstein

Acknowledgements

This thesis, like every difficult endeavour one attempts in life, could not have been completed to fruition without the guidance and support of several individuals who in one way or another made invaluable contributions to the preparation and completion of this study.

I cannot express in words my gratitude to Dr. Stefan Werner, from whom I have learnt so much, for giving me the opportunity to work with him. One simply could not wish for a better or friendlier supervisor and I cannot but look forward to many more years of working with him. I thank Taneli Riihonen, my instructor, for carefully and painstakingly reading every draft of this thesis and for teaching me the value of paying attention to detail. With his diligence, dedication, perseverance and, not to mention, outstanding intelligence, he serves as a role model for anyone wanting to succeed in academia. I have learnt so much from him and because of him!

Furthermore, I wish to extend my sincerest thanks to Prof. Yih-Fang Huang from the University of Notre Dame, Indiana, USA, for being a constant source of ideas and encouragement. It was upon his pioneering work in signal processing that much of the work presented herein is based. I also thank Jing Huang from the University of Notre Dame for helping me get started on the work presented here and for sending me his MATLAB simulation code which helped me greatly in designing functions that are still the mainstay of every simulation I conduct.

I express my deepest gratitude to Pramod J. Mathecken, my close friend, mentor and colleague for guiding me through every single step of my master's studies. Without his support and encouragement, it would have been exceedingly difficult to get to where I am today.

Finally, I thank my family, especially my mother, for being there for me through my darkest and finest hours, and for helping me stay focussed and motivated in everything I did.

Neelabh Kashyap
March 16, 2012

Contents

List of Figures

Chapter 1

Introduction

Power generating stations, end-users (loads) and the network of transmission lines used to transmit power from the generators to the loads collectively constitute a power system also known as the power grid, or simply, "the grid". The power grid is a vast, complex system which needs to be constantly monitored in order to maintain its operations in a normal, secure state, i.e. the net demand for electrical energy must be met by injecting the required amount into the transmission and distribution network. In order to control this flow of electricity, the authority in charge of delivering it must be aware of the present "state" of the power system.

The minimal set of network parameters which represent the state of a power system is the collection of the voltage magnitudes and phase angles at every node in the power network. In the terminology of the power grid, these nodes are called busbars or buses. The process of acquiring measurements from all parts of the grid and extracting the system state from these measurements is called state estimation [1]. The measurements acquired commonly consist of transmission line power flows, generator outputs, loads, circuit breaker positions, transformer tap positions and capacitor bank values.

The state estimation function is one of the key components in energy management systems (EMS) and is used to analyse contingencies, make decisions on real-time market pricing [2], and determine any corrective actions which may be required. While state estimation in power systems has been studied extensively since its introduction in the late 1960s, the traditional state estimation paradigm will need to evolve to meet the demands of the modern energy ecosystem like environmental compliance, market deregulation and improved reliability and security.

The general infrastructure of the power grid, based in part on Nikola Tesla's design published in 1888, has remained largely unchanged for over a century. However, certain assumptions which were hitherto valid, like centralized unidirectional electric power transmission and distribution, are no longer true. Moreover, advances in information and communication technology have not permeated the energy infrastructure adequately. This has given impetus for a new energy infrastructure for the future known as the smart grid [3].

Figure 1.1 offers a general depiction of the three main layers of the electric power grid hierarchy. Based on the multi-level state estimation paradigm in [4], state es-

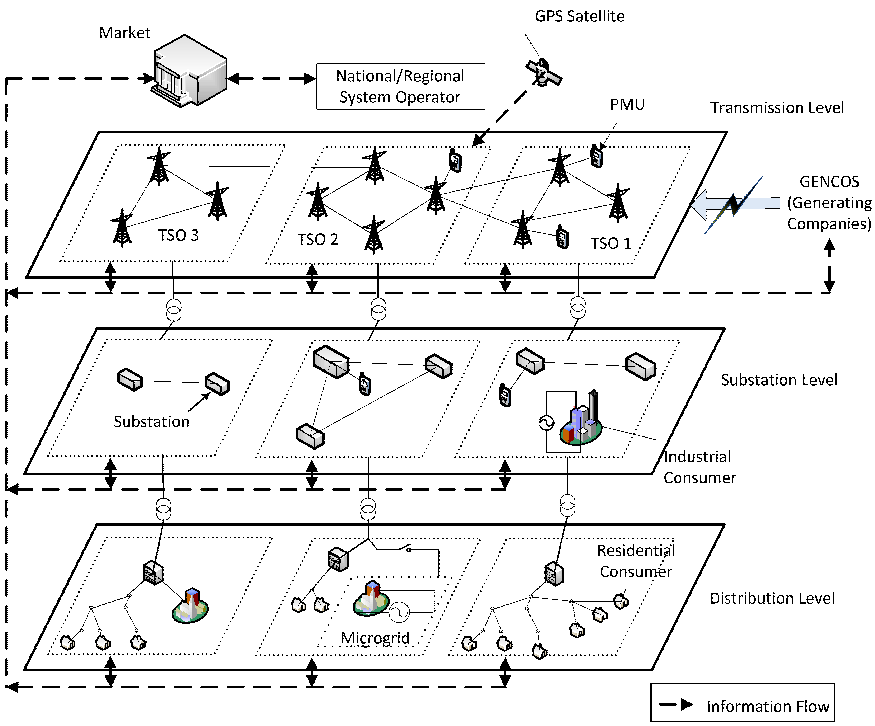


Figure 1.1: Electricity ecosystem of the future grid featuring the various players and levels of interaction.

timation would be carried out at different levels, namely the transmission system operator (TSO) level, the local level or substation level and, increasingly, at the distribution level. The TSO is an entity that operates the transmission grid in order to supply electricity from the generating companies (GENCOs) to the utility companies and then to the consumer. Substations are a vital link between the transmission and distribution networks and are responsible for converting voltage and current levels. The trend of deregulation of vertically integrated utilities, particularly in the United States, would mean that market forces would play an increasing role in the future grid.

1.1 Motivation

There are several aspects of the smart grid that affect research on state estimation in power systems. Two major changes, however, directly motivate the work herein:

Emergence of new measurement technologies [5]: The recent emergence of smart grid technologies like phasor measurement units (PMUs), developed in the early 1980s, has offered hopes for near-real-time monitoring of the power grid. Typically, a PMU takes 30 measurements per second, thereby offering a much more timely view of the power system dynamics than conventional measurements. More importantly, all PMU measurements are synchronized, as they are time-stamped by the global positioning system's (GPS) universal clock. However, PMUs with their higher measurement frequency put unnecessary strain on the communications infrastructure of the smart grid. This drives the need for resource-efficient, event-triggered state estimation solutions that employ data-dependent selective sensing and estimation.

Departure from a vertically integrated utility structure [6]: Until recently, electricity utilities were either private or government monopolies charged with producing, transmitting and distributing power to industries and the general populace. Today, however, transmission system is operated separately by an independent entity (independent system operator (ISO) in North America and transmission system operator (TSO) in Europe) which ensures fair and free access to the electricity supply network. Such regulations may require utility companies to share more information and monitor the grid over a very large geographical area.

In summary, larger and more complicated systems, coupled with the enormous amount of measurement data generated by new measurement technologies, call for a rethinking of the traditional state estimation paradigm in power systems and the development of new algorithms which face up to the challenges posed by power grids of the future.

1.2 Scope of the Research Problem

This work is concerned with the state estimation problems associated with the control of a small portion of the overall power system that is typically maintained by an independent system operator (in America) or a transmission system operator (in Europe). This is termed the own system and is connected to the interconnected system by tie lines. All the algorithms studied herein assume that the controlling authority has access to measurements on all nodes in its own system. The meter readings are not perfect and small random errors are always present. Furthermore, it is assumed that any bad data have been removed and network modelling and parameter errors are not present. The objective here is to develop novel state estimation algorithms which can function within the existing SE framework while, at the same time, meeting the challenges posed by the emergence of the smart grid.

To begin with, resource-efficient dynamic state estimation algorithms must be developed which can combine measurements from both the traditional measurement devices as well as the newer PMUs and operate on them optimally in order to arrive at an estimate of the system state. Furthermore, concentrating the state estimation function at a single point to monitor a large interconnection leads to huge communication and computational overhead. A more feasible approach would be to distribute the state estimation function throughout the interconnection. An *on-demand* estimation scheme featuring event-triggered communication is necessary to reduce the communication and computational overhead associated with distributed estimation in large systems.

1.3 Contributions

The contributions of this thesis are relevant to the fields of statistical signal processing as well as smart grid research. A current research topic in signal processing is the application of signal processing techniques to power systems, and opportunities for achieving this were examined in collaboration with members of the ASPECT research group at the University of Notre Dame, Indiana, USA. An outcome of the extensive research work and literature survey performed resulted in an article [7] to be published in a special issue of the IEEE Signal Processing Magazine. The literature survey presented here, while not exhaustive, examines the evolution of state estimation in power systems from a signal processing standpoint, and, where possible, the applications of signal processing techniques are highlighted.

Specifically, the contributions of this thesis are twofold:

Forecasting-aided state estimation with PMUs: A forecasting-aided state estimation scheme is developed to combine conventional measurements with PMU measurements. However, this scheme has certain drawbacks like high computational complexity and the possibility of ill-conditioned matrices. To overcome these drawbacks, a new, reduced complexity, reduced-order algorithm is proposed for dynamic state estimation in power systems. Next, the ability of the two estimation techniques to track the evolution of the state vector in time is verified using computer simulations. Furthermore, the statistical performance of both algorithms with respect to the root mean-squared error is studied and their performances are compared.

Event-triggered multi-area state estimation: A new selective-update nonlinear adaptive filtering algorithm is derived here, upon which a novel, resource-efficient, recursive algorithm for multi-area state estimation in large interconnections is based. This algorithm was presented at 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP) in San Juan, Puerto Rico [8]. This approach not only reduces the amount of communication between the different regions that make up the large interconnection but also computational complexity. The performance of the

estimation algorithm is studied using computer simulations and comparisons are drawn to recursive least-squares adaptive filtering.

1.4 Organization

The remainder of this work is organized as follows. In Chapter 2 we conduct a of the evolution of various state estimation paradigms and algorithms existing in the literature. In Chapter 3, we explore the impact of PMUs on power system state estimation. In Chapter 4, two new algorithms are introduced. Results originating from computer simulations of these algorithms are also discussed here. In Chapter 5, we introduce a new, event-triggered distributed state estimation algorithm and discuss the results of computer simulations of this algorithm. Conclusions drawn from the research that has led to the publication of this work are discussed briefly in Chapter 6.

Chapter 2

State Estimation and Its Evolution

The study of power generation and transmission is one of the oldest branches of engineering. It is a vast and diverse field with over two hundred years of research and thousands of books dedicated to the field. Therefore, in this chapter, only those aspects of power systems are explained which are crucial in understanding topics covered in later chapters. However, a detailed treatment of the basics of power systems may be found in [9, 10].

In the following, after describing some basics, we survey the development and evolution of various state estimation techniques which work exclusively on conventional power-flow, power-injection and busbar voltage-magnitude measurements.

2.1 Measurement Acquisition

The EMS/SCADA (Energy Management System/Supervisory Control and Data Acquisition) system is a set of computational tools used to monitor, control, and optimize the performance of a power system. Initially, power systems were overseen only by supervisory control systems. These were control systems which monitored the status of circuit breakers at substations along with generator outputs and the overall system frequency [1]. Later, supervisory control systems were enhanced by adding an interconnection-wide real-time data acquisition function giving rise to the first SCADA system. Coupled with the planning and analysis functions, this is called the SCADA/EMS or EMS/SCADA system.

During normal operation, the power system is either in a *secure* or *insecure* state. The security of a power system is defined as "an instantaneous time-varying condition reflecting the robustness of the system relative to imminent disturbances; the complement of the risk of disruption of unimpaired system performance" [11]. In other words, the power system is said to be in a *secure state* if disturbances within the power grid do not impair system performance. State estimation is a vital component of the EMS/SCADA and it is used to analyse the security of the power system and take corrective or preventive action when necessary.

The relationship between state estimation and the EMS/SCADA system is shown in Fig. 2.1. The data acquisition system obtains real-time measurement from devices like remote terminal units (RTU) and, more recently, phasor data concentrators

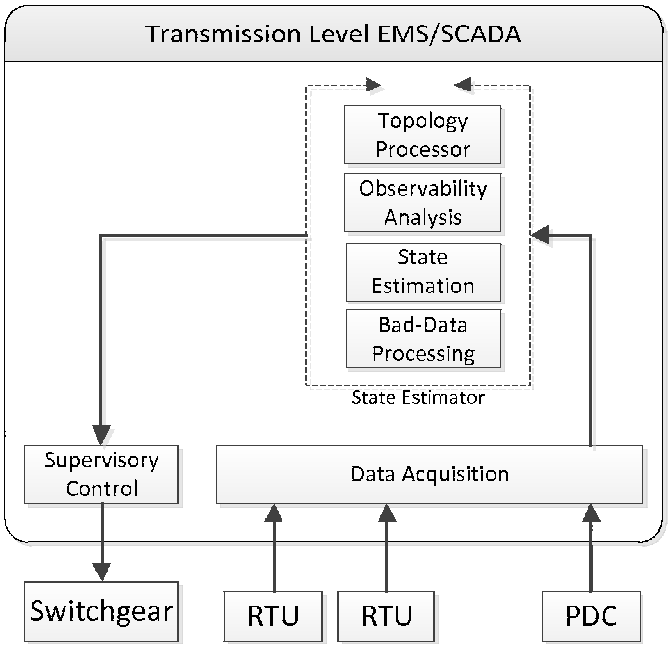


Figure 2.1: Relationship between the different elements that collectively constitute the EMS/SCADA [7].

(PDC) scattered throughout the system. The state estimator then calculates the system state and provides the necessary information to the supervisory control system which then takes action by sending control signals to the switchgear (circuit breakers).

The conventional state estimator built into the EMS consists of four main processes [1] as shown in Fig. 2.1:

Topology Processing which is a process that tracks the network topology and maintains a real-time database of the network model. This is done by analysing the position of circuit breakers and other switchgear in the substations.

Observability Analysis which is a process that is run to ensure the measurement set is sufficient to perform state estimation. If it is found that the measurement set is incomplete, *pseudo-measurements* are added to the measurement sets. Pseudo-measurements are measurements generated from short-term load forecasts, historical records or similar approximation methods.

State Estimator which functions by operating on the measurement set and, using some kind of estimation algorithm, arrive at an estimate of the system state.

Bad-Data Processing which is a process that identifies any gross errors in the measurement set and eliminates bad measurements.

2.2 Structure of the Measurement Set

Commonly used measurements include line power flows, bus power injections, bus voltage magnitudes and line current flows. The system state $\mathbf{x}$ is related to the measurements by a set of L nonlinear expressions known as the measurement function $\mathbf{h}(\mathbf{x})$.

Now, the state vector $\mathbf{x} \in \mathbb{R}^{(2N-1)\times 1}$ consists of the voltage magnitudes and phase angles on each bus in the system, i.e. N bus voltage magnitudes and $N-1$ phase angles. The phase angle of one bus, known as the reference bus or slack bus, is assumed to be known and set to an arbitrary value, such as 0. The state vector will have the following form assuming the first bus is chosen as the reference bus:

$$\mathbf{x} = [\theta_2, \theta_3, \ldots, \theta_N, |V_1|, \ldots, |V_N|]^{\mathrm{T}}, \tag{2.1}$$

where θ_n denote the phase angles and $|V_n|$ the magnitudes of the voltages at the n-th bus.

A transmission line connected between two nodes i and j possesses a conductance g_{ij} which is a function of the length and material of the line and a susceptance b_{ij} which is a function of the material of the line and the frequency of the alternating current flowing through it. Furthermore, the entire transmission line behaves like one of the plates of a capacitor, the other plate being the earth itself. This means that the transmission line behaves as if some electricity is *shunted* to the ground through a capacitor. To make analysis easier, this capacitor is split into two and each capacitor is connected between either end of the transmission line and the ground. Now each capacitor is known as a shunt capacitor. When several lines are connected to a node i, their capacitances are lumped together and treated as one single shunt capacitor connected between node i and the ground. This shunt capacitor has an admittance $y_{Si} = g_{Si} + jb_{Si}$. Figure 2.2 shows the representation of a transmission line as a circuit known as a π-*model* with discrete elements.

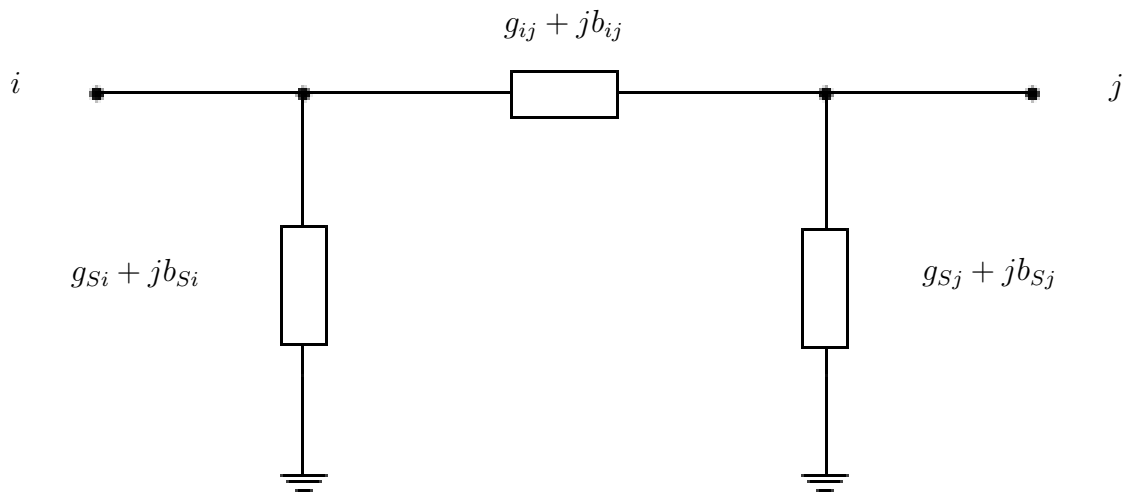


Figure 2.2: Two-port π-model of a transmission line connected between nodes i and j.

We begin with one of the most fundamental expressions in the study of electricity known as Ohm's law formulated by Georg Simon Ohm, a German physicist. Ohm's law is stated as "the current through a conductor between two points, say i and j, is directly proportional to the potential difference across the two points." and may be expressed mathematically as

$$V_{ij} = V_i - V_j = z_{ij} I_{ij} \tag{2.2}$$

where I_{ij} and V_{ij} are the complex current and voltage phasors respectively and z_{ij} is known as the impedance, which is a complex quantity given by $z_{ij} = r_{ij} + jx_{ij}$. The resistance, denoted by r, is the obstruction offered to the flow of direct current, and x_{ij} is termed the reactance is is the resistance offered to the flow of alternating current and is a function of the frequency of the alternating current. Equation (2.2) may also be expressed as

$$I_{ij} = y_{ij}(V_i - V_j) \tag{2.3}$$

where y_{ij} is known as the admittance and is the inverse of the impedance. Furthermore, $y_{ij} = g_{ij} + jb_{ij}$ where g_{ij} and b_{ij} are known as the conductance and susceptance respectively.

A set of fundamental laws that explain the flow of current in an electrical circuit was proposed by the German physicist Gustav Kirchhoff in 1845. In order to develop a model for relating the system state to the conventional measurements we require one of Kirchhoff's laws known as Kirchhoff's first law. It is derived from the principle of conservation of charge and states that "the algebraic sum of currents in a network of conductors meeting at a point is zero." It is expressed mathematically as

$$\sum_{k=1}^{K} I_k = 0 \tag{2.4}$$

where K is the total number of conductors connected to a point and I_k is the current flowing either into the point or away from the point.

Consider a power network consisting of N nodes (buses). We begin by writing a set of nodal equations obtained by applying Kirchhoff's current law at each bus. The vector of net current injections at each bus is denoted by $\bar{\mathbf{i}} \in \mathbb{C}^{N\times 1}$ and the vector of voltage phasors at each bus by $\bar{\mathbf{v}} \in \mathbb{C}^{N\times 1}$, where $\bar{v}_n = |V_n|e^{j\theta_n}$. The nodal equations now take the following form:

$$\bar{\mathbf{i}} = \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_N \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1N} \\ Y_{21} & Y_{22} & \cdots & Y_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ Y_{N1} & Y_{N2} & \cdots & Y_{NN} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_N \end{bmatrix} = \mathbf{Y}\bar{\mathbf{v}}, \tag{2.5}$$

where $\mathbf{Y}$ is called the admittance matrix and any entry Y_{ij} of $\mathbf{Y}$ is given by

$$Y_{ij} = G_{ij} + jB_{ij} = \begin{cases} 0 & \text{if node } m \text{ is not connected to node } n \\ y_{mS} + \sum_{l\in\mathcal{G}_m} y_{ml} & \text{if } m = n \\ -y_{ij} & \text{otherwise} \end{cases} \tag{2.6}$$

where y_{iS} is the sum of all shunt admittances connected at node i, $\mathcal{G}_i$ is the set of all nodes connected to node i, also known as the neighbourhood of i and y_{ij} is the admittance of the line connecting node-i to node-j.

The measurements collected by conventional measuring devices like instrument transformer pairs generally consist of:

1. *Real and reactive power injections*: The real power injection P_i and the reactive power injection Q_i at bus i are related to the state variables $|V_n|$ and θ_n, $n = 1, 2, \ldots, N$, as

$$P_i = |V_i| \sum_{m \in \mathcal{G}_i} |V_m| (G_{im} \cos\theta_{im} + B_{im} \sin\theta_{im}) \tag{2.7}$$

$$Q_i = |V_i| \sum_{m \in \mathcal{G}_i} |V_m| (G_{im} \cos\theta_{im} - B_{im} \sin\theta_{im}) \tag{2.8}$$

2. *Real and reactive power flow*: The real power and reactive flows from bus i to bus j denoted by P_{ij} and Q_{ij} respectively are given by

$$P_{ij} = |V_i|^2 (g_{Si} + g_{ij}) - |V_i||V_j|(g_{ij} \cos\theta_{ij} + b_{ij} \sin\theta_{ij}) \tag{2.9}$$

$$Q_{ij} = -|V_i|^2 (b_{Si} + b_{ij}) - |V_i||V_j|(g_{ij} \sin\theta_{ij} + b_{ij} \cos\theta_{ij}) \tag{2.10}$$

These measurements taken at various points on the power system form the measurement model upon which state estimation is based.

2.3 Overview of State Estimation Methods

State estimation schemes may be classified into three distinct paradigms or frameworks depending on the measurement model upon which the estimation algorithms are based as follows.

- Static state estimation (SSE)
- Forecasting-aided state estimation (FASE)
- Multi-area state estimation (MASE)

Here, a brief overview of the evolution of each of the above frameworks is given.

2.3.1 Static state estimation

For the last four decades, much of the research in power system state estimation has been focused on static state estimation (SSE), primarily due to the fact that the traditional monitoring technologies, such as those implemented in the SCADA system, can only take non-synchronized measurements every two to four seconds. Furthermore, to reduce the computational complexity of implementation, the estimates are usually updated only once every few minutes. Hence the usefulness of

SSE as a means to provide real-time monitoring of the power grid is quite limited in practice.

As mentioned earlier, in an N-bus system, the $(2N-1)\times 1$ state vector has the form $\mathbf{x} = [\theta_2, \theta_3, \ldots, \theta_N, |V_1|, \ldots, |V_N|]^{\mathrm{T}}$. In order to estimate the state $\mathbf{x}$, a set of measurements $\mathbf{z} \in \mathbb{R}^{L\times 1}$, $L > 2N-1$, is collected. These measurements consist of non-synchronized active and reactive power flows in network elements (given by (2.9) and (2.10)), bus power injections (given by (2.7) and (2.8)) and voltage magnitudes at the buses. The measurements are typically obtained within SCADA systems, and are related to the state vector by an overdetermined system of nonlinear equations $\mathbf{h}(\mathbf{x})$, namely

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{n}, \tag{2.11}$$

where $\mathbf{n}$ is a zero-mean Gaussian measurement noise vector with covariance matrix $\mathbf{C}_n \in \mathbb{R}^{L\times L}$.

In the traditional SSE approach first proposed by Schweppe [12], the state vector is estimated from the measurement equation in (2.11) using the weighted least-squares (WLS) method. In particular, the SSE problem is solved by finding

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} [\mathbf{z} - \mathbf{h}(\mathbf{x})]^{\mathrm{T}} \mathbf{W}^{-1} [\mathbf{z} - \mathbf{h}(\mathbf{x})] \tag{2.12}$$

where weighting matrix $\mathbf{W}$ is commonly taken as diagonal with elements related to background noise covariance as $\mathbf{W} = \mathbf{C}_n$.

The solution for $\hat{\mathbf{x}}$ is obtained in an iterative fashion by linearising (2.11) around the available estimate (at iteration j) and applying the Gauss-Newton algorithm to improve the estimate, using the following equations:

$$\mathbf{G}(j)\boldsymbol{\Delta}\mathbf{x}(j) = \mathbf{H}^{\mathrm{T}}(j)\mathbf{W}^{-1}[\mathbf{z} - \mathbf{h}(\mathbf{x}(j)] \tag{2.13}$$

$$\hat{\mathbf{x}}(j+1) = \hat{\mathbf{x}}(j) + \boldsymbol{\Delta}\mathbf{x}(j) \tag{2.14}$$

where $\mathbf{G}(j) = \mathbf{H}^{\mathrm{T}}(j)\mathbf{W}^{-1}\mathbf{H}(j)$ is the gain matrix at iteration j. The Jacobian matrix, $\mathbf{H}(j) \in \mathbb{R}^{L\times(2N-1)}$, at each iteration, is the first-order partial derivative of $\mathbf{h}(\mathbf{x})$, with respect to $\mathbf{x}$, evaluated at $\hat{\mathbf{x}}(j)$, and is given by

$$\mathbf{H}(j) = \left.\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(j)} \tag{2.15}$$

Equation (2.13) is usually referred to as the *normal equation*.

The iterative process is terminated when the norm of the residual falls below a predefined value, i.e. for some $\delta > 0$, $\|\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}(j))\|^2 \leq \delta$, and the covariance of the estimate is given by

$$\mathbf{G}^{-1}(j) = [\mathbf{H}^{\mathrm{T}}(j)\mathbf{W}^{-1}\mathbf{H}(j)]^{-1} \tag{2.16}$$

One of the main problems in solving the normal equation in (2.13) is computational complexity. An approach to reduce the computational cost is to take into account the fact that $\mathbf{G}(j)$ is sparse and symmetric. A more common approach in the literature is to take advantage of the sparseness of matrix $\mathbf{H}(j)$, which is in general even more sparse than $\mathbf{G}(j)$, and employ a robust and computationally

efficient QR factorization of the weighted Jacobian, e.g. using a sequence of Givens rotations (or Householder reflections) of the weighted Jacobian matrix $\mathbf{W}^{-\frac{1}{2}}\mathbf{H}(j)$. A good treatment of this topic and related important references can be found in [13]. Specifically, let $\mathbf{Q}(j) \in \mathbb{R}^{L\times L}$ be an orthogonal transformation that triangularizes the weighted Jacobian as follows

$$\mathbf{Q}(j)\mathbf{W}^{-1/2}\mathbf{H}(j) = \begin{bmatrix}\mathbf{R}(j)\\ \mathbf{0}\end{bmatrix} \tag{2.17}$$

where $\mathbf{R}(j) \in \mathbb{R}^{(2N-1)\times(2N-1)}$ is upper/lower triangular. We may now rewrite (2.13) as

$$\begin{aligned}&\begin{bmatrix}\mathbf{R}^{\mathrm{T}}(j) & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{R}(j)\\ \mathbf{0}\end{bmatrix}\boldsymbol{\Delta}\mathbf{x}(j)\\ &\quad = \begin{bmatrix}\mathbf{R}^{\mathrm{T}}(j) & \mathbf{0}\end{bmatrix}\mathbf{Q}(j)\mathbf{W}^{-1/2}[\mathbf{z}-\mathbf{h}(\mathbf{x}(j))]\end{aligned} \tag{2.18}$$

which allows us to solve for $\boldsymbol{\Delta}\mathbf{x}(j)$ in two stages

$$\begin{bmatrix}\mathbf{y}_1(j)\\ \mathbf{y}_2(j)\end{bmatrix} = \mathbf{Q}(j)\mathbf{W}^{-1/2}[\mathbf{z}-\mathbf{h}(\mathbf{x}(j))] \tag{2.19}$$

$$\mathbf{R}(j)\boldsymbol{\Delta}\mathbf{x}(j) = \mathbf{y}_1(j) \tag{2.20}$$

where $\mathbf{y}_1(j) \in \mathbb{R}^{(2N-1)\times 1}$, as seen in (2.19), is formed by taking the $2N-1$ first elements of the transformed (weighted) measurement error vector. The correction term $\Delta\mathbf{x}(j)$ is obtained via backward (or forward) substitution.

To overcome the computational cost associated with directly solving (2.13), it has often been argued that the gain matrix $\mathbf{G}(j)$ does not change considerably during several iterations, which implies we can assume a piecewise constant Jacobian matrix [12]. This observation is exploited in the *hybrid method* proposed by Monticelli and Garcia [14] to reduce storage requirements when applying orthogonal transformations. In particular, the triangular matrix $\mathbf{R}$ in (2.17) remains constant for those iterations when the measurement Jacobian is not re-evaluated. Thus, by only transforming the left-hand side of (2.13) we may acquire the correction term $\Delta\mathbf{x}(j)$ from

$$\mathbf{R}^{\mathrm{T}}\mathbf{R}\Delta\mathbf{x}(j) = \mathbf{H}^{\mathrm{T}}\mathbf{W}^{-1}[\mathbf{z}-\mathbf{h}(\mathbf{x}(j))]$$

Comparing with (2.19)–(2.20) we see that only $\mathbf{R}$ need to be stored (and not factors of $\mathbf{Q}$) at the expense of an additional forward (or backward) substitution.

The computational complexity of the aforementioned SE approaches may be further reduced by assuming voltage magnitudes and phases to be independent [13]. The state estimate is then obtained by solving two decoupled WLS problems since the measurement Jacobian becomes block-diagonal. This approach renders a particularly efficient implementation of the hybrid method both in terms of storage requirements and computational cost.

A more recent approach proposed by Gómez-Expósito *et al.* for reducing the computational cost is to use a nested, or multi-level, formulation of the nonlinear measurement model [4]. This approach can sustain growth in size, complexity, data

and it is designed to function at different levels of the modeling hierarchy in order to accomplish very large-scale interconnection-wide monitoring. This method uses the same overdetermined set of measurement equations as in (2.11). It is then "unfolded" into K sequential WLS problems by introducing a set of intermediate variables $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_K\}$ with the following nested structure

$$\begin{aligned} \mathbf{z} &= \mathbf{f}_1(\mathbf{y}_1) + \mathbf{n} \\ \mathbf{y}_1 &= \mathbf{f}_2(\mathbf{y}_2) + \mathbf{n}_1 \\ &\vdots \\ \mathbf{y}_{K-1} &= \mathbf{f}_K(\mathbf{y}_K) + \mathbf{n}_{K-1} \\ \mathbf{y}_K &= \mathbf{f}_{K+1}(\mathbf{x}) + \mathbf{n}_K \end{aligned} \tag{2.21}$$

The set $\mathcal{Y}$ is chosen such that the solution of the K-tuple (2.21) offers an advantage over solving equation (2.11), e.g. reduction of computational complexity or amount of information exchange between different levels. This is a particularly appealing solution when the measurement model can be factorized into both linear and nonlinear parts, which may be suitable for a hierarchical structure that comprises a linear substation model and a nonlinear transmission level model leading to a factorized form of the measurement model.

2.3.2 Forecasting-aided state estimation

Conventional SSE relies on a single set of measurements all taken at one snapshot in time. Hence it disregards the evolution of the states over consecutive measurement instants. The basic idea of forecasting-aided state estimation (FASE) is to provide a recursive update of the state estimate that can also track the changes occurring during normal system operation. One of the advantages of FASE is that it includes by design a forecasting feature that can get around the problem of missing measurements, as the predicted states may be used in lieu of those measurements. Note, however, that FASE is somewhat different from the true *dynamic* SE since the transients in power systems usually occur at a much faster time scale than those considered in FASE.

The first step towards a dynamic state estimator was taken by Debs and Larson in 1970 [15]. A simple state transition model was developed assuming the system was in a quasi-steady state. Tracking state estimators [16] came next, but the problem here was that no time evolution model was assumed explicitly to follow the dynamics of the system. The next breakthrough in FASE came from Leite da Silva *et al.* [17] who developed a more appropriate state transition model and used Kalman filtering and an exponential smoothing algorithm for forecasting. A robust FASE algorithm based on M-estimation was introduced by Durgaprasad and Thakur in [18] as an alternative to the Kalman filter based approaches and more recently, a FASE algorithm was proposed by Valverde and Terzija based on unscented Kalman filter (UKF) [19]. A more extensive literature survey and related references may be found in [20].

A typical FASE is formulated with the following dynamic model [17]

$$\mathbf{x}(k+1) = \mathbf{F}(k)\mathbf{x}(k) + \mathbf{g}(k) + \mathbf{w}(k) \tag{2.22}$$

where for time instant k, $\mathbf{F}(k) \in \mathbb{R}^{(2N-1)\times(2N-1)}$ is the state-transition matrix, vector $\mathbf{g}(k)$ is associated with the trend behaviour of the state trajectory, and $\mathbf{w}(k)$ is assumed to be zero-mean Gaussian noise with covariance matrix $\mathbf{C}_w$.

Using (2.22) and the measurements arriving at instant k, $\mathbf{z}(k) = \mathbf{h}\left[\mathbf{x}(k)\right] + \mathbf{n}(k)$, the majority of the FASE algorithms that appear in the literature are based on the extended Kalman filter (EKF) [7], whose recursions are given by

$$\hat{\mathbf{x}}(k+1) = \tilde{\mathbf{x}}(k+1) + \mathbf{K}(k+1)\left[\mathbf{z}(k+1) - \mathbf{h}(\tilde{\mathbf{x}}(k+1))\right] \tag{2.23}$$

where $\mathbf{H}(k+1)$ is the measurement Jacobian evaluated at $\tilde{\mathbf{x}}(k+1)$, $\tilde{\mathbf{x}}(k+1)$ is the predicted or forecast value of the state estimate at time $k+1$ given by

$$\tilde{\mathbf{x}}(k+1) = \mathbf{F}(k)\hat{\mathbf{x}}(k) + \mathbf{g}(k), \tag{2.24}$$

$\mathbf{K}(k+1)$ is the Kalman gain given by

$$\mathbf{K}(k+1) = \mathbf{\Sigma}(k+1)\mathbf{H}^{\mathrm{T}}(k+1)\mathbf{C}_n^{-1}, \tag{2.25}$$

$\mathbf{\Sigma}(k+1)$ is the *a posteriori* estimation error covariance matrix given by

$$\mathbf{\Sigma}(k+1) = \left[\mathbf{H}^{\mathrm{T}}(k+1)\mathbf{C}_n^{-1}\mathbf{H}(k+1) + \mathbf{M}^{-1}(k+1)\right]^{-1} \tag{2.26}$$

and $\mathbf{M}(k+1)$ is the *a priori* estimation error covariance matrix given by

$$\mathbf{M}(k+1) = \mathbf{F}(k)\mathbf{\Sigma}(k)\mathbf{F}^{\mathrm{T}}(k) + \mathbf{C}_w. \tag{2.27}$$

We note that matrix $\mathbf{F}(k)$ and vector $\mathbf{g}(k)$ in (2.22) are in most FASE related papers recursively updated using the classic Holt-Winters method [17]. This rather naive state-transition model appears to work quite well, although it ignores any coupling between state variables.

2.3.3 Multi-area state estimation

Multi-area state estimation (MASE) traces its origins back to the late 70s when microprocessor technology was not mature enough to handle the computational load of SE in very large interconnections and SE was implemented on multiprocessor computing architectures. Since the power grid is inevitably a large network, a centralized solution to the associated SE problem amounts to tremendous computational complexity. An alternative is to divide the large power system into smaller areas, each equipped with a local processor to provide a local SE solution. Comparing to a centralized SE approach, MASE reduces the amount of data that each state estimator needs to process (hence reduces complexity) and it improves the robustness of the system by distributing the knowledge of the state. However, its implementation

requires additional communication overhead and it comes with the time-skewness problem that results from asynchronous measurements obtained in different areas.

In MASE, each area has local measurements formulated by

$$\mathbf{z}_m = \mathbf{h}_m(\mathbf{x}_m) + \mathbf{n}_m, \quad m = 1, \ \ldots, \ M \tag{2.28}$$

where $\mathbf{x}_m = [\mathbf{x}_{\mathbf{i}m}^{\mathrm{T}} \ \mathbf{x}_{\mathbf{b}m}^{\mathrm{T}}]^{\mathrm{T}}$ is the local state vector of area m, which is further partitioned into *internal state variables*, $\mathbf{x}_{\mathbf{i}m}^{\mathrm{T}}$, and *border state variables*, $\mathbf{x}_{\mathbf{b}m}^{\mathrm{T}}$. Internal variables are those state variables that are observable for the particular area while border variables are states of those buses with lines connecting two areas (so-called tie-lines).

A local estimate can be obtained from (2.28) using the techniques outlined in earlier sections with the difference that the measurement Jacobian is derived from the local estimate. Taking into account the coupling between areas located in close proximity, improved state estimates can be obtained by combining local estimates using either a hierarchical structure, a decentralized structure, or a combination of both. In the hierarchical scheme, a central computer controls the local processors which may be either located in disparate geographical areas (distributed architecture) or in the same area (parallel architecture). The local state estimators communicate only with the central computer. In the fully decentralized architecture, i.e. no central computer, each local state estimator communicates only with its neighbours. The amount of data exchange of the solution depends on whether local estimates (or measurements) are transmitted at every iteration of the local estimation algorithm or upon convergence. A survey of various MASE methods is given in [21] along with a good treatment of a two-level hierarchical MASE example. More recently, an enhanced MASE scheme was introduced by Korres [22] which performs state estimation in a fully distributed manner.

2.4 Discussion

The development of computationally efficient algorithms will continue to be a challenge although research on state estimation in power systems may seem to be quite mature. This is because as the power grid becomes more complex, intelligent and interconnected, outcomes of such research will be beneficial to the development of improved energy management systems of the future. For example, since the Jacobian matrix $\mathbf{H}$ is sparse [23], efficient solutions which have been developed recently for sparse systems can be applied, thus enabling reduced-complexity solutions.

The forecasting-aided state estimation (FASE) algorithms found in the literature are based on the extended Kalman filter whose limitations are well known. Fast converging algorithms with good tracking capabilities may be found by exploiting the equivalence between the incremental Gauss-Newton algorithm and the iterated extended Kalman filter [24]. The use of the unscented Kalman and H_∞ filters in power system state estimation also deserves further investigation. Furthermore, novel algorithms need to be developed which incorporate PMUs in the FASE paradigm and

the optimality of such algorithms deserves to be studied. This problem is studied in detail and solutions are proposed and verified in Chapter 4.

Multi-area state estimation can most likely benefit from recent advances in signal processing, particularly distributed estimation. A fully distributed state estimator (static or dynamic) that provides local awareness of the whole state vector can exploit the fruits of average consensus techniques that abound in the literature, see, e.g. [25, 26]. Taking into account the sparseness of the problem, dynamic Kalman filter based solutions for sparse systems, e.g. [27], can be useful when the amount of information shared between neighbours is kept to a minimum to reduce either the amount of communication or if the neighbour is a competing operator in a deregulated market. Algorithms based on *event-triggered* state estimation and communication hold the potential to drastically reduce the computational and communication resources used by state estimators. This means that larger and more complex state estimation problems can be solved in the future without the need to overhaul the existing infrastructure of the SCADA/EMS. A new, event-triggered MASE scheme has been developed in Chapter 5 which reduces the computational and communication resources used.

Chapter 3

Impact of Phasor Measurement Units on State Estimation

A PMU is a transducer which converts a three-phase analogue voltage or current signal into a precision time-tagged positive sequence phasor. PMUs provide accurate measurements of positive sequence voltage and current phasors. The main problem faced by engineers today is to combine those PMU measurements with conventional measurements to obtain an optimal state estimate. This chapter examines the impact that synchrophasor technology has had on the state estimation.

3.1 Measurement Acquisition

Phasor measurement units calculate the positive sequence voltage and current phasors from signals sampled from instrument transformer secondary windings. Figure 3.1 shows the major components of a modern PMU. Typically, the current and voltage signals are oversampled and then conditioned using a combination of analogue and digital anti-aliasing filters. The sampling clock is phase-locked with the GPS clock pulse. A microprocessor then estimates the phasors by using a discrete Fourier transform. A time stamp is created based on two signals derived from the GPS receiver and this, along with the current and voltage phasor data is transmitted to a data concentrator through a modem.

Figure 3.2 shows how PMU measurements are collected and sent to the SCADA system. PMUs are placed at substations and provide measurements of time-stamped positive sequence voltages and currents of all monitored buses and feeders. At the next level of the measurement hierarchy we find *phasor data concentrators* (PDCs). The function of a PDC is to gather data from several PMUs, reject bad data and align the time stamps while creating a coherent record of simultaneously recorded data from the wider-area interconnection. On a system-wide scale, in the case of a centralized EMS/SCADA, a higher level may be envisioned known as the *super data concentrator* (Super PDC) [28] with a functionality similar to that of the PDC.

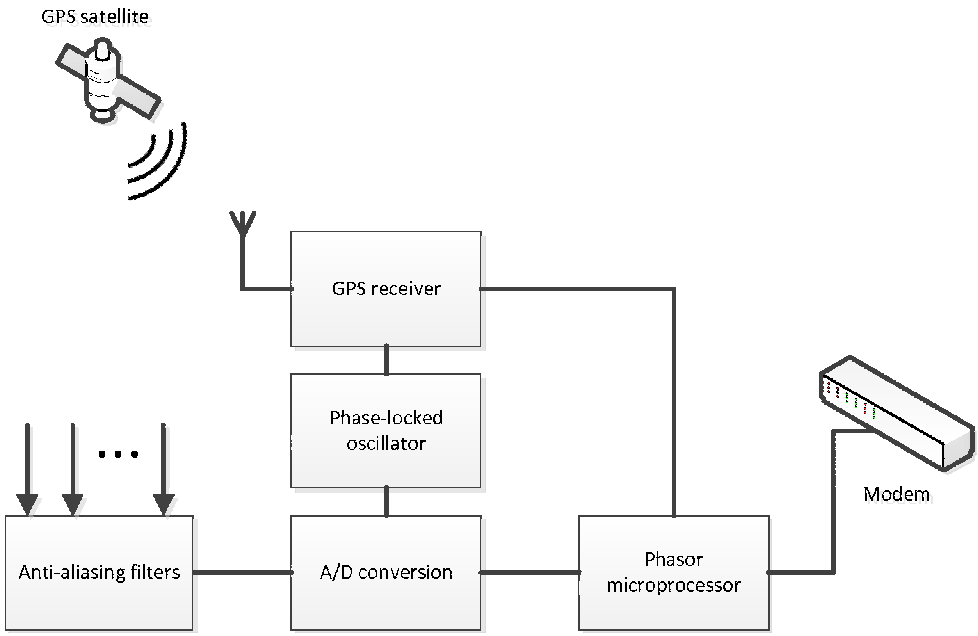


Figure 3.1: Important components of a modern PMU [28].

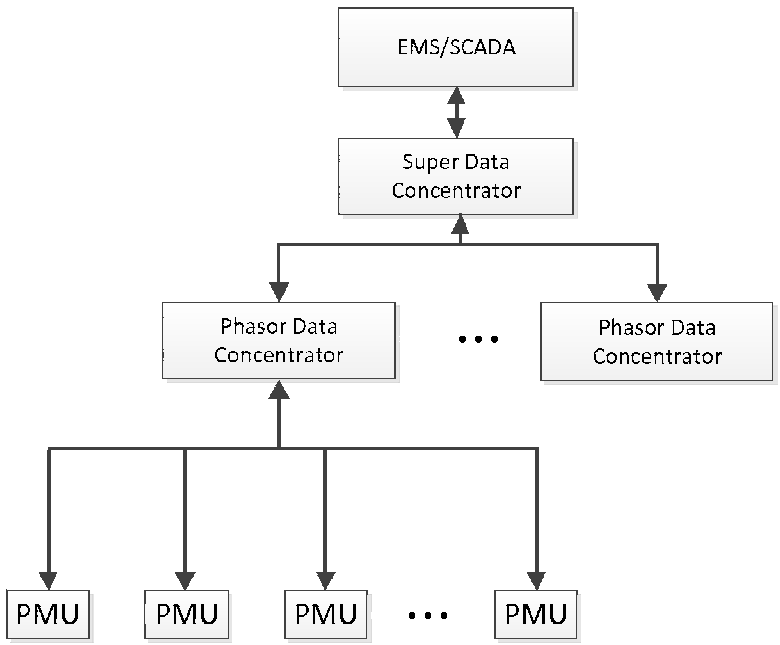


Figure 3.2: Hierarchy of phasor measurement systems and phasor data concentrators.

3.2 Structure of the PMU Measurement Set

In this section we derive a relationship between the PMU measurements and the state vector. However, instead of using the traditional polar form of the voltage phasors, we define a new form of the state vector in Cartesian co-ordinates. This

new state vector denoted by $\mathbf{v}$ has the following form:

$$\mathbf{v} = [\Re\{V_1\},\ \ldots,\ \Re\{V_N\}\ \Im\{V_2\},\ \ldots,\ \Im\{V_N\}]^{\mathrm{T}}, \tag{3.1}$$

where $\Re\{V_n\}$ and $\Im\{V_n\}$ are, respectively, the real and imaginary parts of the voltage phasor at the nth bus [5]. As before, bus 1 is the slack bus and this implies that the imaginary part of the voltage phasor at bus 1 is zero.

Consider the π-model of a transmission line connecting nodes p and q as shown in Figure 3.3. Let us assume there is a PMU at node p. Let the complex voltage phasors at p and q be V_p and V_q, respectively. The PMU measures the current phasor I_p and the voltage phasor V_p. The state vector for this system is given by

$$\mathbf{v} = \begin{bmatrix}\Re\{V_p\} & \Re\{V_q\} & \Im\{V_p\} & \Im\{V_q\}\end{bmatrix}^T. \tag{3.2}$$

The voltage measurements of the PMU at p are given by

$$\begin{aligned}\mathbf{z}_{v_p} &= \begin{bmatrix}\Re\{V_p\}\\ \Im\{V_p\}\end{bmatrix} + \mathbf{n}_1\\ &= \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\mathbf{v} + \mathbf{n}_1\\ &= \mathbf{B}\mathbf{v} + \mathbf{n}_1\end{aligned} \tag{3.3}$$

where $\mathbf{z}_{v_p}$ is the vector of PMU voltage measurements and $\mathbf{n}_1$ is zero-mean Gaussian measurement noise.

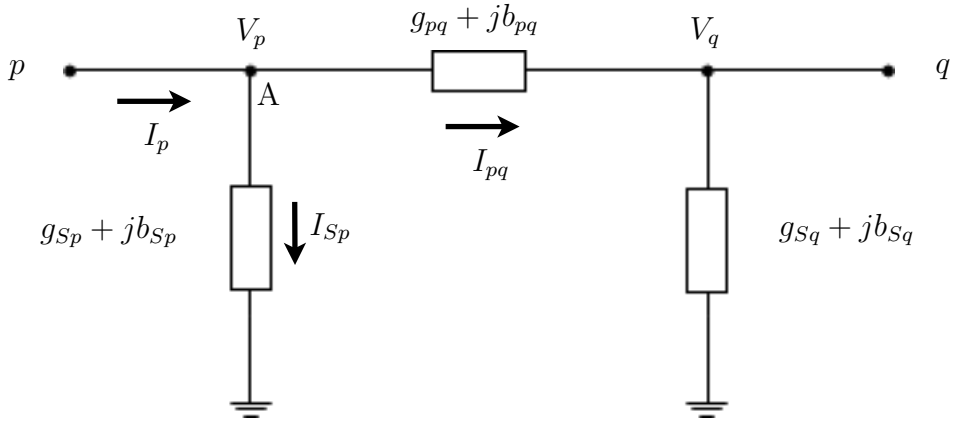


Figure 3.3: Two-port π-model of a transmission line connecting nodes p and q. A PMU is placed at node p.

Similarly, the current measurements of the PMU at p are given by

$$\mathbf{z}_{i_p} = \begin{bmatrix}\Re\{I_p\}\\ \Im\{I_p\}\end{bmatrix} + \mathbf{n}_2 \tag{3.4}$$

where $\mathbf{z}_{i_p}$ is the vector of PMU measurements and $\mathbf{n}_2$ is zero-mean Gaussian measurement noise.

In order to relate the current I_p to the state vector we apply Kirchhoff's current-law at node A. Now, we have

$$I_p = I_{pq} + I_{Sp} \tag{3.5}$$
$$= (g_{pq} + jb_{pq})(V_p - V_q) + (g_{Sp} + jb_{Sp})Vp. \tag{3.6}$$

Simplifying the above and separating the real and imaginary parts, we have

$$\Re\{I_p\} = (g_{pq} + g_{Sp})\Re\{V_p\} - g_{pq}\Re\{V_q\} - (b_{pq} + b_{Sp})\Im\{V_p\} + b_{pq}\Im\{V_q\} \tag{3.7}$$

and

$$\Im\{I_p\} = (b_{pq} + b_{Sp})\Re\{V_p\} - b_{pq}\Re\{V_q\} + (g_{pq} + g_{Sp})\Im\{V_p\} - g_{pq}\Im\{V_q\}. \tag{3.8}$$

Combining (3.7) and (3.8), we have

$$\begin{bmatrix} \Re\{I_p\} \\ \Im\{I_p\} \end{bmatrix} = \begin{bmatrix} (g_{pq} + g_{Sp}) & -g_{pq} & -(b_{pq} + b_{Sp}) & b_{pq} \\ (b_{pq} + b_{Sp}) & -b_{pq} & (g_{pq} + g_{Sp}) & -g_{pq} \end{bmatrix} \mathbf{v} \tag{3.9}$$
$$= \mathbf{Y}\mathbf{v}. \tag{3.10}$$

Substituting (3.9) in (3.4), we have

$$\mathbf{z}_{i_p} = \mathbf{Y}\mathbf{v} + \mathbf{n}_2. \tag{3.11}$$

By combining (3.3) and (3.11) and extending the reasoning employed in their derivation to an N bus system, we have

$$\mathbf{z} = \begin{bmatrix} \mathbf{B} \\ \mathbf{Y} \end{bmatrix} \mathbf{v} + \mathbf{n} \tag{3.12}$$
$$= \mathbf{A}\mathbf{v} + \mathbf{n} \tag{3.13}$$

where $\mathbf{z}$ is the vector of PMU measurements with L_1 voltage measurements and L_2 current measurements, $\mathbf{B} \in \mathbb{R}^{L_1 \times (2N-1)}$ is a matrix in which each row is a vector of zeros with a one in the column associated with the buses containing PMUs and $\mathbf{Y} \in \mathbb{R}^{L_2 \times (2N-1)}$ is a matrix of admittances. The zero-mean Gaussian measurement noise is given by $\mathbf{n}$.

3.3 State Estimation with PMUs in Literature

The first step towards a realizable PMU came with the introduction of the symmetrical component distance relay by Phadke *et al.* [29, 30]. PMUs may be used for a host of other purposes apart from state estimation, which is arguably one of the most important application. Some other applications of PMUs include fault recording, disturbance recording and transmission and generation modelling verification [31].

An important issue which arises in applying PMUs for state estimation is that of the optimal placement of PMUs. Yuill *et al.* present an overview of various methods which exist in the literature to decide where to place PMUs optimally [32].

When a sufficient number of PMUs are deployed on the grid, the system is fully observable and iterative solutions are avoided as the system becomes linear as seen in (3.12). Even though making the system fully observable using PMUs is not yet realizable due to financial constraints, it seems likely that in the near future, we could see large-scale deployment of PMUs in power grid as the deployment costs decrease. However, presently, there is a need for state estimators that combine conventional SCADA and PMU measurements.

Static state estimation using both PMU and traditional SCADA measurements has been studied extensively. There are two ways to include PMU measurements in the SE process [5]. One method is to use a single state estimator, where PMU measurements are mixed with the traditional power flow measurements. This type of state estimator is also called an integrated state estimator. Another method is to use a two-stage scheme, where the state estimate obtained from the traditional SCADA measurements in (2.13) is improved by using a second estimator that employs PMU measurements only. The latter method has the advantage of leaving the existing SCADA software intact, while the former generally shows better performance in terms of accuracy and redundancy.

Let us first consider the approach when conventional SCADA measurements given by (2.11) are mixed with PMU measurements given by (3.12). In order to jointly process the measurements we first need to relate the PMU state $\mathbf{v}$ of complex phasors (Cartesian coordinates) to the conventional state vector $\mathbf{x}$ (polar coordinates), through a simple nonlinear transformation, i.e. there exists a one-to-one mapping $\mathbf{d} : \mathbb{R} \to \mathbb{R}$ between them defined by $\mathbf{v} = \mathbf{d}(\mathbf{x})$. Thus, a single estimator, static or dynamic, that incorporates both conventional and PMU measurements can be derived based on the following augmented measurement model:

$$\begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{h}(\mathbf{x}) \\ \mathbf{A}\mathbf{d}(\mathbf{x}) \end{bmatrix} + \begin{bmatrix} \mathbf{n}_1 \\ \mathbf{n}_2 \end{bmatrix}, \tag{3.14}$$

where $\mathbf{z}_1$ and $\mathbf{n}_1$ are the conventional measurements and noise vectors, and $\mathbf{z}_2$ and $\mathbf{n}_2$ denote the PMU measurements and noise vectors respectively.

Instead of mixing the measurements, we may use a two-step approach, also known as a hierarchical state estimator, where the conventional state estimate $\hat{\mathbf{x}}$ from (2.13) is converted into voltage phasors, i.e. $\hat{\mathbf{v}}_1 = \mathbf{d}(\hat{\mathbf{x}})$, and then used as additional measurements in an augmented form of the linear measurement model given in (3.12)

$$\begin{bmatrix} \hat{\mathbf{v}}_1 \\ \mathbf{z}_2 \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{B}} \\ \mathbf{Y} \end{bmatrix} \mathbf{v} + \begin{bmatrix} \tilde{\mathbf{n}}_1 \\ \mathbf{n}_2 \end{bmatrix}, \tag{3.15}$$

where $\tilde{\mathbf{B}}$, like $\mathbf{B}$ in (3.12), simply sifts out the relevant phasors, and $\tilde{\mathbf{n}}_1$ is the noise vector associated with the (transformed) conventional measurements. We may now solve for the unknown phasors $\mathbf{v}$ using a linear weighted least squares approach [5]. Recently, a constrained formulation for state estimation using both conventional meaurements and PMU measurements was presented in [33]. A method for weighting PMU measurements for use in a two-stage scheme based on PMU measurement uncertainty was introduced by the same authors in [34]

The problem of multi-area state estimation incorporating phasor measurements was first introduced by Zhao and Abur [35] who present a hierarchical scheme for distributed state estimation using PMU measurements. Jiang *et al.* [36] also use PMU measurements in each region to obtain a hierarchical state estimator that functions in three steps. A distributed state estimator which has the same accuracy and redundancy as an integrated state estimator has been introduced by Jiang *et al.* based on *diakoptics*. Diakoptic literally means "tear-through", and consists of breaking a system down into subsystems which can be solved independently before being assembled back together to obtain a solution to the whole problem. The inclusion of PMU measurements in multi-level state estimators has also been considered in [4]. A recent two-level MASE scheme which incorporates PMU measurements both at the substation and the control center level was introduced by Yang *et al.* in [37]

3.4 Discussion

In spite of the substantial body of research work discussed above, many well-known challenges remain in combining PMU measurements that are of a much higher quality with conventional measurements to obtain an optimal state estimate, namely:

Computational burden - The dimensions of the vectors and matrices involved in the state estimation process are increased due to the inclusion of PMU measurements. This results in an increased computational burden on the EMS/SCADA.

Data tsunami [1] - The sampling rate of PMUs is around two orders of magnitude higher than the conventional measurements. Novel techniques need to be developed to extract relevant state information from this tidal wave of measurement data.

Degraded numerical stability - Since PMU measurements are significantly more accurate than traditional measurements, inclusion of those measurements in the estimation process often results in ill-conditioned gain or measurement noise covariance matrices.

Time skewness - Synchronized PMU measurements are sampled much faster than non-synchronized conventional measurements. These two sets of measurements have significantly different sampling rates and are not synchronized with each other.

Retrofitting - According to utility companies, PMUs are currently not used in performing state estimation since retrofitting is a rather onerous task and can be quite costly.

[1] Data tsunami is a term used in information science to refer to a tidal wave of data which swamps computer systems rendering them incapable of separating wheat from chaff.

In short, since PMUs will be deployed in ever increasing numbers in power grids of the future, there is a need for a new set of state estimation algorithms which take into account the issues mentioned above. In the following chapter, we shall introduce two such algorithms.

Chapter 4

Forecasting-Aided State Estimation with Phasor Measurements

As the power grid continues to evolve, it will become more complex and dynamic. Therefore, it becomes necessary to closely track state changes to ensure the dependability, reliability and security of the power system. To this end, the use of forecasting-aided state estimation seems justified since it treats the state vector as a dynamically evolving variable and seeks to track its evolution in time. Furthermore, static state estimation puts a considerable computational load on the EMS/SCADA since the conventional Gauss-Newton weighted least-squares approach requires several iterations at each time instant until it converges. On the other hand, the FASE approach is computationally more resource-efficient efficient since only one iteration is performed at each time instant. In this chapter, we present two novel algorithms to incorporate PMUs into the FASE framework.

State estimators employing PMU measurements can also be cast into the FASE framework in a manner described in the previous chapter, i.e. either PMU measurements are mixed with conventional measurements or included in a post-processing step. In the mixed approach, extended Kalman filters (EKF) can be derived for the mixed data model given by (3.14), reproduced here for convenience:

$$\begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{h}(\mathbf{x}) \\ \mathbf{A}\mathbf{d}(\mathbf{x}) \end{bmatrix} + \begin{bmatrix} \mathbf{n}_1 \\ \mathbf{n}_2 \end{bmatrix}, \tag{4.1}$$

where $\mathbf{z}_1$ and $\mathbf{n}_1$ are the conventional measurements and noise vectors, and $\mathbf{z}_2$ and $\mathbf{n}_2$ denote the PMU measurements and noise vectors, respectively. Matrix $\mathbf{A}$ relates the PMU measurements to the Cartesian form of the state vector given by $\mathbf{d}(\mathbf{x})$.

Alternatively, if the PMU measurements are included in a post-processing step, a linear Kalman filter can be used, or one may also use a simple linear weighted least-squares estimator at the second stage. However, mixing measurements of different qualities into a single state estimator may sometimes cause the covariance matrix of the combined noise vector to become ill-conditioned. In addition to this, the dimensions of the vectors and matrices involved in the SE process are increased

due to the additional PMU measurements, which may lead to significantly increased computational complexity. For this reason, the problem can be cast into a constrained Kalman filtering problem where high-quality measurements are employed as deterministic equality or inequality constraints. For example, more robust FASEs with a reduced order Kalman filter can be derived by applying the ideas proposed in [38], an approach considered here in detail.

In this chapter, we present two novel FASE methods for incorporating phasor measurements in forecasting-aided EKF-based state estimators. These results may be extended to other forms of nonlinear Kalman filters as well.

4.1 Mixed-Measurement Extended Kalman Filter

In the mixed-measurement estimation model described by (4.1), the PMU measurements are appended to the conventional measurements. Although a static state estimator for the same problem has been derived in [5], a more computationally efficient solution to the problem would be to cast it into the FASE framework.

The state transition model remains the same as in conventional FASE, i.e.

$$\mathbf{x}(k+1) = \mathbf{F}(k)\mathbf{x}(k) + \mathbf{g}(k) + \mathbf{w}(k),$$

where for time instant k, $\mathbf{F}(k) \in \mathbb{R}^{(2N-1)\times(2N-1)}$ is the state-transition matrix, vector $\mathbf{g}(k)$ is associated with the trend behavior of the state-trajectory, and $\mathbf{w}(k)$ is assumed to be zero-mean Gaussian noise with covariance matrix $\mathbf{C}_w$.

The dynamic version of measurement model given by (4.1) at a time instant k can be written as

$$\begin{bmatrix}\mathbf{z}_1(k)\\ \mathbf{z}_2(k)\end{bmatrix} = \begin{bmatrix}\mathbf{h}\{\mathbf{x}(k)\}\\ \mathbf{A}\mathbf{d}\{\mathbf{x}(k)\}\end{bmatrix} + \begin{bmatrix}\mathbf{n}_1(k)\\ \mathbf{n}_2(k)\end{bmatrix}, \tag{4.2}$$

where $\mathbf{z}_1(k)$ and $\mathbf{n}_1(k)$ are the conventional measurements and noise vectors, and $\mathbf{z}_2(k)$ and $\mathbf{n}_2(k)$ denote the PMU measurements and the (transformed) noise vectors. The noise vectors $\mathbf{n}_1(k)$ and $\mathbf{n}_2(k)$ are assumed to be zero-mean Gaussian distributed with covariance matrices $\mathbf{C}_1$ and $\mathbf{C}_2$, respectively.

The state vector in Cartesian co-ordinates, denoted by $\mathbf{v}$, is related to the conventional state vector in polar co-ordinates, denoted by $\mathbf{x}$, through a simple nonlinear transformation, i.e there exists a one-to-one mapping $d : \mathbb{R} \to \mathbb{R}$ between them given by

$$\mathbf{v}(k) = \mathbf{d}\{\mathbf{x}(k)\}. \tag{4.3}$$

Now, writing (4.2) as a single expression and combining it with the state transition model we have a pair of equations given by

$$\mathbf{x}(k+1) = \mathbf{F}(k)\mathbf{x}(k) + \mathbf{g}(k) + \mathbf{w}(k), \tag{4.4}$$

$$\bar{\mathbf{z}}(k) = \bar{\mathbf{h}}(\mathbf{x}[k]) + \bar{\mathbf{n}}(k). \tag{4.5}$$

where the mixed measurement set is denoted by $\bar{\mathbf{z}}(k) = \begin{bmatrix}\mathbf{z}_1^T(k) & \mathbf{z}_2^T(k)\end{bmatrix}^T$, $\bar{\mathbf{h}}\{\mathbf{x}(k)\} = \begin{bmatrix}\mathbf{h}^T\{\mathbf{x}(k)\} & (\mathbf{A}\mathbf{d}\{\mathbf{x}(k)\})^T\end{bmatrix}^T$ and $\bar{\mathbf{n}}(k) = \begin{bmatrix}\mathbf{n}_1^T(k) & \mathbf{n}_2^T(k)\end{bmatrix}^T$. We shall now derive

an EKF for this model known as MM/EKF which stands for mixed-measurement extended Kalman filter.

4.1.1 The estimation algorithm

At a time instant k, the first step in the estimation process is to predict the state vector at the next time instant $k+1$. In order to make the forecast, we employ a well-known forecasting tool known as Holt's two-parameter linear exponential smoothing, a technique used in extensively in time-series analysis. The application of this technique for forecasting-aided state estimation in power systems was first introduced in [17]. Let $x_i(k)$ be the ith component of the true state vector $\mathbf{x}(k)$. Now, if $\tilde{x}_i(k)$ and $\tilde{x}_i(k+1)$ are the predictions at time instants k and $k+1$, according to Holt's two parameter linear exponential smoothing,

$$\tilde{x}_i(k+1) = a_i(k) + b_i(k), \tag{4.6}$$

where

$$\begin{aligned} a_i(k) &= \alpha_i \hat{x}_i(k) + (1-\alpha_i)\tilde{x}_i(k) \\ b_i(k) &= \beta_i[a_i(k) - a_i(k-1)] + (1-\beta_i)b_i(k-1), \end{aligned}$$

and α_i and β_i are the smoothing parameters. Typically, in FASE, α_i and β_i are the same for all values of i. Furthermore, (4.6) can be rewritten as

$$\tilde{x}_i(k+1) = f_i(k)x(k) + g_i(k), \tag{4.7}$$

where

$$\begin{aligned} f_i(k) &= \alpha_i(1+\beta_i) \\ g_i(k) &= (1+\beta_i)(1-\alpha_i)\tilde{x}_i(k) - \beta_i a_i(k-1) + (1-\beta_i)b_i(k-1). \end{aligned}$$

Writing the above expressions in matrix form,

$$\tilde{\mathbf{x}}(k+1) = \mathbf{F}(k)\mathbf{x}(k) + \mathbf{g}(k). \tag{4.8}$$

Next, we employ an EKF to correct the forecast made in the first step. The corrected value, also the state estimate at time $k+1$, is given by

$$\hat{\mathbf{x}}(k+1) = \tilde{\mathbf{x}}(k+1) + \mathbf{K}(k+1)\left[\bar{\mathbf{z}}(k+1) - \bar{\mathbf{h}}(\tilde{\mathbf{x}}[k+1])\right]. \tag{4.9}$$

The Kalman gain $\mathbf{K}(k+1)$ is given by

$$\mathbf{K}(k+1) = \mathbf{\Sigma}(k+1)\mathbf{H}^{\mathrm{T}}(k+1)\mathbf{C}_n^{-1}, \tag{4.10}$$

$\mathbf{\Sigma}(k+1)$ is the *a posteriori* estimation error covariance matrix given by

$$\mathbf{\Sigma}(k+1) = \left[\mathbf{H}^{\mathrm{T}}(k+1)\mathbf{C}_n^{-1}\mathbf{H}(k+1) + \mathbf{M}^{-1}(k+1)\right]^{-1}, \tag{4.11}$$

and $\mathbf{M}(k+1)$ is the *a priori* estimation error covariance matrix given by

$$\mathbf{M}(k+1) = \mathbf{F}(k)\mathbf{\Sigma}(k)\mathbf{F}^{\mathrm{T}}(k) + \mathbf{C}_w. \tag{4.12}$$

In the above expressions, $\bar{\mathbf{H}}(k+1)$ is the measurement Jacobian, $\mathbf{C}_n = \mathbb{E}\left[\bar{\mathbf{n}}(k)\bar{\mathbf{n}}^{\mathrm{T}}(k)\right]$ and $\mathbf{C}_w = \mathbb{E}\left[\mathbf{w}(k)\mathbf{w}^{\mathrm{T}}(k)\right]$.

The measurement Jacobian $\bar{\mathbf{H}}(k+1)$ can be written as

$$\bar{\mathbf{H}}(k+1) = \begin{bmatrix} \frac{\partial}{\partial \mathbf{x}}\mathbf{h}\{\mathbf{x}\} \\ \mathbf{A}\frac{\partial}{\partial \mathbf{x}}\mathbf{d}\{\mathbf{x}\} \end{bmatrix}_{\mathbf{x}=\tilde{\mathbf{x}}(k+1)}. \tag{4.13}$$

Now, for an N bus system, assuming there are L conventional measurements,

$$\frac{\partial}{\partial \mathbf{x}}\mathbf{h}\{\mathbf{x}\} = \mathbf{H} = \begin{bmatrix} \frac{\partial h_1\{\mathbf{x}\}}{\partial \theta_2} & \cdots & \frac{\partial h_1\{\mathbf{x}\}}{\partial \theta_N} & \frac{\partial h_1\{\mathbf{x}\}}{\partial |V_1|} & \cdots & \frac{\partial h_1\{\mathbf{x}\}}{\partial |V_N|} \\ \frac{\partial h_2\{\mathbf{x}\}}{\partial \theta_2} & \cdots & \frac{\partial h_2\{\mathbf{x}\}}{\partial \theta_N} & \frac{\partial h_2\{\mathbf{x}\}}{\partial |V_1|} & \cdots & \frac{\partial h_2\{\mathbf{x}\}}{\partial |V_N|} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_L\{\mathbf{x}\}}{\partial \theta_2} & \cdots & \frac{\partial h_L\{\mathbf{x}\}}{\partial \theta_N} & \frac{\partial h_L\{\mathbf{x}\}}{\partial |V_1|} & \cdots & \frac{\partial h_L\{\mathbf{x}\}}{\partial |V_N|} \end{bmatrix} \tag{4.14}$$

where $h_i\{\mathbf{x}\}$, $i = 1, 2, \ldots, L$, are the individual measurement functions for power-flow, power-injection or voltage-magnitude measurements and they are dependent on the placement of PMUs in the system under consideration. The above Jacobian, when evaluated at $\mathbf{x} = \tilde{\mathbf{x}}(k+1)$, is denoted by $\mathbf{H}(k+1)$.

Let us now consider the function $\mathbf{d}\{\mathbf{x}\}$:

$$\mathbf{v} = \mathbf{d}\{\mathbf{x}\} = \begin{bmatrix} |V_1| \\ |V_2|\cos\theta_2 \\ \vdots \\ |V_N|\cos\theta_N \\ 0 \\ |V_2|\sin\theta_2 \\ \vdots \\ |V_N|\sin\theta_N \end{bmatrix}. \tag{4.15}$$

It must be noted that the first bus is the reference bus and therefore, $\theta_1 = 0$.

We can form the following expression for the Jacobian of $\mathbf{d}\{\mathbf{x}\}$ denoted by $\mathbf{\Delta}$:

$$\mathbf{\Delta} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cos\theta_2 & 0 & \cdots & 0 & -|V_2|\sin\theta_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \cos\theta_N & 0 & 0 & \cdots & -|V_N|\sin\theta_N \\ 0 & \sin\theta_2 & 0 & \cdots & 0 & -|V_2|\cos\theta_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sin\theta_N & 0 & 0 & \cdots & -|V_N|\cos\theta_N \end{bmatrix}. \tag{4.16}$$

Since the voltage magnitudes at the buses are all approximately 1 p.u. [5], $\mathbf{\Delta}$ may be approximated by a unitary matrix

$$\mathbf{\Delta} \approx \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cos\theta_2 & 0 & \cdots & 0 & -\sin\theta_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \cos\theta_N & 0 & 0 & \cdots & -\sin\theta_N \\ 0 & \sin\theta_2 & 0 & \cdots & 0 & \cos\theta_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sin\theta_N & 0 & 0 & \cdots & \cos\theta_N \end{bmatrix}. \tag{4.17}$$

When evaluated at $\mathbf{x} = \tilde{\mathbf{x}}(k+1)$, $\mathbf{\Delta}$ is denoted as $\mathbf{\Delta}(k+1)$. This speeds up computation considerably since it may be implemented using a fast algorithm like CORDIC [39].

Therefore, from (4.14) and (4.17), we have

$$\bar{\mathbf{H}}(k+1) = \begin{bmatrix} \mathbf{H}(k+1) \\ \mathbf{A}\mathbf{\Delta}(k+1) \end{bmatrix} \tag{4.18}$$

4.2 Reduced Order EKF State Estimator

The reduced order RO/EKF state estimator presented in this section overcomes many of the drawbacks of MM/EKF state estimator. The problem is solved in two steps where PMU measurements and conventional measurements are handled separately, thereby reducing the dimensions of the matrices and vectors involved. For the same reason, there is also no possibility of having an ill-conditioned measurement error covariance matrix which leads to improved stability. Furthermore, it is an optimal estimator, i.e. the PMU measurements and the conventional measurements are combined in an optimal fashion as shown in [38].

In deriving the RO/EKF state estimator, we use the Cartesian form of the state vector. The state transition equation is given by

$$\mathbf{v}(k+1) = \mathbf{F}(k)\mathbf{v}(k) + \mathbf{g}(k) + \mathbf{w}(k) \tag{4.19}$$

where $\mathbf{F}(k)$ is the state transition matrix, the vector $\mathbf{g}(k)$ is calculated as outlined in the previous section and $\mathbf{w}(k)$ is the state transition noise.

The measurement equation in (4.2) is modified and written as

$$\begin{bmatrix} \mathbf{z}_1(k) \\ \mathbf{z}_2(k) \end{bmatrix} = \begin{bmatrix} \mathbf{r}\{\mathbf{v}(k)\} \\ \mathbf{A}\mathbf{v}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{n}_1(k) \\ \mathbf{n}_2(k) \end{bmatrix}, \tag{4.20}$$

where $\mathbf{r}\{\mathbf{v}(k)\}$ is a set of non-linear functions which relate the conventional measurements in $\mathbf{z}_1(k)$ to the state vector $\mathbf{v}(k)$. There exists a one-to-one mapping $f : \mathbb{R} \to \mathbb{R}$ between the state vector in polar co-ordinates to the state vector in rectangular co-ordinates given by

$$\mathbf{x}(k) = \mathbf{f}\{\mathbf{v}(k)\}. \tag{4.21}$$

Therefore, $\mathbf{r}\{\mathbf{v}(k)\}$ may be written as

$$\mathbf{r}\{\mathbf{v}(k)\} = \mathbf{h}\left(\mathbf{f}\{\mathbf{v}(k)\}\right) \tag{4.22}$$

We then proceed by linearising (4.20) about an arbitrary point $\mathbf{v}_0$. In particular, (4.20) by its first order Taylor series expansion

$$\begin{bmatrix}\mathbf{z}_1(k)\\ \mathbf{z}_2(k)\end{bmatrix} \approx \begin{bmatrix}\mathbf{r}(\mathbf{v}_0) + \mathbf{R}(\mathbf{v}(k) - \mathbf{v}_0)\\ \mathbf{A}\mathbf{v}(k)\end{bmatrix} + \begin{bmatrix}\mathbf{n}_1(k)\\ \mathbf{n}_2(k)\end{bmatrix}, \tag{4.23}$$

where $\mathbf{R}$ is the Jacobian of the modified measurement function $\mathbf{r}(\mathbf{v})$ evaluated at the point of linearisation $\mathbf{v}_0$ and is given by

$$\mathbf{R} = \frac{\partial \mathbf{r}(\mathbf{v})}{\partial \mathbf{v}}, \tag{4.24}$$

which upon application of the well-known chain rule in differential calculus yields

$$\begin{aligned}\mathbf{R} &= \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{f}(\mathbf{v})}{\partial \mathbf{v}}\\ &= \mathbf{H} \cdot \frac{\partial \mathbf{f}(\mathbf{v})}{\partial \mathbf{v}}.\end{aligned} \tag{4.25}$$

since $\mathbf{x} = \mathbf{f}(\mathbf{v})$.

It is now necessary to find the partial derivative of $\mathbf{f}(\mathbf{v})$ with respect to $\mathbf{v}$. The function $\mathbf{f}(\mathbf{v})$, for an N bus system is written as

$$\mathbf{x} = \mathbf{f}(\mathbf{v}) = \begin{bmatrix} \tan^{-1}\left(\frac{\Im\{V_2\}}{\Re\{V_2\}}\right)\\ \vdots \\ \tan^{-1}\left(\frac{\Im\{V_N\}}{\Re\{V_N\}}\right)\\ \sqrt{(\Re\{V_1\})^2 + 0}\\ \sqrt{(\Re\{V_2\})^2 + (\Im\{V_2\})^2}\\ \vdots \\ \sqrt{(\Re\{V_N\})^2 + (\Im\{V_N\})^2}\end{bmatrix}. \tag{4.26}$$

The Jacobian $\mathbf{\Phi} = \frac{\partial \mathbf{f}(\mathbf{v})}{\partial \mathbf{v}}$ is derived in a manner similar to $\mathbf{\Delta}$, and assuming $|V_i| = 1$ for $i = 1, \ldots, N$, it approximated by

$$\mathbf{\Phi} \approx \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0\\ 0 & -\Im\{V_2\} & 0 & \cdots & 0 & \Re\{V_2\} & 0 & \cdots & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots\\ 1 & 0 & 0 & \cdots & -\Im\{V_N\} & 0 & 0 & \cdots & \Re\{V_N\}\\ 0 & \Re\{V_2\} & 0 & \cdots & 0 & \Im\{V_2\} & 0 & \cdots & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & \cdots & \Re\{V_N\} & 0 & 0 & \cdots & \Im\{V_N\}\end{bmatrix}. \tag{4.27}$$

From (4.25) and (4.27), we have

$$\mathbf{R} = \mathbf{H} \cdot \mathbf{\Phi} \tag{4.28}$$

Moving the terms independent of $\mathbf{v}(k)$ to the left hand side we have

$$\begin{bmatrix} \mathbf{z}_1(k) - \mathbf{r}(\mathbf{v}_0) + \mathbf{R}\mathbf{v}_0 \\ \mathbf{z}_2(k) \end{bmatrix} = \begin{bmatrix} \mathbf{R} \\ \mathbf{A} \end{bmatrix} \mathbf{v}(k) + \begin{bmatrix} \mathbf{n}_1(k) \\ \mathbf{n}_2(k) \end{bmatrix} \tag{4.29}$$

which may now be written as

$$\begin{aligned} \begin{bmatrix} \mathbf{z'}_1(k) \\ \mathbf{z}_2(k) \end{bmatrix} &= \begin{bmatrix} \mathbf{R} \\ \mathbf{A} \end{bmatrix} \mathbf{v}(k) + \begin{bmatrix} \mathbf{n}_1(k) \\ \mathbf{n}_2(k) \end{bmatrix} \\ &= \mathbf{J}\mathbf{v}(k) + \begin{bmatrix} \mathbf{n}_1(k) \\ \mathbf{n}_2(k) \end{bmatrix}. \end{aligned} \tag{4.30}$$

Next, we attempt to partition $\mathbf{v}(k)$ into $\mathbf{v}_1(k)$ and $\mathbf{v}_2(k)$ where $\mathbf{v}_2(k)$ are the states which are observable by the PMUs and $\mathbf{v}_1(k)$ are the remaining states. This is achieved by left-multiplying $\mathbf{v}(k)$ by a simple permutation matrix $\mathbf{\Pi}$. Since $\mathbf{\Pi}$ is an orthogonal matrix we may write (4.30) as

$$\begin{bmatrix} \mathbf{z'}_1(k) \\ \mathbf{z}_2(k) \end{bmatrix} = \mathbf{J}\mathbf{\Pi}^T \begin{bmatrix} \mathbf{v}_1(k) \\ \mathbf{v}_2(k) \end{bmatrix} + \begin{bmatrix} \mathbf{n}_1(k) \\ \mathbf{n}_2(k) \end{bmatrix}. \tag{4.31}$$

where

$$\tilde{\mathbf{J}} = \mathbf{J}\mathbf{\Pi}^T = \begin{bmatrix} \tilde{\mathbf{J}}_{11} & \tilde{\mathbf{J}}_{12} \\ \mathbf{0} & \tilde{\mathbf{J}}_{22} \end{bmatrix}. \tag{4.32}$$

The matrix $\tilde{\mathbf{J}}_{21}$ is empty because clearly, PMU measurements do not depend on states not observable by PMUs.

The measurement equation now becomes

$$\begin{bmatrix} \mathbf{z}'_1(k) \\ \mathbf{z}_2(k) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{J}}_{11} & \tilde{\mathbf{J}}_{12} \\ \mathbf{0} & \tilde{\mathbf{J}}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{v}_1(k) \\ \mathbf{v}_2(k) \end{bmatrix} + \begin{bmatrix} \mathbf{n}_1(k) \\ \mathbf{n}_2(k) \end{bmatrix}. \tag{4.33}$$

From the above equation, it is possible to calculate $\mathbf{v}_2(k)$ by using a linear weighted least squares estimator operating on the measurement set $\mathbf{z}_2(k)$. The estimator for the PMU observable states is given by

$$\hat{\mathbf{v}}_2(k+1) = (\tilde{\mathbf{J}}_{22}^T \mathbf{C}_2^{-1} \tilde{\mathbf{J}}_{22})^{-1} \tilde{\mathbf{J}}_{22}^T \mathbf{C}_2^{-1} \mathbf{z}_2(k). \tag{4.34}$$

A state transition model for the partitioned state vector is derived starting from (2.22). By multiplying both sides with the permutation matrix $\mathbf{\Pi}$, we have

$$\begin{aligned} \mathbf{\Pi}\mathbf{v}(k+1) &= \mathbf{\Pi}\mathbf{F}(k)\mathbf{v}(k) + \mathbf{\Pi}\mathbf{g}(k) + \mathbf{\Pi}\mathbf{w}(k) \\ \begin{bmatrix} \mathbf{v}_1(k+1) \\ \mathbf{v}_2(k+1) \end{bmatrix} &= \left(\mathbf{\Pi}\mathbf{F}(k)\mathbf{\Pi}^T\right)(\mathbf{\Pi}\mathbf{v}(k)) + \begin{bmatrix} \mathbf{\Pi}_1 \\ \mathbf{\Pi}_2 \end{bmatrix} \mathbf{g}(k) + \begin{bmatrix} \mathbf{w}_1(k) \\ \mathbf{w}_2(k) \end{bmatrix} \\ &= \mathbf{F}'(k) \begin{bmatrix} \mathbf{v}_1(k) \\ \mathbf{v}_2(k) \end{bmatrix} + \begin{bmatrix} \mathbf{g}_1(k) \\ \mathbf{g}_2(k) \end{bmatrix} + \begin{bmatrix} \mathbf{w}_1(k) \\ \mathbf{w}_2(k) \end{bmatrix} \\ \begin{bmatrix} \mathbf{v}_1(k+1) \\ \mathbf{v}_2(k+1) \end{bmatrix} &= \begin{bmatrix} \mathbf{F}'_{11}(k) & \mathbf{F}'_{12}(k) \\ \mathbf{F}'_{21}(k) & \mathbf{F}'_{22}(k) \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(k) \\ \mathbf{x}_2(k) \end{bmatrix} + \begin{bmatrix} \mathbf{g}_1(k) \\ \mathbf{g}_2(k) \end{bmatrix} + \begin{bmatrix} \mathbf{w}_1(k) \\ \mathbf{w}_2(k) \end{bmatrix}. \end{aligned}$$

From the above expression, the state transition equations for each of the partitions of the state vector are given by

$$\mathbf{v}_1(k+1) = \mathbf{F}'_{11}(k)\mathbf{v}_1(k) + \mathbf{F}'_{12}(k)\mathbf{v}_2(k) + \mathbf{g}_1(k) + \mathbf{w}_1(k) \tag{4.35}$$

and

$$\mathbf{v}_2(k+1) = \mathbf{F}'_{21}(k)\mathbf{v}_1(k) + \mathbf{F}'_{22}(k)\mathbf{v}_2(k) + \mathbf{g}_2(k) + \mathbf{w}_2(k). \tag{4.36}$$

Now, we define a new state variable $\mathbf{s}(k)$ which is essentially a combination of the PMU-observable states and the non-PMU-observable states. The combination is performed using a combiner matrix $\mathbf{P}(k)$. The new state vector is given by

$$\mathbf{s}(k) \triangleq \mathbf{v}_1(k) - \mathbf{P}(k)\mathbf{v}_2(k). \tag{4.37}$$

Finally, we have at a new pair of equations describing state transition and measurement models for the state estimation problem

$$\mathbf{s}(k+1) = \mathbf{A}(k)\mathbf{s}(k) + \mathbf{B}(k)\hat{\mathbf{v}}_2(k) + \mathbf{D}(k)\left(\mathbf{g}(k) + \mathbf{w}(k)\right) \tag{4.38}$$

$$\mathbf{z}'_1(k) = \tilde{\mathbf{J}}_{11}\mathbf{s}(k) + \mathbf{C}(k)\hat{\mathbf{v}}_2(k) + \mathbf{n}_1(k) \tag{4.39}$$

where

$$\begin{aligned}
\mathbf{A}(k) &= \mathbf{F}'_{11}(k) - \mathbf{P}(k+1)\mathbf{F}'_{21}(k) \\
\mathbf{B}(k) &= \mathbf{F}'_{12}(k) - \mathbf{P}(k+1)\mathbf{F}'_{22}(k) + \mathbf{A}(k)\mathbf{P}(k) \\
\mathbf{C}(k) &= \tilde{\mathbf{J}}_{12} + \tilde{\mathbf{J}}_{11}\mathbf{P}(k) \\
\mathbf{D}(k) &= \mathbf{\Pi}_1 - \mathbf{P}(k)\mathbf{\Pi}_2
\end{aligned}$$

4.2.1 The estimation algorithm

The function of the estimator is to arrive at an estimate of the system state vector $\hat{\mathbf{v}}(k)$ from the conventional measurements $\mathbf{z}_1(k)$ and the PMU measurements $\mathbf{z}_2(k)$.

The signal processing steps proceed as follows:

1. *State Forecasting*: The state is forecast for the time instant $k+1$ using a suitable forecasting algorithm. The forecast is denoted by $\tilde{\mathbf{v}}(k+1)$:

$$\tilde{\mathbf{v}}(k+1) = \mathbf{F}(k)\mathbf{v}(k) + \mathbf{g}(k) \tag{4.40}$$

2. *Preprocessing the Measurements*: In this stage, the measurements $\mathbf{z}_1(k)$ are transformed into $\mathbf{z}'_1(k)$ by subtracting the quantities specified in (4.29). The point of linearisation is assumed to be the forecast $\tilde{\mathbf{v}}(k+1)$.

3. *Estimating the PMU states*: The estimate the PMU-observable state vector, denoted by $\hat{\mathbf{v}}_2(k+1)$, is simply calculated by evaluating (4.34).

4. *Kalman Filtering*: In order to make prediction of $\mathbf{s}(k+1)$, say $\tilde{\mathbf{s}}(k+1)$, we need the combiner matrix $\mathbf{P}(k)$ given by

$$\mathbf{P}(k) = \mathbf{\Lambda}_2(k)\mathbf{\Lambda}_1^{-1}(k) \tag{4.41}$$

where

$$\mathbf{\Lambda}_2(k) = \mathbf{F}_{11}(k)\mathbf{\Sigma}(k)\mathbf{F}_{21}^T(k)$$
$$\mathbf{\Lambda}_1(k) = \mathbf{F}'_{21}(k)\mathbf{\Sigma}(k)[\mathbf{F}'_{21}(k)]^T$$

and $\mathbf{\Sigma}(k)$ is the *a posteriori* estimation error covariance matrix of the modified state vector $\hat{\mathbf{s}}(k)$. Now

$$\tilde{\mathbf{s}}(k+1) = \mathbf{A}(k)\hat{\mathbf{s}}(k) + \mathbf{B}(k)\hat{\mathbf{v}}_2(k) + \mathbf{D}(k)\mathbf{g}(k). \tag{4.42}$$

The prediction error covariance matrix is computed next and is given by

$$\mathbf{M}(k+1) = \mathbf{A}(k)\mathbf{\Sigma}(k)\mathbf{A}^T(k) + (\mathbf{I} - \mathbf{P}(k))\mathbf{C}_{w_1}(\mathbf{I} - \mathbf{P}(k))^T \tag{4.43}$$

where $\mathbf{C}_{w_1}$ is the covariance matrix of the state transition uncertainty about those states which are not PMU-observable.

The Kalman gain is computed next and is given by

$$\mathbf{K}(k+1) = \mathbf{M}(k+1)\tilde{\mathbf{J}}_{11}^T(\tilde{\mathbf{J}}_{11}\mathbf{M}(k+1)\tilde{\mathbf{J}}_{11}^T + \mathbf{C}_1)^{-1}. \tag{4.44}$$

The estimate of the modified state vector $\mathbf{s}(k)$ is given by

$$\hat{\mathbf{s}}(k+1) = \tilde{\mathbf{s}}(k+1) + \mathbf{K}(k+1)\{\mathbf{z}'_1(k) - \tilde{\mathbf{J}}_{11}\tilde{\mathbf{s}}(k+1) - \mathbf{C}(k)\hat{\mathbf{v}}_2(k+1)\}. \tag{4.45}$$

The error covariance matrix is updated for the next stage and is given by

$$\mathbf{\Sigma}(k+1) = (\mathbf{I} - \mathbf{K}(k+1)\tilde{\mathbf{J}}_{11})\mathbf{M}(k+1)(\mathbf{I} - \mathbf{K}(k)\tilde{\mathbf{J}}_{11})^T + \mathbf{K}(k+1)\mathbf{C}_1\mathbf{K}^T(k+1). \tag{4.46}$$

5. *Recovery of the Original State Vector*: The original state vector $\hat{\mathbf{v}}(k)$ is computed by evaluating the following expressions:

$$\hat{\mathbf{v}}_1(k+1) = \hat{\mathbf{s}}(k+1) + \mathbf{P}(k)\hat{\mathbf{v}}_2(k+1) \tag{4.47}$$

and

$$\hat{\mathbf{v}}(k+1) = \mathbf{\Pi}^T \begin{bmatrix} \hat{\mathbf{v}}_1(k+1) \\ \hat{\mathbf{v}}_2(k+1) \end{bmatrix}. \tag{4.48}$$

The expressions derived above are for a generic case and do not take into account any special characteristics of the state transition model. The FASE realm is short of a complete analysis of building adequate state evolution models, and mostly state space equations are simply taken without a rigorous justification. For example, almost all FASE algorithms proposed so far assume that there is no correlation

between state variables, making the state transition matrix $\mathbf{F}(k)$ simply diagonal [20]. In a real-world scenario, this is rarely the case as there is a certain amount of correlation between buses in the same neighbourhood because any change in the power demand on a load bus, or the power output of a generator bus affects the power flowing in the neighbourhood of that bus. More accurate dynamic models are therefore needed to incorporate coupling between state variables. However, here we examine the RO/EKF algorithm when a diagonal state transition matrix is used since developing new state transition models is beyond the scope of this thesis.

When $\mathbf{F}(k)$ is diagonal, the matrices $\mathbf{F}_{12}$ and $\mathbf{F}_{21}$ become zero matrices and this simplifies the implementation considerably. From (4.41), we see that when $\mathbf{F}_{21} = \mathbf{0}$, the combiner matrix becomes zero and there is a decoupling of the estimation problem into a linear estimator and a Kalman filter and the modified state vector reduces to $\mathbf{v}_1(k)$. Furthermore, this is the optimal state estimator since this choice of combiner matrix minimizes the trace of the *a posteriori* error covariance matrix [38].

In the light of the above, the prediction equation becomes

$$\tilde{\mathbf{v}}_1(k+1) = \mathbf{F}'_{11}(k)\hat{\mathbf{v}}_1(k) + \mathbf{g}_1(k) \tag{4.49}$$

while the *a priori* error covariance matrix is given by

$$\mathbf{M}(k+1) = \mathbf{F}'_{11}(k)\mathbf{\Sigma}(k)[\mathbf{F}'_{11}(k)]^T + \mathbf{C}_{w_1}. \tag{4.50}$$

The expressions for the Kalman gain and the *a posteriori* error covariance matrix remain the same as (4.44) and (4.46), respectively.

A more computationally efficient version of the above expressions may be derived by applying the matrix inversion lemma to (4.44) and calculating the *a posteriori* error covariance matrix before the Kalman gain. Upon doing this, we arrive at the following update expressions for the state vector associated with PMU-unobservable states:

$$\hat{\mathbf{v}}_1(k+1) = \tilde{\mathbf{v}}_1(k+1) + \mathbf{K}(k+1)\{\mathbf{z}'_1(k) - \tilde{\mathbf{J}}_{11}\tilde{\mathbf{v}}_1(k+1) - \mathbf{J}_{12}(k)\hat{\mathbf{v}}_2(k+1)\} \tag{4.51}$$

where

$$\mathbf{K}(k+1) = \mathbf{\Sigma}(k+1)\mathbf{J}_{11}^T\mathbf{C}_1^{-1} \tag{4.52}$$

$$\mathbf{\Sigma}(k+1) = (\mathbf{M}^{-1}(k+1) + \mathbf{J}_{11}^T\mathbf{C}_1^{-1}\mathbf{J}_{11})^{-1}. \tag{4.53}$$

4.3 Simulation Results

In this section, simulation results are presented for MM/EKF and the RO/EKF algorithms. First, we begin by going over the details of the simulation setup. Secondly, the tracking capabilities of MM/EKF and RO/EKF are verified. Thereafter, we analyse the error performance for both algorithms as the PMU observability increases. We also compare the performance of the MM/EKF and the RO/EKF algorithms. Next, we discuss certain aspects of the implementation of the two algorithms presented here with regard to computational complexity. Finally, a brief discussion of the advantages and disadvantages of both algorithms is presented in the light of the simulation results.

4.3.1 Simulation setup

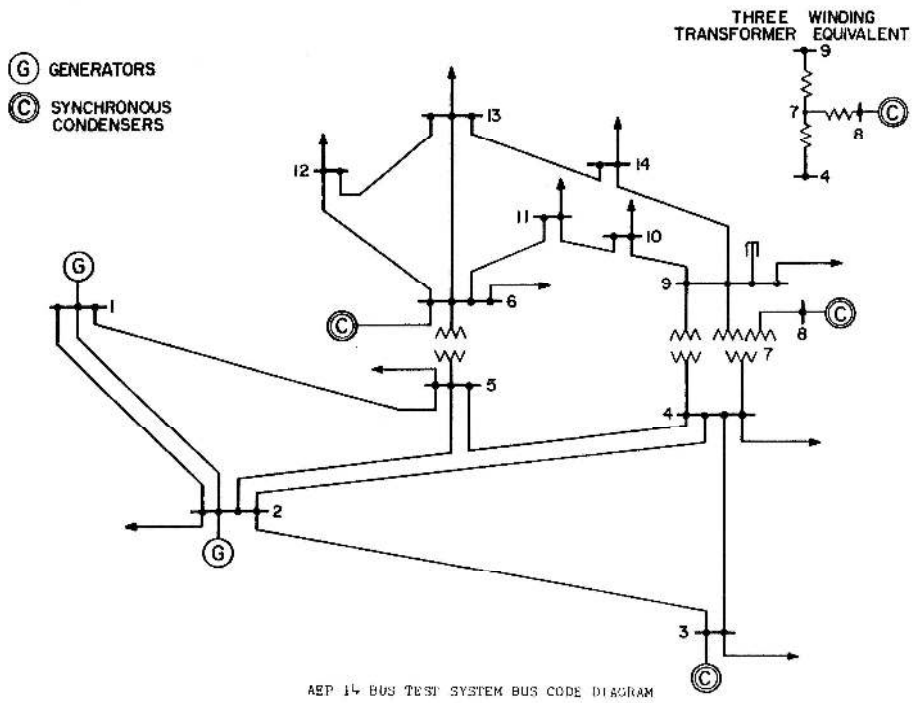


Figure 4.1: Single line diagram of the IEEE 14 bus test system [40] showing tie-lines, generators and loads.

The two algorithms proposed are simulated on the standard IEEE 14 bus system to verify their ability to track the evolution of the power system state. The IEEE 14 Bus Test Case represents a portion of the American Electric Power System (in the Midwestern US) as of February, 1962 and has been used as a standard test-bed for many state estimation algorithms in the literature. The interconnection of the IEEE 14 bus system is shown in Figure 4.1 [40] while the placement of different kinds of conventional measurements is shown in Figure 4.2.

It is assumed that three previous state vectors were known with negligible uncertainty in order to initialize the forecasting algorithm. The Kalman filters, however, have no knowledge of any previous state vectors and assume a *flat start* condition, i.e. the phase angles on all buses is assumed to be zero while the voltage magnitude is assumed to be 1 p.u. Similarly, in the case of the state vector in Cartesian form, it is assumed that the imaginary part of the voltage phasor on all buses is zero while the real part is 1 p.u.

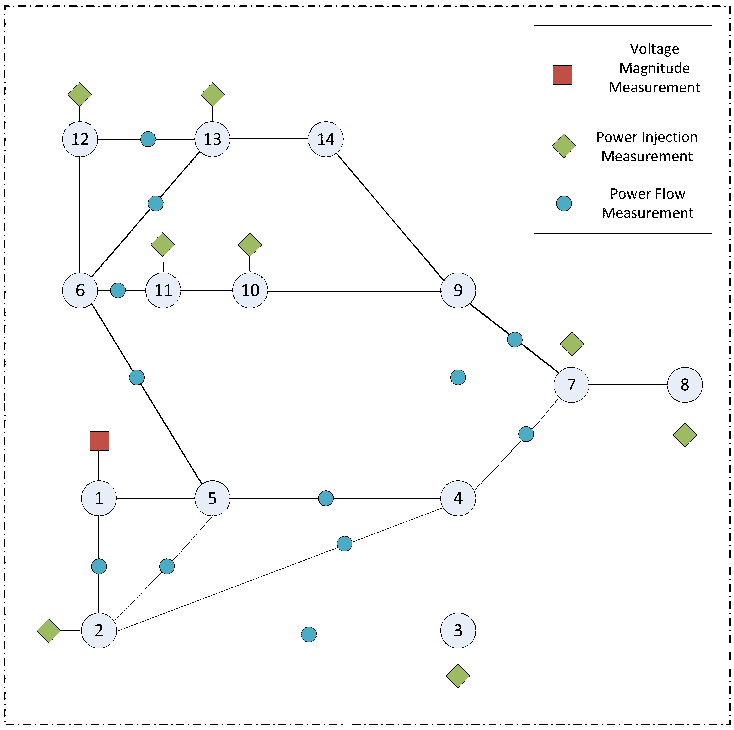


Figure 4.2: The IEEE 14 bus interconnection with the placement of conventional power and voltage magnitude measurements.

In order to simulate the time-domain state evolution, the system is perturbed by increasing the power demand on the load buses by a nominal value at every time instant. The simulation is performed over 200 time instants. A random, zero-mean, normally distributed jitter with a variance of 0.01 is added to the power demand on each bus at every time instant to make the simulation more realistic. Next, a Gauss-Newton load flow calculation is performed using the MatPower toolbox in MATLAB [41] to update line flows, voltage magnitudes and phase angles throughout the system. The vector of voltage magnitudes and phase angles at each time instant represents the true state of the system. The conventional and PMU measurements are generated using (2.11) and (3.12), respectively. The standard deviations of the PMU readings is given in Table 4.1 and the measurement noise associated with each conventional measurement is zero-mean Gaussian distributed with a standard deviation of 0.01.

Table 4.1: Standard deviations of PMU measurements.

Phasor Type	Avg. Magnitude Error (p.u.)	Avg. Phase Error (deg)
Voltage	0.006	0.52
Current	0.006	1.04

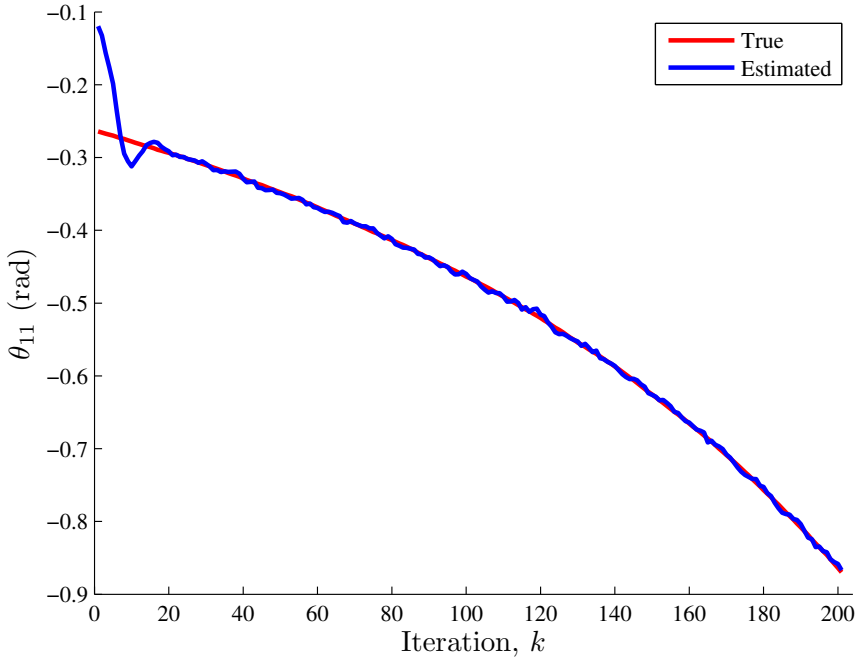
4.3.2 Tracking ability

The inspiration for the simulation scenario chosen here to study the tracking ability of the algorithms proposed here is drawn from the power outage that occurred throughout parts of the Northeastern and Midwestern United States and Ontario, Canada on Thursday, August 14, 2003, just before 4:10 p.m. EDT. It was the second most widespread blackout in history and affected an estimated 10 million people in Ontario and 45 million people in eight U.S. states. A subsequent report [42] by the U.S.-Canada power system outage task force addressed the need for improved situational awareness and time-synchronized phasor data. The motivation behind the algorithms discussed here is to just that; to use improved time-synchronized phasor data in order to have improved situational awareness. An early symptom of the impending blackout was a sharp dip in the phase angle of the Cleveland bus.

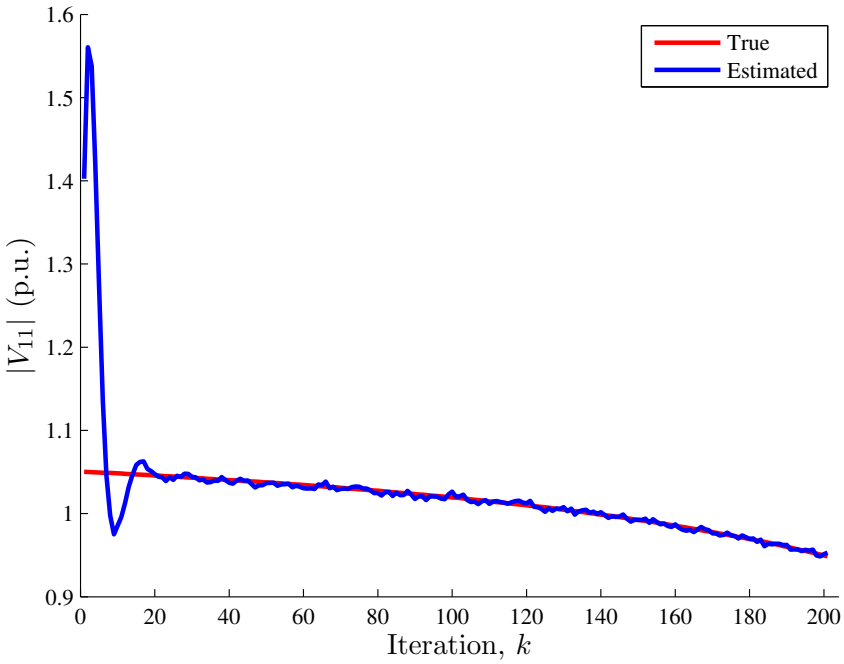
The scenario chosen to be simulated here is a catastrophic increase in power demand on all the load buses which causes the phase angle at load buses to dip sharply, a phenomenon indicative of a blackout. The power demand at each load bus is increased at the rate of 0.5% at every time instant which would overload the generators buses causing them to behave like load buses and also causing the phase angle to dip. An important criterion to verify the algorithms proposed here is to study the ability of the algorithms to track the state vector as it evolves in the simulation scenario.

There is only one PMU placed at bus 3 and the state of an arbitrary, PMU-unobservable busbar, bus 11, is plotted with time. Figures 4.3(a) and 4.3(b) show the ability of MM/EKF algorithm to track the true state vector. It can be seen from the plots that it takes around 22 time instants for the MM/EKF algorithm to converge. It is also clear from the figures that the MM/EKF state estimator is able to track the state vector.

Next, we study the ability of the resource-efficient, RO/EKF state estimator to track the state vector. Again, for the sake of consistency we assume that there is only one PMU and it is placed on bus 3. Looking at Figures 4.4(a) and 4.4(b), we can see that the RO/EKF algorithm is indeed able to track the state vector. Furthermore, it can also be seen that the algorithm converges faster than the MM/EKF algorithm.

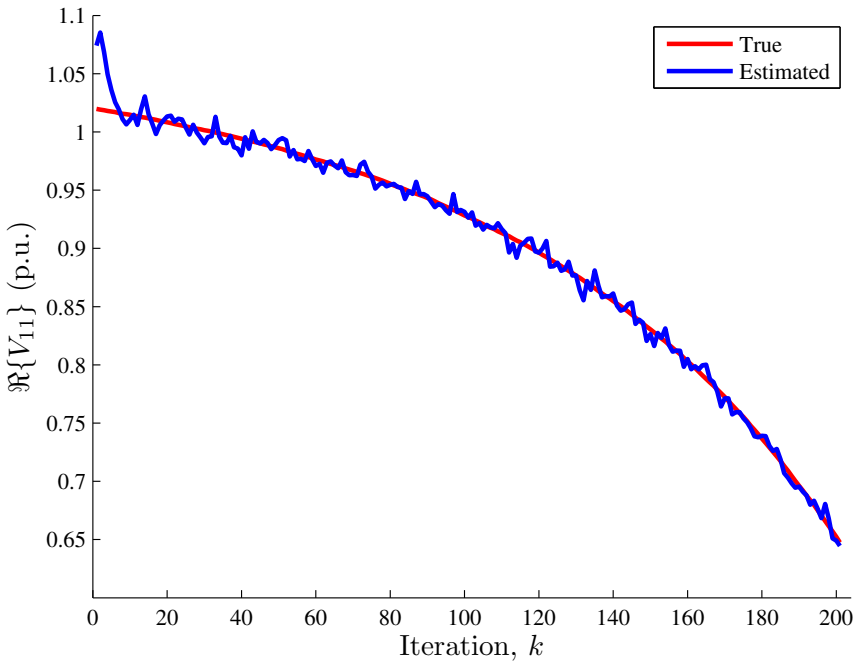


(a)

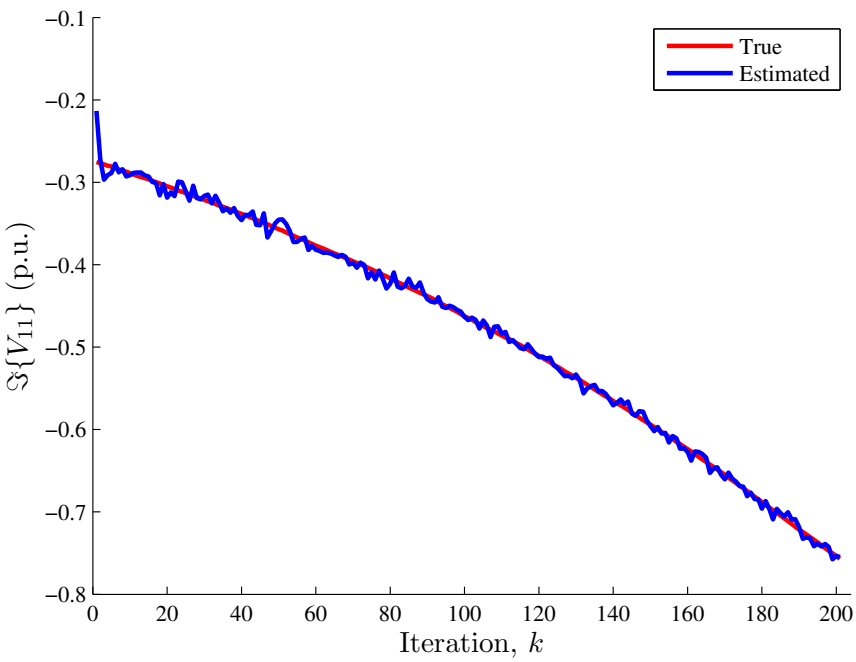


(b)

Figure 4.3: The tracking plot of the phase angle of the voltage phasor on bus 11 by the MM/EKF state estimator is shown in Figure 4.3(a) and Figure 4.3(b) shows the tracking plot of the magnitude of the voltage phasor on bus 11.



(a)



(b)

Figure 4.4: The tracking plot of the real part of the voltage phasor on bus 11 by the RO/EKF state estimator is shown in Figure 4.4(a) and Figure 4.4(b) shows the tracking plot of the imaginary part of the voltage phasor on bus 11.

4.3.3 Performance analysis

In order to compare the two algorithms introduced in this chapter we need a common performance measure. The performance measure here is chosen to be the root mean-square-error (RMSE) over 100 Monte Carlo runs. We begin by evaluating the effect of increasing the number of PMUs placed in the system on the error performance of both algorithms. Next, we compare the RMSE performance of the MM/EKF and the RO/EKF algorithms. Finally, we discuss the insights gained from the examination of the performance plots.

Figures 4.5(a) and 4.5(b) show the performance of the two FASE algorithms with regard to PMU observability. PMU observability denoted by O here is expressed as a percentage and is defined as the ratio of the number of PMU observable state variables in an N bus system to the total number of state variables, i.e.

$$O = \frac{\text{No. of PMU observable state variables}}{2N - 1} \times 100.$$

It can be seen from both error plots that as the number of state variables observable to PMUs increases the RMSE is reduced. This is because PMU measurements are of a much higher quality than conventional measurements and adding PMU measurements also increases redundancy and makes the state estimator more robust.

Now, we compare the performance of the MM/EKF algorithm and the RO/EKF algorithm. Figures 4.6(a) and 4.6(b) compare of the performance of the MM/EKF and RO/EKF FASE algorithms. It can be seen from the plot that the MM/EKF algorithm, being an integrated state estimator, has a better error performance than the RO/EKF algorithm. However, the RO/EKF has several other advantages over the MM/EKF estimator as discussed in the following section.

4.3.4 Discussion

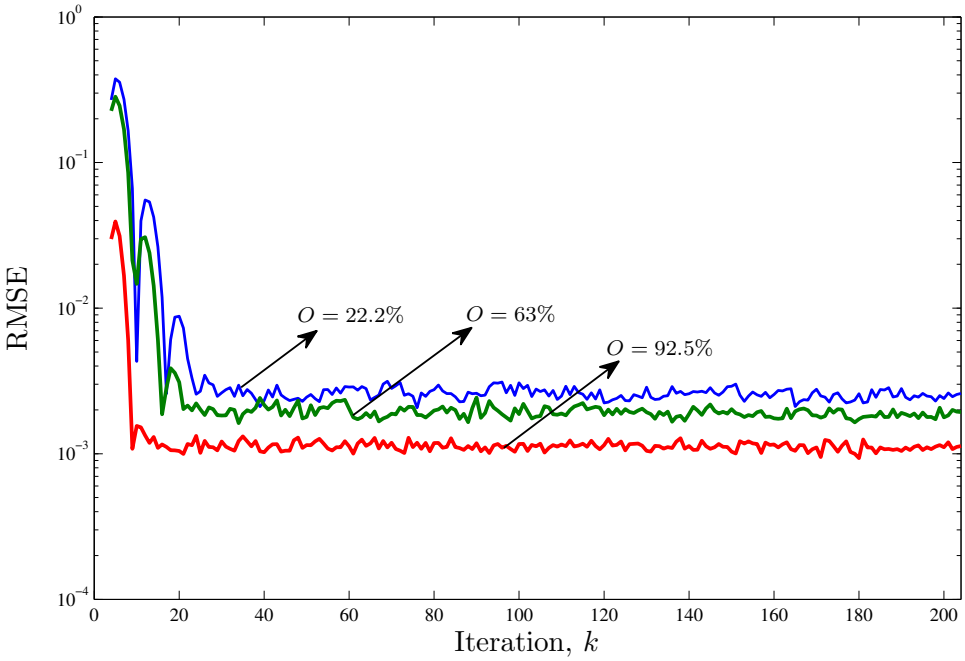
The main advantage of using an MM/EKF forecasting-aided state estimator is the high accuracy as well as a high level of redundancy offered by using multiple measurements of the same parameters. It also uses up less computational resources than the conventional weighted least-squares approach since there is only one iteration per time instant whereas the conventional state estimator iterates until convergence at every time instant. However, the high level of redundancy, while advantageous in terms of accuracy leads to a greater computational cost as the dimensions of the matrices and vectors involved become large. Clearly, a more resource-efficient state estimation method is needed to overcome the detrimental effects of the MM/EKF with regard to computational complexity. The RO/EKF algorithm accomplishes this by separating the PMU-observable states from the PMU-unobservable states and estimating them separately. However, the saving in computational complexity comes at the cost of accuracy.

The most important aspect of the RO/EKF algorithm is the resource-efficient nature of its implementation. Since the processing is done on matrices and vectors of reduced dimensions, the RO/EKF is also more computationally efficient than a

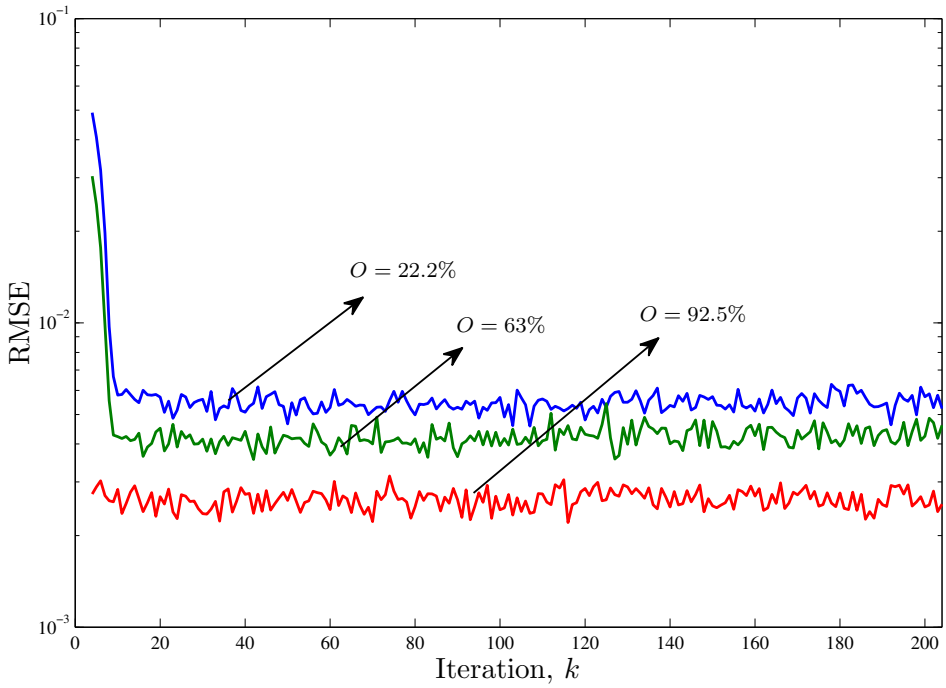
typical integrated approach. For example, consider a system with N buses. Assuming a fairly realistic 50% PMU observability, the dimensions of the largest matrix inversion is halved, when compared to the integrated state estimator. Furthermore, the inherent structure of the RO/EKF algorithm allows for some level of distributed parallel implementation of the state estimation algorithm. For example, the task of estimating the state of the PMU-observable states may be offloaded to the PDC or the super-PDC since it is computationally light and only the state estimate needs to be transmitted to the main EMS/SCADA. This means that PMU current measurements, which are more in number than voltage measurements need never be transmitted to the EMS/SCADA thereby freeing up communication resources. Moreover, the problem of ill-conditioned measurement noise covariance matrices is also avoided because the covariance matrix $\mathbf{C}_n$ is never inverted. This makes the algorithm very stable when compared to the MM/EKF.

PMU measurements follow a time scale different to conventional measurements. Because of this, the MM/EKF estimator must wait for the conventional measurements to arrive before calculating the state estimate. However, in the RO/EKF state estimator, the PMU-observable states are evaluated separately and these may be updated as often as necessary irrespective of whether the conventional measurements have arrived or not as seen from (4.51).

In summary, it is safe to say that while the MM/EKF algorithm is more accurate due to high redundancy, this redundancy also increases the computational burden. Even though the MM/EKF algorithm is more resource-efficient than the traditional weighted least-squares approach, the computational load may be further reduced by using the RO/EKF algorithm, particularly in cases where high accuracy is not required.

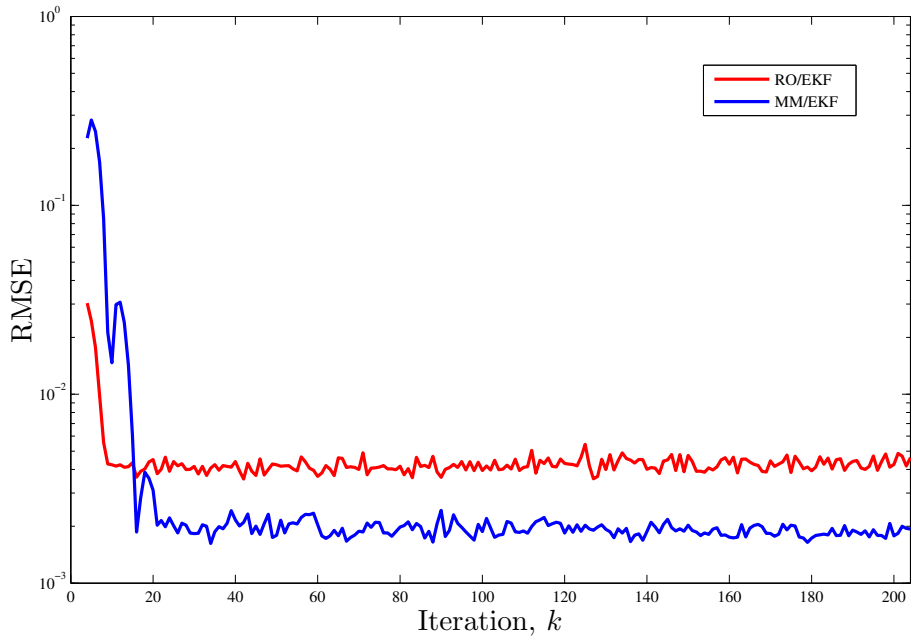


(a)

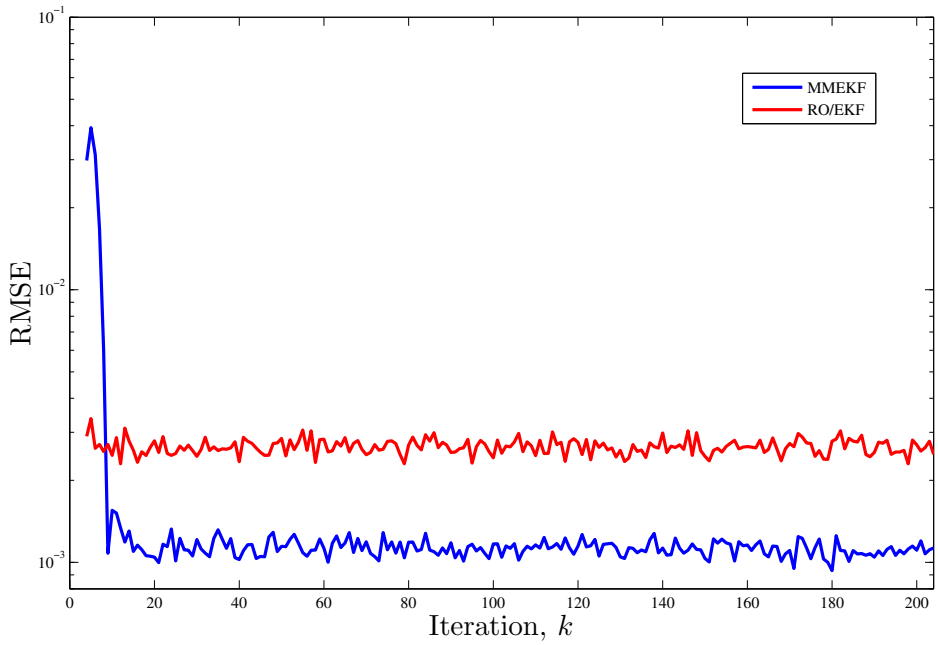


(b)

Figure 4.5: Figure 4.5(a) shows the RMSE of the MM/EKF algorithm with respect to time while Figure 4.5(b) shows the RMSE of the RO/EKF algorithm. The different plots represent the performance of the algorithm for different values of PMU observability O.



(a)



(b)

Figure 4.6: Figure 4.6(a) compares the RMSE of the MM/EKF and the RO/EKF algorithms with $O = 63\%$ and Figure 4.6(b) shows the performance comparison for $O = 92.5\%$

The MM/EKF algorithm is summarised in Algorithm 1 and RO/EKF in Algorithm 2 for an N bus system over K time instants. It is implicitly assumed that three past state vectors are available in order to get the forecasting algorithm started.

Algorithm 1 The Mixed-Measurement/Extended Kalman Filter

MM/EKF Algorithm

Initialise

$\mathbf{a}(0) = \mathbf{x}(-3); \mathbf{b}(0) = \frac{1}{2}\left(\{\mathbf{x}(-2) - \mathbf{x}(-1)\} + \{\mathbf{x}(-3) - \mathbf{x}(-2)\}\right);$

Choose α and β

$\tilde{\mathbf{x}}(0) = \hat{\mathbf{x}}(0) = [\underbrace{0, 0, \cdots, 0}_{N-1}, \underbrace{1, 1, \cdots, 1}_{N}]^T$

for $k = 1 \ldots K$

Forecast state vector:

$$\begin{aligned}
\mathbf{a}(k) &= \alpha\hat{\mathbf{x}}(k) + (1-\alpha)\tilde{\mathbf{x}}(k) \\
\mathbf{b}(k) &= \beta[\mathbf{a}(k) - \mathbf{a}(k-1)] + (1-\beta)\mathbf{b}(k-1) \\
\mathbf{g}(k) &= (1+\beta)(1-\alpha)\tilde{\mathbf{x}}(k) - \beta\mathbf{a}(k-1) + (1-\beta)\mathbf{b}(k-1) \\
\tilde{\mathbf{x}}(k) &= \mathbf{F}\hat{\mathbf{x}}(k-1) + \mathbf{g}(k), \mathbf{F} = \alpha(1-\beta)\mathbf{I}
\end{aligned}$$

Kalman Filtering:

1. Calculate Jacobian $\bar{\mathbf{H}}(k)$ and measurement function $\bar{\mathbf{h}}(\cdot)$ using $\tilde{\mathbf{x}}(k)$
2. Compute Kalman gain:

$$\begin{aligned}
\mathbf{M}(k) &= \mathbf{F}\boldsymbol{\Sigma}(k-1)\mathbf{F}^{\mathrm{T}} + \mathbf{C}_w \\
\boldsymbol{\Sigma}(k) &= \left[\bar{\mathbf{H}}^{\mathrm{T}}(k)\mathbf{C}_n^{-1}\bar{\mathbf{H}}(k) + \mathbf{M}^{-1}(k)\right]^{-1} \\
\mathbf{K}(k) &= \boldsymbol{\Sigma}(k)\bar{\mathbf{H}}^{\mathrm{T}}(k)\mathbf{C}_n^{-1}
\end{aligned}$$

3. Wait for both conventional and PMU measurements to arrive. Then form integrated measurement vector $\bar{\mathbf{z}}(k)$.
4. Calculate the state estimate:

$$\hat{\mathbf{x}}(k) = \tilde{\mathbf{x}}(k) + \mathbf{K}(k)\left[\bar{\mathbf{z}}(k) - \bar{\mathbf{h}}(\tilde{\mathbf{x}}[k])\right]$$

end

Algorithm 2 The Reduced Order/Extended Kalman Filter

RO/EKF Algorithm

Initialise

$\mathbf{a}(0) = \mathbf{x}(-3); \mathbf{b}(0) = \frac{1}{2}\left(\{\mathbf{v}(-2) - \mathbf{v}(-1)\} + \{\mathbf{v}(-3) - \mathbf{v}(-2)\}\right);$
Choose α and β
$\tilde{\mathbf{v}}(0) = \hat{\mathbf{v}}(0) = [\underbrace{1, 1, \cdots, 1}_{N}, \underbrace{0, 0, \cdots, 0}_{N-1}]^T$

for $k = 1 \ldots K$

Forecast state vector:

$$\begin{aligned}
\mathbf{a}(k) &= \alpha\hat{\mathbf{v}}(k) + (1-\alpha)\tilde{\mathbf{v}}(k) \\
\mathbf{b}(k) &= \beta_i[\mathbf{a}(k) - \mathbf{a}(k-1)] + (1-\beta_i)\mathbf{b}(k-1) \\
\mathbf{g}(k) &= (1+\beta)(1-\alpha)\tilde{\mathbf{v}}(k) - \beta\mathbf{a}(k-1) + (1-\beta)\mathbf{b}(k-1) \\
\tilde{\mathbf{v}}(k) &= \mathbf{F}\hat{\mathbf{v}}(k-1) + \mathbf{g}(k), \mathbf{F} = \alpha(1-\beta)\mathbf{I}
\end{aligned}$$

Estimate PMU-observable states:

Wait for PMU measurements $\mathbf{z}_2(k)$ to arrive. Update estimate $\hat{\mathbf{v}}_2(k)$:

$$\hat{\mathbf{v}}_2(k) = (\tilde{\mathbf{J}}_{22}^T\mathbf{C}_2^{-1}\tilde{\mathbf{J}}_{22})^{-1}\tilde{\mathbf{J}}_{22}^T\mathbf{C}_2^{-1}\mathbf{z}_2(k)$$

Kalman filtering:

1. Calculate matrices $\tilde{\mathbf{J}}_{11}(k)$ and $\tilde{\mathbf{J}}_{12}(k)$ using $\tilde{\mathbf{v}}(k)$ given by

$$\begin{bmatrix}\tilde{\mathbf{J}}_{11}(k) & \tilde{\mathbf{J}}_{12}(k)\end{bmatrix} = \mathbf{R}(k)\mathbf{\Pi}^T$$

2. Compute Kalman gain:

$$\begin{aligned}
\mathbf{M}(k) &= \mathbf{F}_{11}'\mathbf{\Sigma}(k-1)(\mathbf{F}_{11}')^T + \mathbf{C}_{w_1} \\
\mathbf{\Sigma}(k) &= (\mathbf{M}^{-1}(k) + \tilde{\mathbf{J}}_{11}^T\mathbf{C}_1^{-1}\tilde{\mathbf{J}}_{11})^{-1} \\
\mathbf{K}(k) &= \mathbf{\Sigma}(k)\tilde{\mathbf{J}}_{11}^T\mathbf{C}_1^{-1}
\end{aligned}$$

3. Wait for conventional measurements $\mathbf{z}_1(k)$ to arrive. Then calculate the state estimate:

$$\hat{\mathbf{v}}_1(k) = \tilde{\mathbf{v}}_1(k) + \mathbf{K}(k)\{\mathbf{z}_1'(k) - \tilde{\mathbf{J}}_{11}\tilde{\mathbf{v}}_1(k) - \tilde{\mathbf{J}}_{12}(k)\hat{\mathbf{v}}_2(k)\}$$

Form full state estimate:

$$\hat{\mathbf{v}}(k) = \mathbf{\Pi}^T\begin{bmatrix}\hat{\mathbf{v}}_1(k) \\ \hat{\mathbf{v}}_2(k)\end{bmatrix}$$

end

Chapter 5

Event-Triggered Multi-Area State Estimation

The power grid of the future will be equipped with a myriad of smart measurement devices which collect and transmit vast quantities of data. Consequently, the control center will need to process those data, convert the raw data into useful information and transform this information into actionable intelligence. In fact, the deployment of PMUs at the transmission level has already resulted in more data than the legacy grid's control center can handle [7]. When the smart grid is fully deployed, there is a risk that the grid's operator will be drowned in data. In designing the smart grid, engineers must be mindful in preventing the data tsunami. Furthermore, it is desirable to make the communication infrastructure throughout the grid energy- and bandwidth-efficient. Hence an event-triggered approach to sensing, communication and information processing would be quite appealing.

To reduce the tremendous computational complexity and large storage and data transfer requirements, multi-area state estimation (MASE) has been proposed as a viable alternative to the traditional centralized state estimation solutions. Figure 5.1 depicts a typical two-level MASE approach where the grid is divided into smaller areas, each of which is equipped with a local state estimator to provide a local state estimate. These are then aggregated in some way with other local solutions to render a global solution. Most of the existing MASE algorithms employ the weighted least-squares (WLS) method using traditional SCADA measurements, and they are typically implemented with a two-level architecture — a lower level that provides local SE solutions and a higher level that coordinates local estimates to obtain a global SE solution.

In the conventional MASE approach, the state estimates are updated and sent to the co-ordination level *at every time instant* irrespective of whether or not an update is warranted. In this chapter, we present an *event-triggered* MASE approach that updates the local state estimates only when needed. Accordingly, those estimates are transmitted to the co-ordination level only when such an action is *informative*, i.e. an on-demand estimation and communication approach. The triggering event is characterized by the estimation error exceeding a prescribed magnitude bound. The proposed approach will not only reduce computational complexity, but the

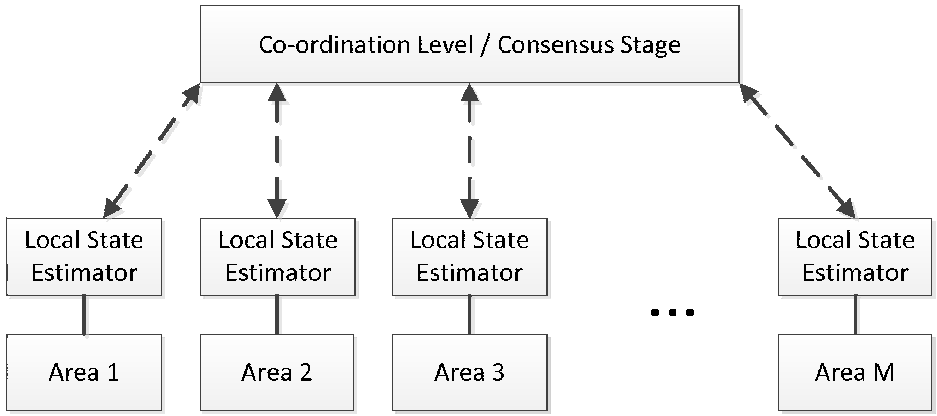


Figure 5.1: Schematic representation of two-level multi-area state estimation in a power system divided into M subsystems.

communication requirements in MASE.

In the event-triggered MASE framework, the local areas update their state estimates only when needed and cooperate (transmit the estimates) only when such an action is *informative*. An adaptive estimation paradigm, referred to as set-membership adaptive filtering (SMAF) [43], offers a viable solution to this approach. SMAF algorithms feature selective update of parameter estimates unlike conventional adaptive estimation algorithms such as recursive least-squares (RLS) and least-mean-squares (LMS) where parameter estimates are updated continually regardless of the benefits of such updates. In SMAF, estimates are updated only when the measurements offer sufficient *innovation*, as measured by some function of estimation error. Accordingly, distributed estimation derived from SMAF communicates only when such an action is informative.

The fundamental idea behind set-membership filtering is the assumption that the measurement noise is considered bounded, and the bound is either known or can be estimated. Then, a *feasibility set* is found such that the bounded error specification is met for any member of this set [44]. The problem of recursive state estimation in a linear dynamic system in the presence of bounded noise was first studied by Schweppe in his 1968 paper [45]. The concept of a bounding ellipsoid was introduced here. This bounding ellipsoid in the state space encloses the true states and bounds the intersection of two ellipsoidal feasibility sets, one resulting from the state-transition equations and the other from the measurement equation. However, the optimization of this bounding ellipsoid was not considered. Furthermore, this algorithm updated the parameter estimates even when new data does not contribute any *innovation*.

The same problem was formulated as a system identification problem by Fogel and Huang, where the bounding ellipsoid was optimized with respect to both the volume and over the sum of the semi-axes of the bounding ellipsoid [46]. Another quantity, σ, which is a characteristic of the bounding ellipsoid related to the mean-square error [47], can also be used to optimize the bounding ellipsoid and this class of optimal bounding ellipsoid (OBE) algorithms are known as σ-optimizer OBE

algorithms [47–51]. A classification of different SMAF algorithms can be found in [52].

5.1 The Extended BEACON Algorithm

In this section we develop an event-triggered nonlinear estimation algorithm known as extended Bounding Ellipsoid Adaptive CONtrained least squares or eBEACON. This algorithm is based on the BEACON algorithm developed in [48] by Nagraj *et al.* We shall begin by deriving SMAF-based estimator for a generalized nonlinear measurement model. In the following section, we apply this solution to the two-level MASE framework in order to arrive at the proposed state estimator.

Let $\boldsymbol{\theta}(n)$ be an arbitrary state vector at time n. Then, the set of observations $\mathbf{y}(n)$ that relates to the state vector is expressed as

$$\mathbf{y}(n) = \mathbf{f}\{\boldsymbol{\theta}(n)\} + \mathbf{u}(n), \tag{5.1}$$

where $\mathbf{f}\{\cdot\}$ is a set of nonlinear functions operating on $\boldsymbol{\theta}(n)$ and $\mathbf{u}(n)$ is the additive zero-mean observation noise at time n.

We assume the norm of the observation noise vector is bounded in magnitude by a fixed value, say γ which is either estimated or known *a priori*. Therefore,

$$\|\mathbf{y}(n) - \mathbf{f}\{\boldsymbol{\theta}(n)\}\|^2 \leq \gamma^2. \tag{5.2}$$

Let us now replace $\mathbf{f}\{\boldsymbol{\theta}(n)\}$ by its first-order Taylor series expansion in (5.2). The point of linearization is assumed to be the latest state estimate $\hat{\boldsymbol{\theta}}(n-1)$. This gives us

$$\|\mathbf{y}(n) - \mathbf{f}\{\hat{\boldsymbol{\theta}}(n-1)\} + \mathbf{F}_{n-1}\hat{\boldsymbol{\theta}}(n-1) - \mathbf{F}_{n-1}\boldsymbol{\theta}(n)\|^2 \leq \gamma^2, \tag{5.3}$$

which may be written as

$$\|\tilde{\mathbf{y}}(n) - \mathbf{F}_{n-1}\boldsymbol{\theta}(n)\|^2 \leq \gamma^2, \tag{5.4}$$

where $\tilde{\mathbf{y}}(n) = \mathbf{y}(n) - \mathbf{f}\{\hat{\boldsymbol{\theta}}(n-1)\} + \mathbf{F}_{n-1}\hat{\boldsymbol{\theta}}(n-1)$ and $\mathbf{F}_{n-1}$ is the Jacobian given by

$$\mathbf{F}_{n-1} = \left.\frac{\partial \mathbf{f}\{\boldsymbol{\theta}\}}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}(n-1)}. \tag{5.5}$$

Now, (5.4) gives us a constraint set for all possible values of the state vector $\boldsymbol{\theta}(n)$ at time n given the observation set $\mathbf{y}(n)$. This constraint set forms an ellipsoid given by

$$\mathcal{H}(n) = \{\boldsymbol{\theta} \in \mathbb{R}^K : \|\tilde{\mathbf{y}}(n) - \mathbf{F}_{n-1}\boldsymbol{\theta}\|^2 \leq \gamma^2\}. \tag{5.6}$$

Given a sequence of pairs $\mathbf{y}(n), \boldsymbol{\theta}(n)$, $n = 1, 2, \ldots, N$, assuming the parameter vector to be estimated has remained constant, it must lie in the inside the intersection of all constraint sets up to N, i.e.

$$\boldsymbol{\theta} \in \Omega(N) \triangleq \cap_{n=1}^{N} \mathcal{H}(n) = \Omega(N-1) \cap \mathcal{H}(N). \tag{5.7}$$

However, $\Omega(N)$ forms a polytope in the parameter space which is difficult to describe mathematically. Therefore, it is easier to outer-bound this polytope with another ellipsoid. Consider an ellipsoid in the state space of θ that encloses the polytope formed by the set $\Omega(N-1)$. Let us denote this by $\mathcal{E}(N-1)$. Now, the ellipsoid that bounds the set $\Omega(N)$, denoted as $\mathcal{E}(N)$, is a superset containing the intersection of $\mathcal{E}(N-1)$ and $\mathcal{H}(N)$, i.e.

$$\mathcal{E}(N) \supset \mathcal{E}(N-1) \cap \mathcal{H}(N). \tag{5.8}$$

It is important to note that $\mathcal{E}(N)$ contains the true state at time N. Figure 5.2 shows a simple, two-dimensional pictorial representation of of the different ellipsoidal sets.

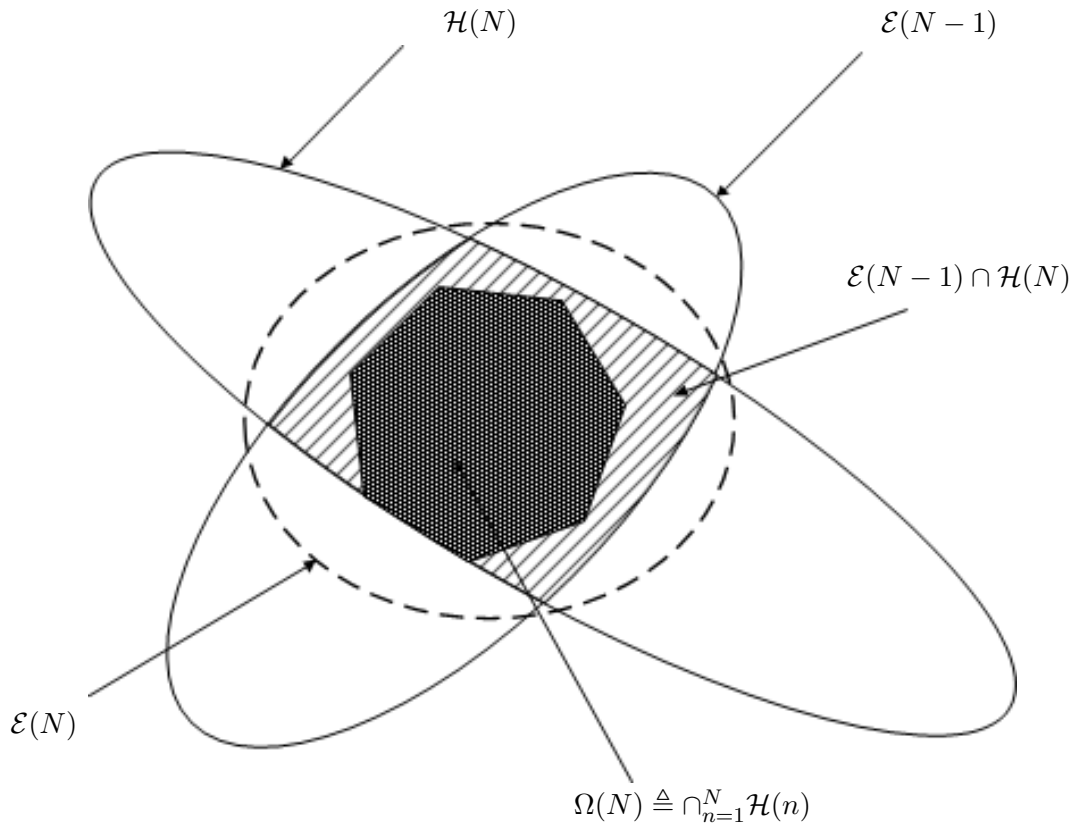


Figure 5.2: Geometric representation of set-membership filtering.

By extending the above rationale, we may derive an event-triggered ellipsoidal bounding state estimation algorithm to estimate the state vector $\boldsymbol{\theta}(n)$ at a time instant n, assuming the state vector is *quasi-static*. Consider an ellipsoid $\mathcal{E}(n-1)$ at a time instant $n-1$ expressed as

$$\mathcal{E}(n-1) = \{\boldsymbol{\theta} \in \mathbb{R}^K : (\boldsymbol{\theta} - \boldsymbol{\theta}(n-1))^T \mathbf{P}_{n-1}^{-1} (\boldsymbol{\theta} - \boldsymbol{\theta}(n-1)) \leq \sigma^2(n-1)\}, \tag{5.9}$$

where $\sigma^2(n-1)$ is a scalar which decides the size of the ellipsoid at time $n-1$.

At a time n, the true state $\boldsymbol{\theta}(n)$ is contained in the ellipsoid that outer bounds the intersection of the ellipsoidal set $\mathcal{E}(n-1)$ and the degenerate ellipsoid resulting

from the bounded noise assumption, $\mathcal{H}(n)$. This ellipsoid is denoted by $\mathcal{E}(n)$ and is given by a linear combination of (5.9) and (5.4)

$$\mathcal{E}(n) = \alpha\mathcal{E}(n-1) + \beta\mathcal{H}(n),$$

specifically,

$$\begin{aligned}\mathcal{E}(n) =& \{\boldsymbol{\theta} \in \mathbb{R}^K : \alpha(\boldsymbol{\theta} - \boldsymbol{\theta}(n-1))^T \mathbf{P}_{n-1}^{-1}(\boldsymbol{\theta} - \boldsymbol{\theta}(n-1)) \\ &+ \beta\,||\tilde{\mathbf{y}}(n) - \mathbf{F}_{n-1}\{\boldsymbol{\theta}\}||^2 \leq \alpha\sigma^2(n-1) + \beta\gamma^2\}. \end{aligned} \tag{5.10}$$

Upon expansion of (5.10), we obtain the following relations:

$$\boldsymbol{\theta}^T \mathbf{P}_n^{-1} \boldsymbol{\theta} = \boldsymbol{\theta}^T \left[\alpha \mathbf{P}_{n-1}^{-1} + \beta \mathbf{F}_{n-1}^T \mathbf{F}_{n-1}\right] \boldsymbol{\theta}, \tag{5.11}$$

$$\boldsymbol{\theta}(n)^T \mathbf{P}_n^{-1} \boldsymbol{\theta} = \left[\alpha \boldsymbol{\theta}^T(n-1) \mathbf{P}_{n-1}^{-1} + \beta \tilde{\mathbf{y}}^T(n) \mathbf{F}_{n-1}\right] \boldsymbol{\theta}, \tag{5.12}$$

$$\text{and } \boldsymbol{\theta}^T \mathbf{P}_n^{-1} \boldsymbol{\theta}(n) = \boldsymbol{\theta}^T \left[\alpha \mathbf{P}_{n-1}^{-1} \boldsymbol{\theta}(n-1) + \beta \mathbf{F}_{n-1}^T \tilde{\mathbf{y}}(n)\right]. \tag{5.13}$$

From (5.11), we get a recursive expression for the shaping matrix given by

$$\mathbf{P}_n^{-1} = \alpha \mathbf{P}_{n-1}^{-1} + \beta \mathbf{F}_{n-1}^T \mathbf{F}_{n-1}, \tag{5.14}$$

and from (5.13), we have the update expression for the state vector

$$\boldsymbol{\theta}(n) = \alpha \mathbf{P}_n \mathbf{P}_{n-1}^{-1} \boldsymbol{\theta}(n-1) + \beta \mathbf{P}_n \mathbf{F}_{n-1}^T \tilde{\mathbf{y}}(n). \tag{5.15}$$

Now, (5.14) may be written as

$$\mathbf{P}_n = \left[\alpha \mathbf{P}_{n-1}^{-1} + \beta \mathbf{F}_{n-1}^T \mathbf{F}_{n-1}\right]^{-1} \tag{5.16}$$

$$= \left[\alpha \mathbf{I} + \beta \mathbf{P}_{n-1} \mathbf{F}_{n-1}^T \mathbf{F}_{n-1}\right]^{-1} \mathbf{P}_{n-1}. \tag{5.17}$$

Upon the application of the matrix inversion lemma and simplification, we have

$$\mathbf{P}_n = \frac{1}{\alpha}\left[\mathbf{I} - \beta \mathbf{P}_n \mathbf{F}_{n-1}^T \mathbf{F}_{n-1}\right] \mathbf{P}_{n-1}. \tag{5.18}$$

Now, substituting the above expression in (5.15), we have

$$\begin{aligned}\boldsymbol{\theta}(n) &= \left[\mathbf{I} - \beta \mathbf{P}_n \mathbf{F}_{n-1}^T \mathbf{F}_{n-1}\right] \boldsymbol{\theta}(n-1) + \beta \mathbf{P}_n \mathbf{F}_{n-1}^T \tilde{\mathbf{y}}(n) \\ &= \boldsymbol{\theta}(n-1) + \beta \mathbf{P}_n \mathbf{F}_{n-1}^T \left[\tilde{\mathbf{y}}(n) - \mathbf{F}_{n-1} \boldsymbol{\theta}(n-1)\right] \\ &= \boldsymbol{\theta}(n-1) + \beta \mathbf{P}_n \mathbf{F}_{n-1}^T \left[\mathbf{y}(n) - \mathbf{f}(\boldsymbol{\theta}(n-1))\right] \\ &= \boldsymbol{\theta}(n-1) + \beta \mathbf{P}_n \mathbf{F}_{n-1}^T \mathbf{e}_n, \end{aligned} \tag{5.19}$$

where $\mathbf{e}_n = \{\mathbf{y}(n) - \mathbf{f}(\boldsymbol{\theta}(n-1))\}$.

An alternative form of the update expression which utilizes only past values of the shaping matrix $\mathbf{P}$ may be derived as follows: Consider the update term in (5.19), $\beta \mathbf{P}_n \mathbf{F}_{n-1}^T \mathbf{e}_n$. Substituting (5.14) in the expression for the update term, we have

$$\beta \mathbf{P}_n \mathbf{F}_{n-1}^T \mathbf{e}_n = \beta \left[\alpha \mathbf{P}_{n-1}^{-1} + \beta \mathbf{F}_{n-1}^T \mathbf{F}_{n-1}\right]^{-1} \mathbf{F}_{n-1}^T \mathbf{e}_n.$$

Applying the matrix inversion lemma we have

$$\beta \mathbf{P}_n \mathbf{F}_{n-1}^T \mathbf{e}_n = \frac{\beta}{\alpha}\left[\mathbf{P}_{n-1} - \frac{1}{\alpha}\mathbf{P}_{n-1}\mathbf{F}_{n-1}^T\left\{\frac{1}{\beta}\mathbf{I} + \frac{1}{\alpha}\mathbf{F}_{n-1}\mathbf{P}_{n-1}\mathbf{F}_{n-1}^T\right\}^{-1}\mathbf{F}_{n-1}\mathbf{P}_{n-1}\right]\mathbf{F}_{n-1}^T\mathbf{e}_n$$
$$= \beta \mathbf{P}_{n-1}\mathbf{F}_{n-1}^T\left\{\alpha\mathbf{I} + \beta\mathbf{F}_{n-1}\mathbf{P}_{n-1}\mathbf{F}_{n-1}^T\right\}^{-1}\mathbf{e}_n. \tag{5.20}$$

Therefore, the alternative form of (5.19) is given by

$$\boldsymbol{\theta}(n) = \boldsymbol{\theta}(n-1) + \beta\mathbf{P}_{n-1}\mathbf{F}_{n-1}^T\left\{\alpha\mathbf{I} + \beta\mathbf{F}_{n-1}\mathbf{P}_{n-1}\mathbf{F}_{n-1}^T\right\}^{-1}\mathbf{e}_n. \tag{5.21}$$

Considering the remaining terms in (5.10), we have the expression

$$\boldsymbol{\theta}^T\mathbf{P}_n^{-1}\boldsymbol{\theta} - \boldsymbol{\theta}(n)^T\mathbf{P}_n^{-1}\boldsymbol{\theta} - \boldsymbol{\theta}^T\mathbf{P}_n^{-1}\boldsymbol{\theta}(n) + \alpha\boldsymbol{\theta}(n-1)^T\mathbf{P}_{n-1}^{-1}\boldsymbol{\theta}(n-1)$$
$$+\beta\|\tilde{\mathbf{y}}(n)\|^2 + \boldsymbol{\theta}^T(n)\mathbf{P}_n^{-1}\boldsymbol{\theta} - \boldsymbol{\theta}(n)^T\mathbf{P}_n^{-1}\boldsymbol{\theta}(n) \leq \alpha\sigma^2(n-1) + \beta\gamma^2.$$

Rearranging the terms we have

$$(\boldsymbol{\theta} - \boldsymbol{\theta}(n))^T\mathbf{P}_n^{-1}(\boldsymbol{\theta} - \boldsymbol{\theta}(n)) \leq \alpha\sigma^2(n-1) + \beta\gamma^2 - \alpha\boldsymbol{\theta}(n-1)^T\mathbf{P}_{n-1}^{-1}\boldsymbol{\theta}(n-1)$$
$$- \boldsymbol{\theta}^T(n)\mathbf{P}_n^{-1}\boldsymbol{\theta}(n) - \beta\|\tilde{\mathbf{y}}(n)\|^2.$$

This can be written as

$$\sigma^2(n) = \alpha\sigma^2(n-1) + \beta\gamma^2 - \beta\|\tilde{\mathbf{y}}(n)\|^2 - \alpha\boldsymbol{\theta}(n-1)^T\mathbf{P}_{n-1}^{-1}\boldsymbol{\theta}(n-1) - \boldsymbol{\theta}^T(n)\mathbf{P}_n^{-1}\boldsymbol{\theta}(n). \tag{5.22}$$

Substituting (5.14) and (5.21) in (5.22), we arrive at the following expression for $\sigma^2(n)$.

$$\sigma^2(n) = \alpha\sigma^2(n-1) + \beta\gamma^2 - \alpha\beta\mathbf{e}_n^T\mathbf{Q}_n^{-1}\mathbf{e}_n, \tag{5.23}$$

where $\mathbf{Q}_n = \left[\alpha\mathbf{I} + \beta\mathbf{F}_{n-1}\mathbf{P}_{n-1}\mathbf{F}_{n-1}^T\right]$.

If α and β are chosen to be 1 and λ respectively (as with the BEACON algorithm), where λ is a scalar and $\lambda \geq 0$, and employing the eigenvalue decomposition of $\mathbf{Q}_n$ in (5.23) we have

$$\sigma^2(n) = \sigma^2(n-1) + \lambda\gamma^2 - \lambda\sum_i \frac{(\mathbf{e}_n^T\mathbf{d}_i)^2}{1+\lambda\rho_i} \tag{5.24}$$

where ρ_i and $\mathbf{d}_i$ are the ith eigenvalue and eigenvector or the matrix $\mathbf{F}_{n-1}\mathbf{P}_{n-1}\mathbf{F}_{n-1}^T$, respectively. Now, $\sigma^2(n)$ is upper bounded by

$$\sigma^2(n) = \sigma^2(n-1) + \lambda\gamma^2 - \frac{\lambda\|\mathbf{e}_n\|^2}{1+\lambda\rho_{max}} \tag{5.25}$$

where ρ_{max} is the maximum singular value of $\mathbf{F}_{n-1}\mathbf{P}_{n-1}\mathbf{F}_{n-1}^T$. Equation (5.26) can be rewritten as

$$\sigma^2(n) = \sigma^2(n-1) - \left(\frac{\lambda\|\mathbf{e}_n\|^2}{1+\lambda\rho_{max}} - \lambda\gamma^2\right) \tag{5.26}$$

The parameter $\sigma^2(n)$ relates to the ellipsoid $\mathcal{E}(n)$ and the central idea behind the estimation algorithm is to find an optimally small ellipsoid that outer bounds the intersection of n ellipsoidal feasibility sets. For that to be true, $\sigma^2(n)$ must be less than $\sigma^2(n-1)$. This gives us a condition

$$\frac{\lambda\|\mathbf{e}_n\|^2}{1+\lambda\rho_{max}} - \lambda\gamma^2 > 0. \tag{5.27}$$

Next, in order to find the smallest possible bounding ellipsoid, i.e. the *optimal bounding ellipsoid* we must minimize the second term of (5.26) with respect to λ. Upon minimization, the optimal value of λ is found to be

$$\lambda = \frac{1}{\rho_{max}}\left(\frac{\|\mathbf{e}_n\|^2}{\gamma^2} - 1\right) \tag{5.28}$$

substituting this in (5.27), the condition for which $\sigma^2(n) < \sigma^2(n-1)$ is given by

$$\|\mathbf{e}_n\|^2 > \gamma^2. \tag{5.29}$$

This expression is the key to the event-triggered nature of the estimation algorithm. This is because (5.29) implies that unless the condition $\|\mathbf{e}_n\|^2 > \gamma^2$ is satisfied, there is no point in updating since this would do nothing to shrink the bounding ellipsoid further. This means that (5.29) forms an *innovation check*, i.e. (5.29) tells us whether or not there is sufficient innovation in the measurement set $\mathbf{y}(n)$ in order to warrant an update. Therefore λ is given by

$$\lambda = \begin{cases} 0 & \text{if } \|\mathbf{e}_n\|^2 \leq \gamma^2 \\ \frac{1}{\rho_{max}}\left(\frac{\|\mathbf{e}_n\|^2}{\gamma^2} - 1\right) & \text{otherwise} \end{cases} \tag{5.30}$$

The recursions of the SMAF-based estimation algorithm are summarized below.

Algorithm 3 The eBEACON algorithm

if $\|\mathbf{y}(n) - \mathbf{f}\{\hat{\boldsymbol{\theta}}(n-1)\}\| > \gamma^2$

$$\begin{aligned} \lambda &= \frac{1}{\rho_{max}}\left(\frac{\|\mathbf{e}_n\|^2}{\gamma^2} - 1\right) \\ \mathbf{P}_n^{-1} &= \alpha\mathbf{P}_{n-1}^{-1} + \beta\mathbf{F}_{n-1}^T\mathbf{F}_{n-1} \\ \hat{\boldsymbol{\theta}}(n) &= \hat{\boldsymbol{\theta}}(n-1) + \beta\mathbf{P}_n\mathbf{F}_{n-1}^T\left[\mathbf{y}(n) - \mathbf{f}\{\hat{\boldsymbol{\theta}}(n-1)\}\right] \end{aligned}$$

else

$$\begin{aligned} \lambda &= 0 \\ \hat{\boldsymbol{\theta}}(n) &= \hat{\boldsymbol{\theta}}(n-1) \end{aligned}$$

5.2 Event-Triggered Two-Level MASE

In this section, we apply the eBEACON algorithm to solve the problem of state estimation in the two-level MASE framework. In order to solve the problem of state estimation in large, complex, and highly interconnected power systems, we divide the system into M non-overlapping areas or subsystems. Each area m is associated with its local measurement function, namely,

$$\mathbf{z}_m(k) = \mathbf{h}_m\left[\mathbf{x}_m(k)\right] + \mathbf{n}_m(k), \quad m = 1, \ \ldots, \ M \tag{5.31}$$

where, as in (2.28), $\mathbf{z}_m(k) \in \mathbb{R}^{L_m \times 1}$ is the local measurement vector, and $\mathbf{x}_m(k) = [\mathbf{x}_{\mathbf{i}m}^{\mathrm{T}}(k) \ \mathbf{x}_{\mathbf{b}m}^{\mathrm{T}}(k)]^{\mathrm{T}}$ is the state vector which is further partitioned into *internal state variables*, $\mathbf{x}_{\mathbf{i}m}^{\mathrm{T}}$, and *border state variables*, $\mathbf{x}_{\mathbf{b}m}^{\mathrm{T}}$. Internal state variables are those that are observable for the particular area while border state variables are the states of those buses with lines connecting two areas (referred to as tie-lines).

In the conventional MASE framework, the coordinator reads tie-line parameters, and recalculates the estimates of the border state variables and the phase angles on the local slack buses and sends these back to the regional state estimators at every time instant [53]. In contrast, in the scheme proposed here, the regional state estimators transmit their local estimate of the border variables only when those estimates are updated (triggered by an event). This reduces the amount of data being transferred and minimizes communication requirements.

For two-level MASE, the distributed SMAF algorithm described proposed here consists of two steps: 1) the local state update, and; 2) co-ordination step where border states are combined. These two steps are described next.

5.2.1 Local update

In the local update phase, the state estimator associated with each area calculates the innovation check given by (5.29); specifically,

$$\|\mathbf{z}_m(k) - \mathbf{h}_m\left[\hat{\mathbf{x}}'_m(k-1)\right]\|^2 > \gamma_m^2 \tag{5.32}$$

where as in $\hat{\mathbf{x}}'_m(k-1)$ is the state estimate after the co-ordination step at time $k-1$. If (5.32) is found to be true, it means that there is sufficient innovation in the measurement set $\mathbf{z}_m(k)$ to warrant an update of the state vector. The update expressions for the state vector at a time k are given below and they are cognates of (5.19) and (5.14):

$$\begin{aligned} \hat{\mathbf{x}}_m(k) &= \hat{\mathbf{x}}'_m(k-1) + \lambda_m(k)\mathbf{P}_m(k)\mathbf{H}_m^{\mathrm{T}}(k)\mathbf{e}_m(k) \\ \mathbf{e}_m(k) &= \mathbf{z}_m(k) - \mathbf{h}_m\left[\hat{\mathbf{x}}'_m(k-1)\right] \\ \mathbf{P}_m^{-1}(k) &= \mathbf{P}_m^{-1}(k-1) + \lambda_m(k)\mathbf{H}_m^{\mathrm{T}}(k)\mathbf{H}_m(k) \end{aligned} \tag{5.33}$$

where $\hat{\mathbf{x}}'_m(k-1) = [\hat{\mathbf{x}}_{im}^{\mathrm{T}}(k-1) \ \hat{\mathbf{x}}_{bm}'^{\mathrm{T}}(k-1)]^{\mathrm{T}}$ and vector $\hat{\mathbf{x}}'_{bm}(k-1)$ contains the combined border state variables obtained (and fed back) from the co-ordination level. The update parameter $\lambda_m(k)$ is given by

$$\lambda_m(k) = \frac{1}{\rho_{max}}\left[\frac{\|\mathbf{e}_m(k)\|}{\gamma_m} - 1\right] \tag{5.34}$$

where ρ_{max} is the largest eigenvalue of $\mathbf{H}_m(k)\mathbf{P}_m(k-1)\mathbf{H}_m^{\mathrm{T}}(k)$.

Each region updates its state estimate based on (5.33). Thereafter, the states are transmitted to the co-ordination level. We note that the algorithm in (5.33) updates the state estimates only when the received data contain sufficient *innovation*, as indicated by (5.32). As such, $\lambda_m(k) \neq 0$, or equivalently $\|\mathbf{e}_m(k)\| > \gamma_m$, defines the triggering event.

5.2.2 Co-ordination phase

The function of the co-ordination step is to obtain an interconnection-wide state estimate while exploiting the diversity of border state estimates to improve the local state estimates. Typically, in a two-level MASE scheme, the co-ordination level receives tie-line measurements, boundary measurements and suitable pseudo-measurements from all the regions [53]. These collectively make up a set of measurements $\mathbf{z}_c(k)$ and a consensus state estimate is obtained by solving a second estimation problem expressed as

$$\mathbf{z}_c(k) = \mathbf{h}_c\left[\mathbf{x}_c(k)\right] + \mathbf{n}_c(k) \tag{5.35}$$

where $\mathbf{n}_c(k)$ is the error associated with the measurements $\mathbf{z}_c(k)$ and $\mathbf{x}_c(k) = [\mathbf{x}_{b1}^{\mathrm{T}}(k)\cdots\mathbf{x}_{bM}^{\mathrm{T}}(k)\theta_1^{sk}(k)\cdots\theta_M^{sk}(k)]^{\mathrm{T}}$, $\theta_m^{sk}(k)$ being the phase angle of the local slack bus of region m referred to the global slack bus.

The updated (or consensus) border variables $\{\hat{\mathbf{x}}'_{bm}(k)\}_{m=1}^{M}$, obtained by solving (5.35), are sent back to all the regions to be used in the next iteration of (5.33). As noted above, if no update takes place in a particular area it will not transmit anything to the co-ordination level, which will then use the most recent reported variables. Furthermore, if none of the areas sharing border variables performed an update, the co-ordination level need not solve (5.35) and $\hat{\mathbf{x}}'_m(k) = \hat{\mathbf{x}'}_m(k-1)$.

The above method, however, is not a unique solution for the co-ordination step and other algorithms for co-ordination are also possible. Advances in distributed estimation, which has been an active field of research in the signal processing community lately, can also be applied to the MASE framework. In distributed estimation, several nodes (or areas in case of MASE) estimate a common parameter vector through local collaborations. In the case of MASE, the measurements of each area only relates to a small part of the whole state vector. Thus, the resulting computational and communication costs of a distributed estimation approach depend on whether local knowledge of the whole state vector is required or not. For example, by redefining the correction vector in (2.13) as $\mathbf{\Delta x} = [\mathbf{\Delta x}_1^{\mathrm{T}} \ \cdots \ \mathbf{\Delta x}_M^{\mathrm{T}}]^{\mathrm{T}}$, an iterative WLS solution for the MASE takes the form

$$\begin{aligned} \mathbf{\Delta x}(j) &= \left[\sum_{m=1}^{M}\mathbf{H}_m^{\mathrm{T}}(j)\mathbf{W}_m^{-1}\mathbf{H}_m(j)\right]^{-1}\sum_{m=1}^{M}\mathbf{H}_m^{\mathrm{T}}(j)\mathbf{W}_m^{-1}[\mathbf{z}_m - \mathbf{h}_m(\mathbf{x}_m(j))] \\ \hat{\mathbf{x}}_m(j+1) &= \hat{\mathbf{x}}_m(j) + \mathbf{\Delta x}_m(j) \end{aligned} \tag{5.36}$$

where $\mathbf{H}_m(j)$ is the measurement Jacobian of area m obtained with the local state estimate $\hat{\mathbf{x}}_m(j)$. The simplest method seems to be to express (5.36) in terms of averages $\frac{1}{M}\sum_m \mathbf{H}_m^{\mathrm{T}}\mathbf{W}_m^{-1}\mathbf{H}_m$ and $\frac{1}{M}\sum_m \mathbf{H}_m^{\mathrm{T}}\mathbf{W}_m^{-1}[\mathbf{z}_m - \mathbf{h}_m(\mathbf{x}_m)]$ across the areas. These

so-called consensus algorithms (also related to gossip algorithms) can be expected to be of relevance here, see, e.g. [25, 26]. Then, two separate consensus algorithms can be used to compute these quantities, and in turn, the quantity $\Delta\mathbf{x}_m$ to be used at the jth iteration of the estimator. However, this method requires communication corresponding to two consensus algorithms being executed in parallel. Perhaps more importantly, it requires the two algorithms to converge before the quantity $\Delta\mathbf{x}_m$ can be computed and hence the iteration of the estimation algorithm can be done. In other words, this approach requires the consensus algorithm to be executed at a much faster time scale than the consensus algorithm. While there has been some characterization of the performance loss when the time scales do not separate smoothly [54–56], the general problem still remains open.

The local update step and the co-ordination step described in this section constitute a novel, two-level MASE algorithm which is known as Multiple-Input eBEACON or MI-eBEACON and is similar to the MI-eBEACON algorithm in [57].

5.3 Simulation Results

In this section we discuss the results of a computer simulation of the proposed algorithm as formulated in (5.33). The MI-eBEACON algorithm is tested with the standard IEEE 14 bus system, which is partitioned into three areas as shown in Fig. 5.3. The simulation is performed over 250 time instants. The dynamics of the system are simulated by introducing a sudden load increase of 10% on all load buses at the 90th time instant and then restoring the load to its original value at the 150th time instant while a zero-mean normally distributed jitter with standard deviation $\sigma_P = 0.8 \times 10^{-3}$ is added to the load at every time instant. A load flow calculation is then performed using the MatPower toolbox in MATLAB [41] to update line flows, voltage magnitudes and phase angles throughout the system.

For simplicity we assume that the system is fully observable by PMUs, which are placed at buses 2,6,8 and 9. PMUs measure not only voltage phasors at buses where they are installed but also current phasors through all incident buses. Since the current phasor on a line between two buses is linearly related to the two voltage phasors at those two buses [37], the PMU measurements across the system are linearly related to the voltage phasors [5]. Furthermore, phasor measurements are synchronized with a global time reference so the co-ordination level is simplified and a consensus estimate of the border state variables within a neighbourhood are combined by simply taking a weighted average. While more sophisticated consensus algorithms may be considered in the future, see, e.g. [58], this topic is out the scope of the work here. In particular, the border variable $\hat{x}_{bm,n}(k)$ of area m is common with areas defined by neighbourhood $\mathcal{G}_{mn}$, and consequently we get

$$\hat{x}'_{bm,n}(k) = c^{(n)}_{m,m}(k)\hat{x}_{bm,n}(k) + \sum_{l \in \mathcal{G}_{mn} \setminus \{mn\}} c^{(n)}_{l,m}(k)\hat{x}_{bl,n}(k)$$

The regional state estimators have no *a priori* information about the state and begin with a flat start, i.e. the real part of all voltage phasors are set to 1 p.u. (per

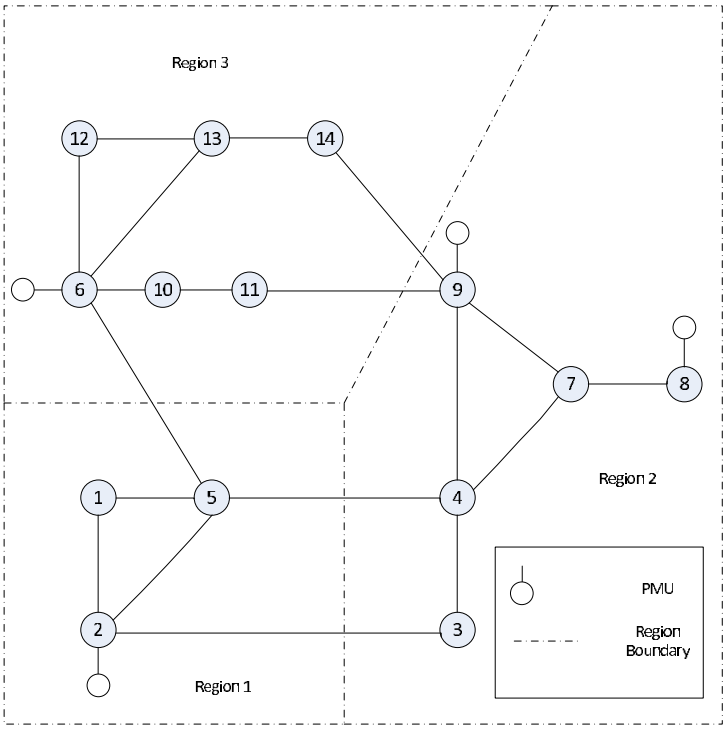


Figure 5.3: IEEE 14 bus system divided into three areas with PMU placements.

unit) and the imaginary parts are set to 0 p.u. To make the measurements emulate a real system, we introduced a zero-mean white Gaussian noise to corrupt the measurements. In a typical PMU, both the voltage and current magnitude measurements have a standard deviation of 0.006 p.u., while the voltage and current phase angles have standard deviations of 0.52° and 1.04°, respectively. This must be taken into account when generating the random measurement noise in the simulation.

The simulations are performed for different error bounds γ_m which are specified by

$$\gamma_m = \sqrt{\alpha \mathrm{tr}(\mathbb{E}[\mathbf{n}_m(k)\mathbf{n}_m^{\mathrm{T}}(k)])} \tag{5.37}$$

where α can be varied so as to control the frequency of updates, trading off with performance.

Figure 5.4 shows that the algorithm is quite capable of tracking the sudden drop in voltage caused by, e.g. a surge in demand. In this case, $\alpha = 4$, and close observation of the figure reveals that updates occur mostly after the bus-voltage changes. This kind of event-triggered SE offers considerable saving in communication between subsystems, particularly when the grid is quasi-static.

The performance measure $\delta(k)$ is calculated by averaging the measurement error over I realizations

$$\delta(k) = \frac{1}{MI}\sum_{i=1}^{I}\sum_{m=1}^{M}\frac{1}{L_m}||\mathbf{z}_m^{(i)}(k) - \mathbf{H}_m^{(i)}\hat{\mathbf{x}}_m^{(i)}(k)||^2 \tag{5.38}$$

where $\hat{\mathbf{x}}_m(k)$ is the estimate of $\mathbf{x}_m(k)$.

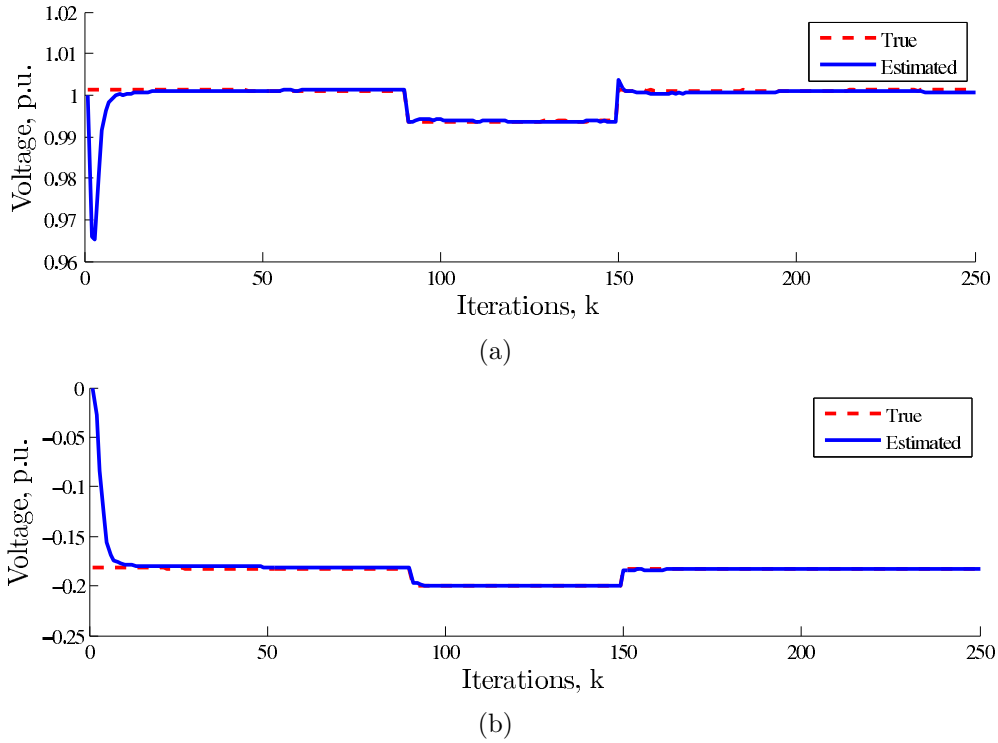


Figure 5.4: Tracking ability of the MI-eBEACON algorithm showing the real part (Figure 5.4(a)) and imaginary part(Figure 5.4(b)) of the voltage at bus 4.

Figure 5.5 shows the performance of the proposed algorithm, and compares it to that of the recursive least-squares (RLS) algorithm. The parameter λ is known as the forgetting factor or the weighting factor and reduces the influence of old data. It is clear from the figure that as we increase the error bound, the performance degrades. Also, the proposed algorithm is clearly superior to the RLS estimator.

The amount of communication is measured simply by calculating the number of times an update occurred as a percentage of the total number of time instants. This is because communication occurs only if the local state estimate has been updated. The average update frequency per area is a good measure of the communication costs associated with updates. Figure 5.6 shows how changing the error bound affects the average update frequency and the steady-state performance as measured by $\delta_s = \frac{1}{50}\sum_{k=200}^{250}\delta(k)$. We can see that the performance degrades at a much slower rate when compared to the decrease of the update frequency. This means that a careful choice of γ_m would significantly reduce the amount of communication between regions without significantly compromising the performance.

In the scenario studied, we observe that the computational complexity at each update instant is of roughly same order, $\mathcal{O}(N_m^3)$, for both the RLS algorithm and the MI-eBEACON algorithm. However, for the MI-eBEACON algorithm updates occur only 14% of the time (when $\alpha = 4$) in the case studied (see Figure 5.6).

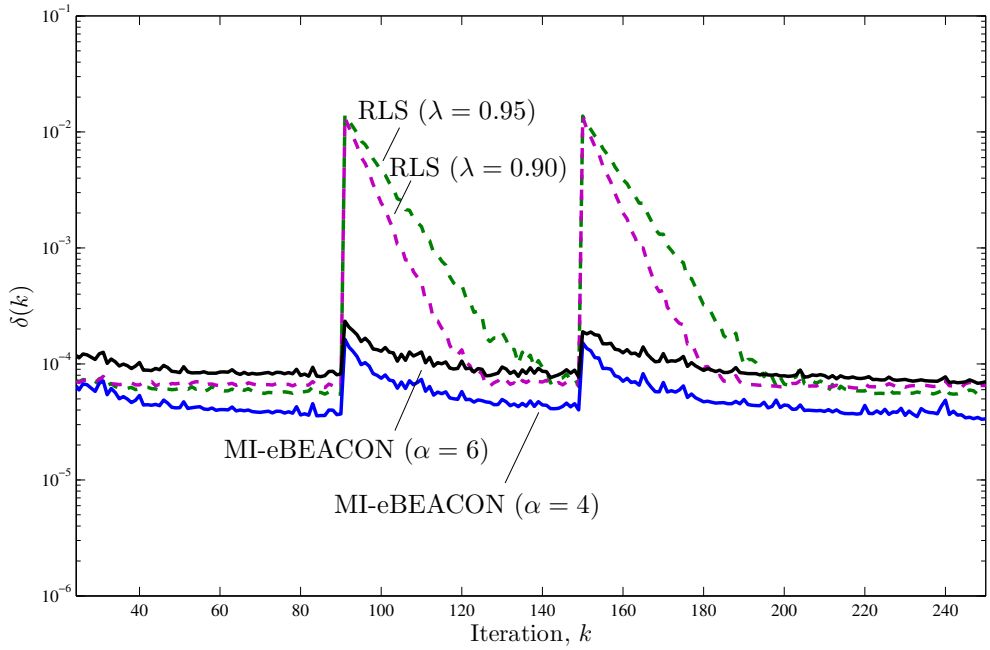


Figure 5.5: Comparing MI-eBEACON to RLS in terms of $\delta(k)$.

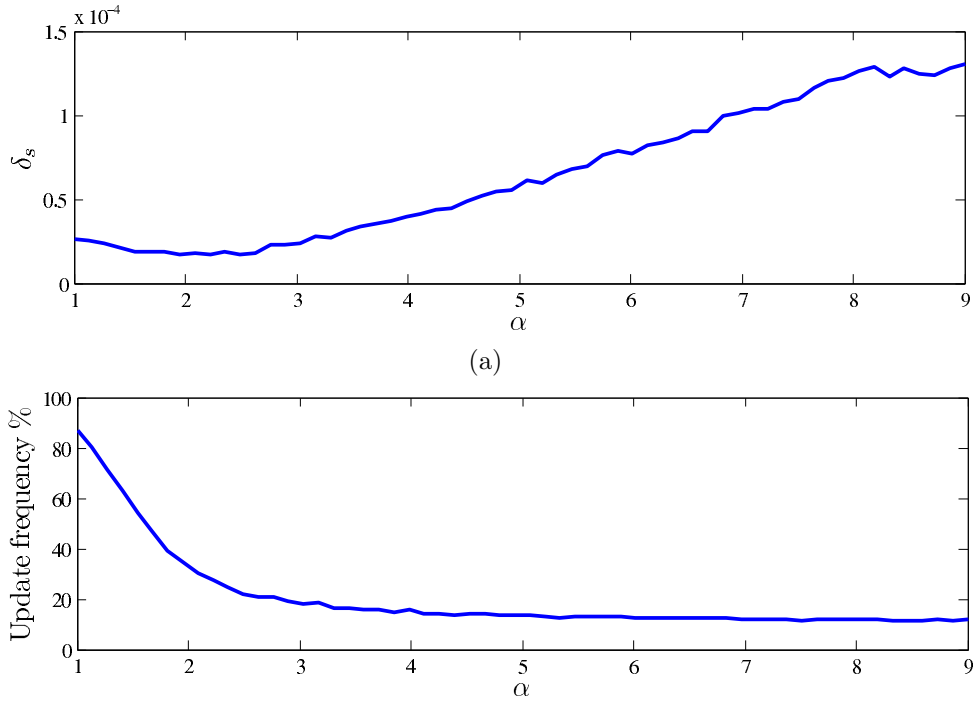


Figure 5.6: Steady state performance measure δ_s (Figure 5.6(a)) and update frequency (Figure 5.6(b)) as a function of α.

We note that the aforementioned complexity figures are pessimistic, for the computational complexity can be further reduced by exploiting the sparseness of $\mathbf{H}_m(k)$. Furthermore, the communication cost of the MI-eBEACON algorithm is almost 86% less when compared to the RLS approach. This is because data transfer occurs between the co-ordination level and the local state estimators at every time instant for the RLS approach, while it occurs, on average, only 14% of the time for the MI-eBEACON algorithm. In general, updates mostly take place when there is a change, e.g. load or topological changes, in the system.

5.4 Discussion

In this chapter we design a distributed adaptive estimation technique for multi-area state estimation in power systems and study its performance using a simulation. Distributed approaches can enhance the computational performance and the reliability of SE algorithms. Efficient and reliable communication is the backbone for the distributed SE algorithms. The scheme described here, known as MI-eBEACON, updates only when it is necessary (event-triggered) while conventional approaches update the state estimate at every time instant regardless of the benefits of the update. This resulting sparse update feature leads to reduced communication between neighbours, reduced interference and reduced local processing cost.

The steps of the MI-eBEACON algorithm are summarized below:

Algorithm 4 The MI-eBEACON Algorithm

for $k = 1 \ldots K$ and $m = 1 \ldots M$

Local update step:

if $\|\mathbf{z}_m(k) - \mathbf{h}_m\left[\hat{\mathbf{x}}'_m(k-1)\right]\|^2 > \gamma_m^2$

1. Update local state estimate.

$$\begin{aligned}
\mathbf{e}_m(k) &= \mathbf{z}_m(k) - \mathbf{h}_m\left[\hat{\mathbf{x}}'_m(k-1)\right] \\
\lambda_m(k) &= \frac{1}{\rho_{max}}\left[\frac{\|\mathbf{e}_m(k)\|}{\gamma_m} - 1\right] \\
\mathbf{P}_m^{-1}(k) &= \mathbf{P}_m^{-1}(k-1) + \lambda_m(k)\mathbf{H}_m^{\mathrm{T}}(k)\mathbf{H}_m(k) \\
\hat{\mathbf{x}}_m(k) &= \hat{\mathbf{x}}'_m(k-1) + \lambda_m(k)\mathbf{P}_m(k)\mathbf{H}_m^{\mathrm{T}}(k)\mathbf{e}_m(k)
\end{aligned}$$

2. Transmit $\hat{\mathbf{x}}_m(k)$ to the co-ordination level.

else

1. Keep previous state estimate.

$$\begin{aligned}
\lambda_m(k) &= 0 \\
\hat{\mathbf{x}}_m(k) &= \hat{\mathbf{x}}'_m(k-1)
\end{aligned}$$

2. No communication occurs.

Co-ordination step:

1. Solve $\mathbf{z}_c(k) = \mathbf{h}_c\left[\mathbf{x}_c(k)\right] + \mathbf{n}_c(k)$
2. Form vectors $\{\hat{\mathbf{x}}'_{bm}(k)\}_{m=1}^{M}$ and transmit to local state estimators.

Chapter 6

Conclusions

The research work presented here deals with discovering new ways to carry out state estimation in power systems taking into account recent advances in the field of measurement technology and statistical signal processing. Two factors in particular, namely, the emergence of new measurement technologies and the departure from a vertically integrated utility structure, have revitalized research in power system state estimation. Specifically, larger and more complicated systems, coupled with the enormous amount of measurement data generated by new measurement technologies, call for a rethinking of the traditional state estimation paradigm in power systems and the development of new algorithms which face up to the challenges posed by power grids of the future.

The focus of the work presented in this thesis is the development of novel state estimation algorithms which can function within the existing state estimation framework while, at the same time, meeting the constraints imposed by the emerging smart grid. To begin with, resource-efficient dynamic state estimation algorithms must be developed which can combine measurements from both the traditional measurement devices as well as the newer phasor measurement units (PMUs) and operate on them optimally in order to arrive at an estimate of the system state. Furthermore, concentrating the state estimation function at a single point to monitor a large interconnection leads to huge communication and computational overheads. A more feasible approach would be to distribute the state estimation function throughout the interconnection. An *on-demand* estimation scheme featuring event-triggered communication is necessary to reduce the communication and computational overheads associated with distributed estimation in large systems.

In this thesis, a forecasting-aided state estimation scheme known as the mixed-measurement extended Kalman filter (MM/EKF) is developed to combine conventional measurements with PMU measurements. Also, a new, reduced complexity, reduced-order algorithm known as the reduced order extended Kalman filter (RO/EKF) algorithm is proposed which addresses certain drawbacks of the MM/EKF scheme. The ability of the MM/EKF and RO/EKF to track the evolution of the state vector in time is verified with computer simulations and the statistical performance of the RO/EKF algorithm with respect to the root mean-squared error is studied and compared with the performance of the MM/EKF estimator.

In order to address the demands on communication and computational resources placed by larger and more complicated systems, a novel, resource-efficient, adaptive estimation algorithm is developed. Using the concept of set-membership adaptive filtering, a new selective-update nonlinear adaptive filtering algorithm known as eBEACON is derived. This is extended to a two-level multi-area state estimation algorithm, known as known as MI-eBEACON [8], for multi-area state estimation in large interconnections. The performance of this is studied using a computer simulation and comparisons are drawn to recursive least-squares adaptive filtering.

In conclusion, while some researchers may consider the field of power system state estimation somewhat mature, new techniques for state estimation must be developed as the power grid becomes more complex, more interconnected and more intelligent. Looking ahead, progress in the field of estimation theory and adaptive filtering will greatly facilitate and benefit the development of the smart grid. In return, research on state estimation within the framework of one of the most complex man-made systems can invigorate the statistical signal processing research as a whole.

Bibliography

[1] A. Abur and A. Goméz-Expósito, *Power system state estimation: theory and implementation*, Marcel Dekker, New York, NY, 2004.

[2] T. Zheng and E. Litvinov, "On ex post pricing in the real-time electricity market," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 153–164, Feb. 2011.

[3] S. Massoud Amin and B.F. Wollenberg, "Toward a smart grid: power delivery for the 21st century," *IEEE Power Energy Mag.*, vol. 3, no. 5, pp. 34–41, Sep. 2005.

[4] A. Gómez-Expósito, A. Abur, A. de la Villa Jaén, and C. Gómez-Quiles, "A multilevel state estimation paradigm for smart grids," *Proc. of the IEEE*, vol. 99, no. 6, pp. 952–976, Jun. 2011.

[5] Ming Zhou, V.A. Centeno, J.S. Thorp, and A.G. Phadke, "An alternative for including phasor measurements in state estimators," *IEEE Trans. Power Syst.*, vol. 21, no. 4, pp. 1930–1937, Nov. 2006.

[6] V. Terzija, G. Valverde, D. Cai, P. Regulski, Madani V., J. Fitch, S. Skok, M. Begovic, and A. Phadke, "Wide-area monitoring, protection, and control of future electric power networks," *Proc. of the IEEE*, vol. 99, no. 1, pp. 80–93, Jan. 2011.

[7] Y.-F. Huang, S. Werner, J. Huang, N. Kashyap, and V. Gupta, "State estimation for the future grid," *IEEE Signal Process. Mag.*, 2012, to be published.

[8] N. Kashyap, S. Werner, and Y.-F. Huang, "Event-triggered multi-area state estimation in power systems," in *Proc. 4th IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Process. (CAMSAP), San Juan, Peurto Rico*, Dec. 2011, pp. 133–136.

[9] B.M. Weedy and B.J. Cory, *Electric Power Systems*, John Wiley Sons, New York, 1998.

[10] Ned Mohan, *First Course on Power Systems*, Minnesota Power Electronics Research & Education, Minneapolis, 2006.

[11] N. Balu, T. Bertram, A. Bose, V. Brandwajn, G. Cauley, D. Curtice, A. Fouad, L. Fink, M.G. Lauby, B.F. Wollenberg, and J.N. Wrubel, "On-line power system security analysis," *Proc. IEEE*, vol. 80, no. 2, pp. 262–282, Feb. 1992.

[12] F.C. Schweppe and J. Wildes, "Power system static-state estimation, Part I," *IEEE Trans. Power App. Syst.*, vol. 89, no. 1, pp. 120–135, Jan. 1970.

[13] A. Monticelli, "Electric power system state estimation," *Proc. IEEE*, vol. 88, no. 2, pp. 262–282, Feb. 2000.

[14] A. Monticelli and A. Garcia, "A hybrid state estimator: solving normal equations by orthogonal transformations," *IEEE Trans. Power App. Syst.*, vol. 104, no. 12, pp. 3460–3468, Dec. 1985.

[15] A.S. Debs and R.E. Larson, "A dynamic estimator for tracking the state of a power system," *IEEE Trans. Power App. Syst.*, vol. 89, no. 7, pp. 1670–1678, Sep. 1970.

[16] F.C. Schweppe and R.D. Masiello, "A tracking static state estimator," *IEEE Trans. Power App. Syst.*, vol. 90, no. 3, pp. 1025–1033, May 1971.

[17] A.M. Leite da Silva, M.B. Do Coutto Filho, and J.F. de Queiroz, "State forecasting in electric power systems," *IEE Proc. Gener. Transm. Distrib.*, vol. 130, no. 5, pp. 237–244, Sep. 1983.

[18] G. Durgaprasad and S.S. Thakur, "Robust dynamic state estimation of power systems based on M-estimation and realistic modeling of system dynamics," *IEEE Trans. Power Syst.*, vol. 13, no. 4, pp. 1331–1336, Nov. 1998.

[19] G. Valverde and V. Terzija, "Unscented Kalman filter for power system dynamic state estimation," *IET Generation, Transmission Distribution*, vol. 5, no. 1, pp. 29–37, Jan. 2011.

[20] M. Brown Do Coutto Filho and J.C.S. de Souza, "Forecasting-aided state estimation - Part I & II," *IEEE Trans. Power Syst.*, vol. 24, no. 4, pp. 1667–1677, Nov. 2009.

[21] A. Gómez-Expósito, A. de la Villa Jaén, C. Gómez-Quiles, P. Rousseaux, and T. Van Cutsem, "A taxonomy of multi-area state estimation methods," *Electr. Power Syst. Res.*, vol. 81, no. 4, pp. 1060–1069, 2011.

[22] G.N. Korres, "A distributed multi-area state estimation," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 73–84, Feb. 2011.

[23] J.F. Marsh, "Structure of measurement jacobian matrices for power systems," *IEE Proc. Gener. Transm. and Distrib.*, vol. 136, no. 6, pp. 407–412, Nov. 1989.

[24] B.M. Bell and F.W. Cathey, "The iterated Kalman filter update as a Gauss-Newton method," *IEEE Trans. Autom. Control*, vol. 38, no. 2, pp. 294–297, Feb. 1993.

[25] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. Int. Conf. on Inform. Process. in Sensor Networks*, Los Angeles, CA, Apr. 2005, pp. 63–70.

[26] F.S. Cattivelli and A.H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2069–2084, Sep. 2010.

[27] U.A. Khan and J.M.F. Moura, "Distributing the Kalman filter for large-scale systems," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4919–4935, Oct. 2008.

[28] Thorp J.S. Phadke, A.G., *Synchronized Phasor Measurements and Their Applications*, Springer, 2008.

[29] A.G. Phadke, T. Hlibka, M. Ibrahim, and M.G. Adamiak, "A microcomputer based symmetrical component distance relay," in *Proc. IEEE Conf. Power Ind. Comput. Applicat., PICA-79*, May 1979, pp. 47–55.

[30] A.G. Phadke, "Synchronized phasor measurements in power systems," *IEEE Comput. Appl. Power*, vol. 6, no. 2, pp. 10–15, Apr. 1993.

[31] R.O. Burnett Jr., M.M. Butts, T.W. Cease, V. Centeno, G. Michel, R.J. Murphy, and A.G. Phadke, "Synchronized phasor measurements of a power system event," *IEEE Trans. Power Syst.*, vol. 9, no. 3, Aug. 1994.

[32] W. Yuill, A. Edwards, S. Chowdhury, and S.P. Chowdhury, "Optimal pmu placement: A comprehensive literature review," in *IEEE Power and Energy Soc. General Meeting*, Jul. 2011, pp. 1–8.

[33] G. Valverde, S. Chakrabarti, E. Kyriakides, and V. Terzija, "A constrained formulation for hybrid state estimation," *IEEE Trans. Power Syst.*, vol. 26, no. 3, pp. 1102–1109, Aug. 2011.

[34] S. Chakrabarti and E. Kyriakides, "PMU measurement uncertainty considerations in WLS state estimation," *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 1062–1071, May 2009.

[35] Liang Zhao and A. Abur, "Multi-area state estimation using synchronized phasor measurements," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 611–617, May 2005.

[36] Weiqing Jiang, V. Vittal, and G.T. Heydt, "A distributed state estimator utilizing synchronized phasor measurements," *IEEE Trans. Power Syst.*, vol. 22, no. 2, pp. 563–571, May 2007.

[37] T. Yang, H. Sun, and A. Bose, "Transition to a two-level linear state estimator part ii: Algorithm," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 54–62, Feb. 2011.

[38] E. Fogel and Y. Huang, "Reduced-order optimal state estimator for linear systems with partially noise corrupted measurement," *IEEE Trans. Autom. Control*, vol. 25, no. 5, pp. 994–996, Oct. 1980.

[39] Jack E. Volder, "The cordic trigonometric computing technique," *IRE Trans. on Electron. Comput.*, vol. EC-8, no. 3, pp. 330–334, Sep. 1959.

[40] Richard D. Christie, "Power systems test case archive," `http://www.ee.washington.edu/research/pstca/pf14/pg_tca14bus.htm`.

[41] R.D. Zimmerman, C.E. Murillo-Sánchez, and R.J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Systems,*, vol. 26, no. 1, pp. 12–19, Feb. 2011.

[42] U.S.-Canada Power System Outage Task Force, "Final report on the august 14th blackout in the United States and Canada," Tech. Rep., April 2004.

[43] S. Gollamudi, S. Nagaraj, S. Kapoor, and Yih-Fang Huang, "Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size," *IEEE Signal Process. Lett.*, vol. 5, no. 5, pp. 111–114, May 1998.

[44] Diniz, *Adaptive filtering : algorithms and practical implementation*, Springer, New York, 2008.

[45] F. Schweppe, "Recursive state estimation: Unknown but bounded errors and system inputs," *IEEE Trans. Autom. Control*, vol. 13, no. 1, pp. 22–28, Feb. 1968.

[46] Eli Fogel and Y.-F. Huang, "On the value of information in system identification - bounded noise case," *Automatica*, vol. 18, pp. 229–238, Mar. 1982.

[47] S. Dasgupta and Yih-Fang Huang, "Asymptotically convergent modified recursive least-squares with data-dependent updating and forgetting factor for systems with bounded noise," *IEEE Trans. Inf. Theory*, vol. 33, no. 3, pp. 383–392, May 1987.

[48] S. Nagaraj, S. Gollamudi, S. Kapoor, and Y.-F. Huang, "BEACON: an adaptive set-membership filtering technique with sparse updates," *IEEE Trans. Signal Process.*, vol. 47, no. 11, pp. 2928–2941, Nov. 1999.

[49] S. Gollamudi, S. Kapoor, S. Nagaraj, and Y.-F. Huang, "Set-membership adaptive equalization and an updator-shared implementation for multiple channel communications systems," *IEEE Trans. Signal Process.*, vol. 46, no. 9, pp. 2372–2385, Sep. 1998.

[50] L. Guo and Y.-F. Huang, "Frequency-domain set-membership filtering and its applications," *IEEE Trans. Signal Process.*, vol. 55, no. 4, pp. 1326–1338, Apr. 2007.

[51] P.S.R. Diniz and S. Werner, "Set-membership binormalized data-reusing LMS algorithms," *IEEE Trans. Signal Process.*, vol. 51, no. 1, pp. 124–34, Jan. 2003.

[52] J.R. Deller, "Set membership identification in digital signal processing," *IEEE ASSP Mag.*, vol. 6, no. 4, pp. 4–20, Oct. 1989.

[53] G. Valverde and V. Terzija, "PMU-based multi-area state estimation with low data exchange," in *Proc. IEEE Innovative Smart Grid Technologies Conference Europe (ISGT Europe)*, Oct. 2010, pp. 1–7.

[54] D.P. Spanos and R.M. Murray, "Distributed sensor fusion using dynamic consensus," in *Proc. IFAC World Congress*, 2005.

[55] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proc. 46th IEEE Conf. Decision and Control, Dec.*, 2007.

[56] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Approximate distributed Kalman filtering in sensor networks with quantifiable performance," in *Proc. Int'l Conf. Information Processing in Sensor Networks (IPSN)*, 2005.

[57] S. Werner, M. Mohammed, Yih-Fang Huang, and V. Koivunen, "Decentralized set-membership adaptive estimation for clustered sensor networks," in *Proc. IEEE Int'l Conf. Acoust., Speech and Sig. Processing*, Apr. 2008, pp. 3573–3576.

[58] U.A. Khan and A. Jadbabaie, "On the stability and optimality of distributed Kalman filters with finite-time data fusion," in *Proc. American Control Conf.*, June 2011, pp. 3405–3410.