NTNU
Norwegian University of
Science and Technology

A Substation Level State Estimator for Local Data Processing

Algorithms for Power System Monitoring

Bård Haga Bringeland

Master of Energy and Environmental Engineering
Submission date: June 2017
Supervisor: Vijay Vadlamudi, IEL

Norwegian University of Science and Technology
Department of Electric Power Engineering

Abstract

Modelling power systems with substation details in State Estimation (SE) has several advantages. An increased measurement redundancy is possible, because the measurements can be evaluated individually instead of being aggregated to the bus-branch level. Conventional SE methods, relying on bus-branch models, cannot provide an efficient topology error processing. Though Generalised State Estimation (GSE) overcomes this limitation, with its built-in ability to estimate the state of the system by using node-breaker models, the problem size and solution times render it infeasible to build an estimator based on centralised GSE for real-time monitoring of large power systems. A reasonable trade-off for the design of an efficient state estimator for real-time monitoring would be to use a two-level method:

- Level 1: In the substation level, a local SE is performed by utilising only *some of the salient* features of the GSE algorithm.
- Level 2: The validated pre-processed measurements obtained as output of Level 1 are then fed to a SE processor at the Transmission System Operator (TSO)-level, which runs on the conventional SE algorithm.

The central research question investigated in this thesis is whether or not a suitably designed substation level state estimator, naturally restricted in size, is able to retain the same advantages as the centralised state estimator based on GSE with respect to accuracy, bad data, and topology error processing.

Based on extensive literature survey and critical analysis of its consequent findings, this thesis provides the conceptual basis for a two-level state estimation: linear and computationally non-demanding substation level estimation, based on GSE, integrated with conventional state estimation at the TSO-level. Though the designed framework is mostly theoretical in nature,

conceptual tests have been conducted for demonstration and preliminary testing, paving the way for addressing further identified research questions in this area. A modified 14-bus IEEE test system has been employed as the case study of choice for conducting the relevant investigations for realising the proposed two-level state estimator.

The proposed implementation logic and design choices for the two-level state estimator have been presented and executed in MATLAB, including the program needed to perform conceptual testing.

Sammendrag

Inkluderingen av nettstasjonsdetaljer i kraftsystemmodellen har flere fordeler innen tilstandsestimering (State Estimation). En økt måleredundans er mulig, fordi målingene vurderes individuelt, i stedet for å måtte aggregeres til *bus-branch*-nivå. Konvensjonell tilstandsestimering, som er avhengig av bus-branch-modeller, kan ikke gi en effektiv behandling av topologifeil. Selv om generalisert tilstandsestimering (Generalised State Estimation, GSE) løser denne begrensningen, med sin evne til å estimere tilstanden til nett ved bruk av *node-breaker*-modeller, gjør problemstørrelsen og løsningstiden det umulig å bygge en estimator basert på sentralisert GSE for sanntidsovervåking av store kraftsystemer. En rimelig avveining i utformingen av en effektiv tilstandsestimator for sanntidsovervåking vil være bruk av en tonivåmetode:

- Nivå 1: En lokal tilstandsestimering utføres lokalt, på nettstasjonsnivå, ved bruk av *kun noen* av de fremtredende egenskapene i GSE.
- Nivå 2: De validerte, forbehandlede målingene som mottas fra nivå 1 blir deretter matet til systemoperatørnivå (TSO-nivå), hvor en konvensjonell tilstandsestimering blir utført.

Det sentrale forskningsspørsmålet som undersøkes i denne oppgaven er om en hensiktsmessig utformet tilstandsestimator på nettstasjonsnivå, som er naturlig begrenset i størrelse, er i stand til å oppnå de samme fordelene som den sentraliserte GSEen, med hensyn til nøyaktighet og behandling av feilaktige data og topologifeil.

Basert på en omfattende litteraturstudie og kritisk analyse av resulterende funn, legger denne oppgaven et teoretisk grunnlag for en tonivåestimering: lineær og rask lokal estimering på nettstasjonsnivå, basert på GSE, kombinert med en konvensjonell tilstandsestimering på TSO-nivå. Selv om det utformede rammeverket for det meste er teoretisk, har det blitt utført konseptuelle tester, for demonstrering og som foreløpig testing, som baner vei for videre identifiserte spørsmål

i forskningsfeltet. En modifisert versjon av IEEEs 14-samleskinners testsystem har blitt benyttet i studien for å gjennomføre relevante undersøkelser for realisering av den foreslåtte tonivåestimatoren.

De foreslåtte logikk- og designvalg i implementeringen av tonivåestimatoren har blitt presentert og utført i MATLAB, inkludert programmet nødvendig for utførelse av konseptuelle tester.

Preface

This thesis is submitted to the Department of Electric Power Engineering at the Norwegian University of Science and Technology in partial fulfilment of the requirements for the degree of Master of Science.

I wish to thank my supervisor during this project, Associate Professor Vijay Venu Vadlamudi, for teaching me the scientific mindset, for encouragement through the whole year, and for genuinely caring for me and my fellow students. With more rigid guidance, I could have taken a straighter path; but driven by curiosity and wondering, I found a path of my own.

Thanks also to my fellow students, for all the fun and hard work of these past five years. I could not have managed it without you.

Thanks are also due to my parents, for their never-ending support and encouragement.

My final thanks goes to my girlfriend, who has made *me* a better person and *my life* so much better. I owe you everything.

Acronyms

BD Bad data

CT Current transformer

EMS Energy management system

EHV Extra high voltage

LF Load flow

LV Low voltage

HV High voltage

MASE Multi-area state estimation

MCS Monte carlo simulations

PMU Phasor measurement units

SE State estimation

SCADA Supervisory control and data acquisition

SLSE Substation level state estimation

SSE Static state estimation

TSO Transmission system operator

VT Voltage transformer

WLS Weighted least squares

List of Figures

List of Tables

Contents

Chapter 1

Introduction

State Estimation (SE) software determines the current operation point of a power system, i.e., all the power flows, generation, load and voltages. The current operating point, the state, is necessary to perform all the analysis functions of the Energy Management System (EMS), such as contingency analysis and optimal power flow, making the SE software crucial in ensuring a secure and economically efficient operation of the power system.

Static State Estimation (SSE) is the conventional method for SE, providing an estimate of the system state at an instant in time, given all the voltage and power measurements at that same instant. Grid topology and parameters are assumed to be correct, measurement errors are assumed stemming only from white noise, and so the widely applied Weighted Least Squares (WLS) method can be utilised to achieve the optimal estimate of the state.

Gross errors in the measurements, the so-called Bad Data (BD), are identified only after the filtering of small measurement errors. Bad data detection is a necessary and vital function of SE software, because of the relatively frequent measurement failures that occur in big and wide-spread power systems.

Topology errors stem from erroneously reported Circuit Breaker (CB) statuses, which in turn lead to the state estimation being applied for the wrong network model and hence giving a deteriorated solution. Generalised State Estimation (GSE) is an extension of conventional state estimation, utilising the more detailed node-breaker network model and treats circuit breaker statuses as stochastic measurements.

1.1 Background

New, more detailed network models and the corresponding SE methods have been developed since the 90s, namely, models including substation topology and the methods needed to solve them. These have received some, but limited, research focus. However, some trends in the area of power system monitoring that support a renewed interest are:

- increased focus on interconnection and seamless operation of interconnected power systems, through, e.g., the EU project PEGASE (Pan European Grid Advanced Simulation and state Estimation).
- increased availability of accurate measurements from Phasor Measurement Units (PMUs).
- digitisation of substations through the standard IEC 61850 and increased substation computational power.

The level of detail in the system model of GSE, complicates its application on full-scale networks. Lately, GSE has been envisioned for a local implementation in the power substation, naturally limiting the problem size. Within the substaton, raw measurements could be locally pre-processed, so that only a limited, but verified, measurement set is sent to the EMS's central estimator. Local data redundancy and easy substation-internal communication, supported by the trend of substation digitalisation, may also render possible the use of an extended measurement set.

The goals of employing such a Substation Level State Estimation (SLSE) could be

1. improving topology error processing, hence increasing the *reliability* of both the estimate and of the network topology;
2. provide fast, local processing of both bad data and topology errors, rendering new monitoring schemes feasible;
3. possibly improving bad data processing, hence increasing the *reliability* of the estimate; and
4. possibly improving the estimators resistance to noise, hence increasing the *accuracy* of the estimate.

Both improving the state estimate and improving the estimated topology improves the real-time database, which is used to build the models for all other EMS functions, e.g., load flow models for contingency analysis.

1.1.1 Problem Statement

Use of detailed modelling of substation topology with Generalised State Estimation can improve error processing. Would a local application, namely a substation level state estimator, be expedient to achieve the same advantages? If so, how can such an estimator be designed for realisation?

1.2 Objectives

The main objectives of this Master's project, in accordance with the problem statement above, are as follows:

1. Provide an extensive review of relevant aspects of the vast SE literature, especially regarding GSE.
2. Provide an overview of power substation monitoring, focusing on aspects relevant to SLSE.
3. Provide an implementation of an SLSE, highlighting important properties and aspects.
4. Through conceptual tests, demonstrate the execution of the implemented SLSE and investigate its capabilities for local processing of bad data, both analogue and status, and also for noise resistance.

1.3 Structure of the Report

The report is structured as follows.

Chapters 2 and 3 provide the theoretical basis for all discussions, for the non-expert reader. Chapter 2 provides a review of relevant SE literature. In Chapter 3 the digital substation and its measurement and communications systems are presented.

In Chapter 4 the application of a node-breaker model in state estimation is presented. Earlier approaches receive a critical evaluation, providing the theoretical basis for the design of a new SLSE.

Chapter 5 describes the design and implementation of the proposed SLSE, so as to provide directions for later development. In Chapter 6, the properties and error processing capabilities of the SLSE are demonstrated. Chapter 7 provides concluding remarks and suggestions for future work.

Chapter 2

Power System State Estimation

For establishing narrative clarity and with an aim to make this thesis a complete and independent unit in and of itself, a pedagogical treatment of introductory aspects of SE is presented in this chapter.

First, the general concept of SE is introduced, exemplified through the classical approach to the widely deployed Static State Estimation (SSE). Subsequently an important SE subsidiary, which is still mostly restricted to the research-domain, is introduced and discussed, viz., Multi-Area State Estimation (MASE).

Remark: This work builds on the literature review carried out as part of specialisation project TET4520, and as such there is extensive reproduction/usage of the content therefrom.

2.1 Introduction

The state of a power system is the set of complex voltages at every bus in the system. It is called the state of the system because every other operational quantity, such as power flow and voltage at all points, can be calculated from it (assuming topology and branch parameters are known). For proper monitoring of the grid, these voltages should be determined. Presently, that is not achieved through direct measurement, because, unlike the voltage magnitude, the voltage angle can not be determined by simply analysing the voltage waveform.

2.1.1 The need for estimation

Direct measurements are inevitably prone to uncertainty, which issue some treatment. Also, during measurement or communication failures, some state variables could end up being erroneous or unavailable to the operators. Of these reasons, the state is *not* measured directly, but is instead *estimated* on the basis of all the measurements taken in the power system.

The State Estimation (SE) algorithm determines the most likely state, given all the available measurements. Traditionally, measurements of line power flow (active and reactive), power injections, and voltage and current magnitudes are taken. Uncertainties are introduced due to measurement noise and errors in sensors, communication, model and coordination. This makes the measurement set partly self-contradictory – there will simply not exist any state that perfectly fits all the measured voltages, power flows and currents.

For an intuitive understanding, SE can be compared with Load Flow (LF) analysis. Although both procedures produce the state as output, the input of LF is certain (deterministic), and exactly the minimum of information needed. The input for a SE, on the other hand, is not deterministic but stochastic. To produce a good estimate when the input is stochastic, additional measurements over the bare minimum are needed (overdetermined problem), i.e., there has to be redundancy in measurements. The function of a SE may hence be described as *filtering* the measurements for random noise and gross errors, thereby exploiting the redundancy in measurements. *State filtering* is used interchangeably with SE in some literature.

2.1.2 State Estimation in the Energy Management System

SE, first proposed by Schweppe in [6], is a very central function of power system operation, because it makes the state of the power system readily available to the operator and all other functions in the control centre (cf. Fig. 2.1). From the estimated states the real-time network models are built, which are utilised by Energy Management System (EMS) functions like contingency analysis, stability analysis, optimal power flow, short-circuit analysis and controlled switching [7]. For that reason, all power system control centres EMSs today have SE capabilities.

Proposed and implemented already in the 1970s, some might consider SE a mature research field, but new challenges are arising constantly. The fast deployment of intermittent renewable

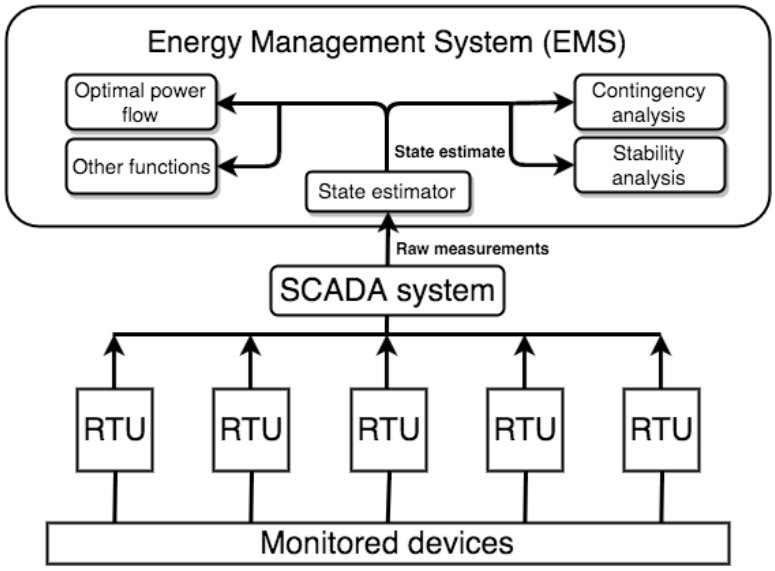


Figure 2.1: Monitoring scheme of power systems [1]. The SCADA system collects measurements from substations, which is sent to the EMS. The SE is the intermediate process that transforms the output of the SCADA system (raw measurements) to the input of the EMS functions (the state).

energy generation, and distributed generation and storage changes the conditions for power system operation. Also, deployment of Phasor Measurement Units (PMUs) has big implications on SE research. Conventional measurement devices, from which the measurements are collected by the Supervisory Control And Data Acquisition (SCADA) system, are only able to measure voltage magnitudes, while the phase angles must be estimated. With PMUs, the direct measurement of voltage phase angles (and magnitudes) is made possible, both fast and accurately [8] (cf. Section. 2.4). A purely PMU-based SE, with phasor measurements of all system buses, was proposed already in the 1980s [9]. This would require only a simple SE algorithm. But, complete coverage with the expensive PMUs will nevertheless be economically inefficient, and so SE will still be relevant for most applications in the foreseeable future. Yet, the new measurements can be utilised for dramatic improvement of SE accuracy and applicability.

2.1.3 Features of State Estimation Software

The SE core procedure is not the only function of a SE software, and these will now be explored. Typically, SE is said to have five functions [1]:

1. Topology processor: Gathers information about the circuit configuration (e.g., transformer

and switch settings) to construct the one-line diagram.

2. Observability analysis: Determines if there is enough measurement data available to obtain a SE for the whole system, or, alternatively, what observable islands exist.

3. State estimation core procedure (often denoted state filtering): Produces the state estimate.

4. Bad data (BD) detection and correction: Is a vital function of the real-life SE software, because it roots out and eliminates any data that is likely to be corrupted, so that these BD do not affect the solution adversely.

5. Detection of errors in parameters and topology: Checks if the information collected from the breakers and switches are likely to be correct – a breaker could for example have opened without reporting it – and that all the parameters are likely to be correct.

A SE software should act as a filter for smoothing out measurement noise, i.e., small, unavoidable errors, but also gross errors (BD) due to malfunction in measurement device or acquisition system. The first goal is achieved by the SE core algorithm, while the latter is usually achieved separately, by the BD detection and correction algorithm.

It should be noted that the electrical measurements (power, voltage, current) are modelled as stochastic data, while status data (switch position, tap position, etc.) and network parameters (impedances) are assumed deterministic by the topology processor, but they are still checked for big discrepancies.

Both BD detection, and parameter and topology error detection require redundant measurements. Where there are no more measurements than strictly needed, the measurements are denoted as critical, i.e., the removal of a critical measurement would render the system unobservable. Errors, small or gross, in such critical measurements will bias the estimate [1].

Robustness is a very important feature of the practical estimator, and describes its ability to converge and produce an acceptable solution for a wide range of circumstances, such as topology, load, measurement configuration, presence of large BD, etc. [10]. Both the choice of SE solution method and auxiliary functions, like BD detection, affect robustness.

2.2 Static State Estimation

SSE aims to provide the system state at an instant of time. The state to be estimated, the complex voltages, is usually expressed in polar form, so that the state vector consists of all the bus voltage magnitudes and phase angles of the system:

$$\boldsymbol{x} = \begin{bmatrix} \theta_2 & \theta_3 & \dots & \theta_n & |V_1| & |V_2| & \dots & |V_n| \end{bmatrix}^T \tag{2.1}$$

The phase angle at bus 1, θ_1, is taken as a reference and set to zero, giving a total of $N = 2n-1$ state variables.

An estimation is basically an "optimised guess", given some information with uncertainty. The SE problem is therefore an optimisation problem, which can be formulated and solved in many different ways. For a hands-on intuitive approach, the classical problem formulation and solution method of SSE is presented in the following subsections. This will form the basis for the discussion of other aspects of SE.

2.2.1 Classical Weighted Least Squares Problem Formulation

The power system is modelled using the bus admittance matrix ($\boldsymbol{Y}_{bus}$) representation known from LF studies.

The classical SE solution method is the Weighted Least Squares (WLS) algorithm (cf. [1]). The measurement model used is

$$\boldsymbol{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_M \end{bmatrix} = \begin{bmatrix} h_1(x_1, x_2, ..., x_n) \\ h_2(x_1, x_2, ..., x_n) \\ \vdots \\ h_M(x_1, x_2, ..., x_n) \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_M \end{bmatrix} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{e}, \tag{2.2}$$

where

- $\boldsymbol{z}$ is the measurement vector ($M \times 1$), i.e., the actual values measured (such as power flows).
- $\boldsymbol{h}(\boldsymbol{x})$ is the measurement function vector ($M \times 1$), i.e., the expected measurements according to the power flow equations, for a given $\boldsymbol{x}$ (cf. App. A for the complete set of functions).

Every row in $\boldsymbol{h}(\boldsymbol{x})$ corresponds to the measurement of the same row in $\boldsymbol{z}$.

- $\boldsymbol{e}$ is the measurement error vector ($M \times 1$) (equivalent to the residual vector), which is assumed to be normally distributed noise, with zero mean and diagonal covariance matrix $\boldsymbol{R} = diag(\sigma_i^2)$.

The problem consists of finding the minimum of the scalar sum of squared residuals,

$$J(x) = \sum_{\forall j} [w_j \cdot (z_j - h_j(\boldsymbol{x}))^2] = \sum_{\forall j} [w_j \cdot r_j^2] \tag{2.3}$$

where

- j is the index $\{1, 2, ..., M\}$ of a measurement in the measurement vector $\boldsymbol{z}$.
- $r_j = z_j - h_j(x)$ is the residual of measurement j, i.e., the deviation between measurement z_j and estimated measurement $h_j(\boldsymbol{x})$.
- w_j is the weight of measurement j, used to prioritise minimisation of residuals r_j of accurate confidence measurements.

In vector form this becomes

$$J(x) = [\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})]^T \boldsymbol{W} [\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})] \tag{2.4}$$

where $\boldsymbol{W}$ is a diagonal matrix with the weights on its diagonal [1]. To reach maximum likelihood (the most likely state) the weight of the individual measurement is chosen as the inverse of that measurement's variance, determined from the uncertainty of the whole measurement chain [1], or equivalently, the weight matrix is chosen as the inverse of the diagonal covariance matrix:

$$\boldsymbol{W} = \boldsymbol{R}^{-1} = diag(\frac{1}{\sigma_i^2}) = diag(w_i) \tag{2.5}$$

2.2.2 Classical Weighted Least Squares Solution Method

The solution algorithm is traditionally iterative, using Gauss-Newton method, similar to the solution method used for LF studies, with which the reader may likely already be familiar with. Simply put, the iterations continue with new estimations of the state $\boldsymbol{x}$ until the residual vector,

$\boldsymbol{r} = \boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})$, is small enough – or in other words, until the difference between the expected measurement values (based on the newest state estimate), $\boldsymbol{h}(\boldsymbol{x})$, and the actual measurements, $\boldsymbol{z}$, is small enough.

The non-linear vector of measurement functions, $\boldsymbol{h}(\boldsymbol{x})$, is linearised using the Jacobian with respect to the state vector $\boldsymbol{x}$. This Jacobian matrix ($M \times N$), $\boldsymbol{H}(\boldsymbol{x})$, is evaluated at the best-estimate of the state of every iteration, again very similar to the Newton-Raphson method for LF. The iterative update, for iteration k, is

$$\Delta \boldsymbol{x_k} = \left[\boldsymbol{H}^T(\boldsymbol{x_k})\boldsymbol{W}\boldsymbol{H}(\boldsymbol{x_k})\right]^{-1}\boldsymbol{H}^T(\boldsymbol{x_k})\boldsymbol{W}[\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x_k})] \tag{2.6}$$

or, for simplicity,

$$\Delta \boldsymbol{x_k} = \left[\boldsymbol{H}_k^T\boldsymbol{W}\boldsymbol{H}_k\right]^{-1}\boldsymbol{H}_k^T\boldsymbol{W}[\boldsymbol{z} - \boldsymbol{h_k}] \tag{2.7}$$

which is the solution of the so-called normal equation [1]. Each new state solution is updated as

$$\boldsymbol{x_{k+1}} = \boldsymbol{x_k} + \Delta \boldsymbol{x_k} \tag{2.8}$$

The iterations continue until the state update, $\Delta \boldsymbol{x}_k$, is smaller than a given tolerance (cf. App. A for a step-by-step overview of the WLS solution method).

The result of the whole process is the newest estimated state (at time-step t),

$$\hat{\boldsymbol{x}}_t = \boldsymbol{x}_k \tag{2.9}$$

and the corresponding state covariance matrix (using standard notation)

$$\boldsymbol{\Sigma}_t = Cov(\hat{\boldsymbol{x}}_t) = \left[\boldsymbol{H}_k^T\boldsymbol{W}\boldsymbol{H}_k\right]^{-1} \tag{2.10}$$

2.2.3 Assumptions of State Estimation Methods

There are many assumptions made in the estimation just described, implicitly as well as explicitly, to theoretically guarantee the optimal estimate. The following discussion is on the assumptions of this particular SSE method, but they cover well the assumptions made for SE in

general.

Implicitly, since a single-frequency single-phase model is used, it is assumed that this model describes the static state power system completely. Thus, underlying modelling assumptions for the power system are [11]:

- current and voltage waveforms are purely sinusoidal with constant frequency;
- balanced and symmetrical three-phase system;
- system can completely be described from its positive sequence.

The input parameters are assumed deterministic and certain, as are the status data, which together determine the network model. For the electrical measurements, several assumptions are made, most of which are explicitly mentioned, but summarised here.

There is no synchronisation between measurement devices. The SSE simply scans the SCADA system for the newest measurements available. The measurements may therefore to a certain degree originate from different time instants. One assumption is therefore that the system is in a row of consecutive steady-state operating points (called a quasi-static assumption), so that the system does not change significantly between the non-synchronised measurements: The magnitude of currents and voltages are in other words assumed constant during the measurements.

For WLS estimation to be optimal, in general, the measurement error, $\boldsymbol{e}$, must be unbiased and normally distributed. The variance of the measurement errors are assumed to be known with certainty, although this may not be the case in practice. The variance of the whole measurement chain, from transducers to sampling and communication, could be used to determine the variance. A fixed variance (e.g., in MW) is used in SE for a measurement, no matter the magnitude of the measurement, when in reality the error may depend heavily on the measurement magnitude (in %) [12]. A more realistic noise-model could be the two-component model, with both a fixed and a magnitude-dependent component [13].

The practical extra assumption, also described earlier, is that there is no correlation between measurement errors at different time-steps (classified as white noise). Also, it is assumed that there is no correlation between errors of different measurements at the same time-step, resulting in a diagonal co-variance matrix. The co-variance matrix is required to achieve optimal weighting, with $\boldsymbol{W} = [\boldsymbol{R}]^{-1}$.

Summarised, the measurement assumptions are:

- quasi-static system, or equivalently, constant state variables during one time-step;
- unbiased and normally distributed measurement errors;
- no measurement error correlation in time (white noise);
- no measurement error correlation between different measurements (diagonal covariance matrix);
- variance of measurements can be determined.

2.2.4 Issues and Improvements of the Classical Method

The practical considerations relating to numerical convergence, computation effort and estimate bias, will briefly be discussed. Also, some appropriate improvements to mitigate the issues of the classical methods are discussed.

The classical WLS method, the normal equation approach, using (2.7) as presented, though giving good results, is very computationally tedious. The biggest burden is the computation of the inverse of the gain, $\boldsymbol{G} = [\boldsymbol{H}_k^T \boldsymbol{W} \boldsymbol{H}_k]$, which is being inverted in every iteration.

Great efforts have been made to lighten the computation. This was earlier motivated by the limited computer capacity, but is now motivated by speed requirements and growing control areas. The standard method is to observe the sparsity and symmetry of the gain matrix, $\boldsymbol{G}$, and use the Cholesky decomposition, where the gain is decomposed to $\boldsymbol{G} = \boldsymbol{L}\boldsymbol{L}^T$ (an LU factorisation, where L is a triangular matrix) [1]. The inversion used to solve the NE can then be solved more efficiently by backward-forward substitution.

Another big issue linear equations systems ($\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$) is *ill-conditioning*, which indicates that small errors in the coefficient matrix ($\boldsymbol{A}$) or right hand-side vector ($\boldsymbol{b}$) results in big errors in the solution vector ($\boldsymbol{x}$) [1]. The more singular a matrix is, the more ill-conditioned its associated system will be [1].

An ill-conditioned gain matrix produces numerical issues in NE solution of SE [1], resulting, possibly, in convergence problems and inaccurate results. Some commonly described sources of ill-conditioning in the NE are [14]:

- Very different weighting factors.
- Short and long lines incident to the same bus.
- A large proportion of injection measurements.

Zero-injection virtual measurements are used to model buses which have no generators or loads connected, so that they definitely have zero power injected. The so-called pseudo-measurements are sometimes necessary to ensure observability, when the real measurements are unavailable, but they are based on for example historical loads, making them little accurate. Very high weights on virtual measurements are desirable to enforce the zero-injection constraint. But a combination of very accurate measurements and very inaccurate ones in the same measurement set, like the virtual measurements and pseudo-measurements, gives very different scales of weights.

In the extreme case, if virtual measurements were given infinite weights, all other measurements would effectively be ignored. Since the virtual measurements normally are not alone enough to obtain observability, the result of such a weighting would be a gain matrix extremely close to singularity, i.e., the system would be extremely ill-conditioned.

In [1], examples and theoretical reasons are given for how the different sources produce ill-conditioning.

Many methods for reducing the problem of ill-conditioning have been proposed, so-called numerically robust methods. The more stable orthogonal factorisation instead of Cholesky decomposition can be applied for the solution of the normal equation (2.7) [14]. Other methods no longer apply the normal equation directly, e.g., representing zero-injections with equality-constraints instead of measurements [14]. These numerically robust methods are discussed further in Section 4.2.2.

Conventional SE removes BD after the filtering, and runs the filtering anew when the BD has been removed. However, with grave BD the SE might not even converge and with a sufficiently small BD, when the BD cannot be properly identified and removed, the estimate is biased.

Robust SE methods were developed to avoid these problems. By enforcing other weighting functions than the least squared error, which weights outliers high, BD is automatically down-weighted or ignored. A well-known and straight-forward example of these robust methods is

the Least Absolute Value (LAV) method. In LAV the errors are weighted linearly, so that BD that stand out are weighted less than with WLS. Since the problem is linear, it can be solved using Linear Programming (LP) without iterations [15].

It is obvious that many background techniques exist in practical SSE which are not presented in the simplified example of 2.2.2. Issues with convergence, computational burden and minimisation of BD estimation biases have long been a focus of research, with important results in [1], [14], [16], [17], [18], but a detailed review has purposefully been left out of the scope of this thesis.

2.2.5 Inclusion of Phasor Measurements

With some PMUs already deployed, they could be used to improve the SE accuracy. Inclusion of phasor measurements in the SE measurement set is an impending improvement in control centres.

Some important aspects of PMUs are mentioned here. First, the voltage phasors are measured with the common Universal Time Constant (UTC) reference. The classical SE does not include any phase angle measurement and is therefore referenced to a reference bus, whose angle is arbitrarily set to zero (and thus is not included in the state vector). This can be handled by choosing one of the buses with a PMU as the reference, so that all other measured phasors can be re-referenced to that one [19].

Inclusion of phasor measurements in SSE have been proposed in two basic ways. The obvious way is by including the new measurements in the measurement vector, resulting in a so-called *mixed measurement vector*, which means that the existing SE software must be modified to include the new measurement functions [20]. The alternative approach is to include the phasors in a second stage. Using this two-stage approach, the old software (and often, dedicated hardware) performing the first-stage SE can be kept quite intact, while the second-stage SE is performed by new software [21]. The two-stage procedure is:

1. The conventional estimation by the existing software: A SE that results in a first state estimate, denoted $\hat{\boldsymbol{x}}^{(1)}$, and the covariance matrix of the state estimate, denoted $\boldsymbol{\Sigma}^{(1)}$ (according to (2.10)).

2. A linear SE by the new software: In addition to the phasor measurements, it uses results from the first step as pseudo-measurements, i.e., $\hat{\boldsymbol{x}}^{(1)}$ with the weightage of $[\boldsymbol{\Sigma}^{(1)}]^{-1}$.

The linearity of the second stage can be achieved by using rectangular voltage representation of the state vector, because the measurement functions become linear [21]. The second stage is thus non-iterative, which would make the additional software for PMU inclusion quite computationally light, while legacy software and hardware does the demanding part of the procedure.

Assuming that the errors only stem from measurement noise, these two methods should produce the same estimate, but the execution is quite different.

In a power system with a very high number of PMUs, the system could become completely observable using PMUs only. Direct measurement of state variables on all buses, gives a simple, linear SE solution (even with state vector in polar form). It must be noted that the PMUs measure not only bus voltages, but also currents on connecting branches. Therefore, not every bus would need a PMU to get complete observability of the system, but only about every third bus [22]. In this case, the state vector must be in rectangular form for the measurement functions to become linear [21]. Note that such a SE system could achieve a high enough update rate for tracking slow (electromechanical) dynamics, because of the fast SE algorithm and the relatively high update rate and precision of PMUs.

2.3 Bad Data Detection and Identification

BD detection and identification is a necessary and vital function of SE software, because of the relatively frequent measurement failures that occur in a big and wide-spread power system. Only introduced briefly in Section 2.1.3, some further considerations are added in the following.

The classical approach to BD detection and identification, namely residual analysis, i.e., analysis based on residual vector $\boldsymbol{r} = \boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{x}})$, is introduced in the following. For BD detection, which is the process of determining the presence of BD in the measurement set, the χ^2-test is applied. BD identification, which is the process of identifying the exact measurements that contain BD, the *largest normalised residual* (r^N) test is applied, which is also applicable to BD detection.

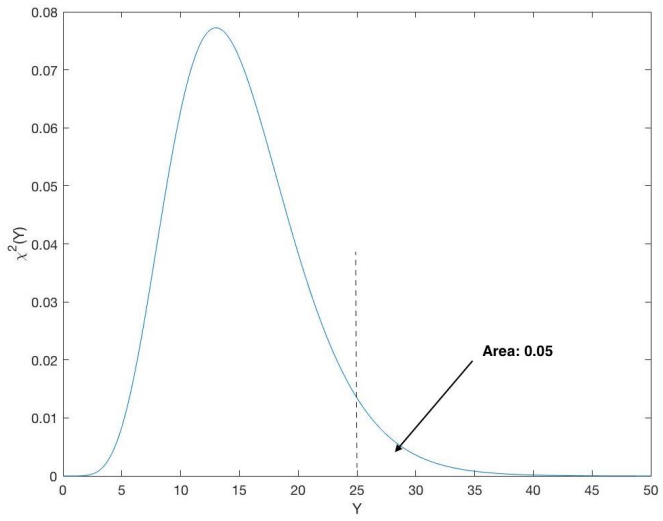


Figure 2.2: χ^2 distribution, with $N = 15$ degrees of freedom. For $p = 95\%$ confidence level, the upper limit of the χ^2-test is marked: $\chi^2_{15,0.95} = 25$ [1].

2.3.1 The χ^2 distribution

The χ^2 probability distribution can be obtained from normally distributed variables. If X_j is standard normally distributed,

$$X_j \sim N(0,1) \tag{2.11}$$

then any sum of squared X_j,

$$Y = \sum_{\forall j} X_j^2 \tag{2.12}$$

has a χ^2 distribution,

$$Y \sim \chi^2_U \tag{2.13}$$

where U is the number of linearly independent X_j variables.

If Y is assumed to follow a χ^2 distribution, its value should follow the distribution similar to that of Fig. 2.2. In the figure, the χ^2-distributed variable Y is $p = 95$ % likely to be less than the corresponding number $\chi^2_{N,p}$, which is what the well-known χ^2-test from mathematics relies on. It tests whether or not the the value of Y is higher than the threshold $\chi^2_{N,p}$, because such a value would be unlikely (in the case of Fig. 2.2 only 5 % likely). A value of Y exceeding the threshold, indicates that Y is not really χ^2-distributed, which further indicates that X_j is not really standard normally distributed.

2.3.2 The χ^2-test

The objective function (2.3) of the classical state estimation formulation is, as given in Section 2.2,

$$J(\boldsymbol{x}) = \sum_j W_{jj} r_j^2 = \sum_j \frac{r_j^2}{R_{jj}} \tag{2.14}$$

where r_j is the normally distributed residual (and a function of $\boldsymbol{x}$), with an approximate variance of R_{jj} [1].

It will in the following be shown that $J(\boldsymbol{x})$ is approximately χ^2 distributed. The ratio

$$\frac{r_j}{\sqrt{R_{jj}}} \tag{2.15}$$

is normalised, or in other words, it has a standard normal distribution. The objective function hence becomes

$$J(\boldsymbol{x}) = \sum_{j=1}^{m} \left(\frac{r_j}{\sqrt{R_{jj}}} \right)^2 \tag{2.16}$$

which is a sum of squared standard normally distributed quantities. Similar to (2.13), this means that J is χ^2 distributed,

$$J(\boldsymbol{x}) \sim \chi^2_{M-N} \tag{2.17}$$

with a degree of freedom of $M-N$ [1], where M is the total number of measurements and N is the number of state variables (cf. Section 2.2)

The χ^2-test can be applied to the detection of BD in SE. A $J(\boldsymbol{x})$ higher than the chosen threshold $\chi^2_{N,p}$ is too unlikely, indicating that $J(\boldsymbol{x})$ is most likely not χ^2 distributed, hence the residual r_j is not really normally distributed, implying that the measurement contains BD.

The procedure is straightforward, and follows three basic steps:

1. Use the solution of the SE, $\hat{\boldsymbol{x}}$, to calculate the final objective function value:

$$J(\hat{\boldsymbol{x}}) = \sum_{j=1}^{m} \frac{z_j - h_j(\hat{\boldsymbol{x}})}{R_{jj}} = [\boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{x}})]^T \boldsymbol{W} [\boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{x}})] \tag{2.18}$$

2. Determine from statistical tables the threshold value

$$\chi^2_{N,p} \tag{2.19}$$

given degree of freedom $N = m - n$ and confidence level p.

3. Test if

$$J(\hat{\boldsymbol{x}}) \geq \chi^2_{N,p} \tag{2.20}$$

i.e., if the value of the final objective function is less likely than $(1 - p)$, assuming $J(\hat{\boldsymbol{x}})$ is in fact χ^2 distributed. If yes, BD is suspected in the measurement set.

2.3.3 The largest normalised residual test

The output of the SE is the expected state vector, $\hat{\boldsymbol{x}}$, and its covariance matrix, from (2.10), is

$$\boldsymbol{\Sigma} = \left[\boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{W}\boldsymbol{H}\right]^{-1} \tag{2.21}$$

Applying conditional expectation on the measurement estimate, $\boldsymbol{h}(\hat{\boldsymbol{x}})$, the covariance matrix of the expected measurements $\boldsymbol{h}(\hat{\boldsymbol{x}})$ is found as

$$\boldsymbol{S} = \boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{H}^T \tag{2.22}$$

where all functions, such as $\boldsymbol{\Sigma}$, are evaluated at $\hat{\boldsymbol{x}}$ for a specific time-step. The covariance of the measurements, $\boldsymbol{z}$, is $\boldsymbol{R}$.

The residual, $\boldsymbol{r}$, is given as a function of the measurements and estimated measurements, $\boldsymbol{r} = \boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{x}})$. The covariance matrix of the residual vector hence becomes [23]:

$$\boldsymbol{\Omega} = \boldsymbol{R} + \boldsymbol{S} = \boldsymbol{R} + \boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{H}^T \tag{2.23}$$

The normalised residual hence becomes

$$r_j^N = \frac{r_j}{\sqrt{\Omega_{jj}}} \tag{2.24}$$

The largest normalised residual test can be described in five steps [14]:

1. Use the output of the SE process to calculate the residual covariance:

$$\boldsymbol{\Omega} = \boldsymbol{R} + \boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{H}^T \tag{2.25}$$

2. Calculate the normalised residuals:

$$r_j^N = \frac{r_j}{\sqrt{\Omega_{jj}}} \tag{2.26}$$

3. Identify the measurement l of all measurements $1, 2, ..., j, ...m$ with the largest normalised residual,

$$r_l^N = \max_{j=1,...,m} (r_j^N) \tag{2.27}$$

4. Test if

$$r_l^N \geq \epsilon \tag{2.28}$$

where ϵ is the threshold parameter. If yes, BD is detected and r_l^N is suspected.

5. Remove measurement l from the measurement set and go back to step 1.

Assuming there is no BD, r_j^N will be a standard normally distributed variable and a statistics table would show that $r_j^N \geq 3$ is only 1 % likely [1]. For a single BD in the measurement set (provided it is not a critical measurement), the largest normalised residual has been shown to belong to the erroneous measurement [14]. This extends also to the case of multiple BD, provided that there is negligible correlation between the impacted measurements [14]. Therefore, this identification method performs satisfactorily for single BD and for multiple but uncorrelated BD.

A gross error in a critical measurement will still result in a very low residual, because the error moves the state estimate to fit the erroneous measurement [24]. Therefore, residual analysis methods do not have the ability to detect BD among critical measurements.

2.4 Phasor Measurement Units

Traditionally, the measurement of phasors (both magnitude and angle) at distant locations has been very hard. Phasor angles do not make sense one by one, only angle *differences* have a physical meaning. Therefore some synchronisation is needed, and it must be very accurate. GPS technology has provided such means of synchronisation, which is being exploited in the PMU. A PMU has a GPS signal receiver, which synchronises the internal PMU clock every second to the UTC reference [8].

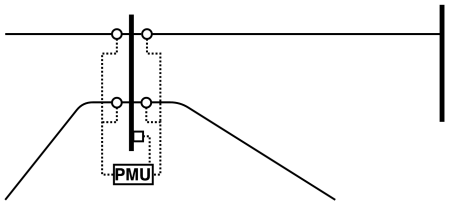


Figure 2.3: A PMU has multiple channels, measuring bus voltage and multiple branch currents.

The PMU samples the voltage and current signals at a very high frequency, e.g., 40 samples per period of the 50 Hz signal. From these samples, the amplitude and angle (referenced to UTC) of the 50 Hz signal can be accurately estimated [8]. A PMU placed on a bus can therefore provide the operator with both voltage phasor of the bus and current phasors of the connecting lines (cf. Fig. 2.3).

Measuring both voltage at a PMU-bus and current between a PMU-bus and a neighbouring bus, the voltage of the neighbouring bus is easily calculated, calculating the voltage drop using Ohm's law. Using these indirect measurements (which assumes knowledge of the line parameters), each PMU can be though of as measuring the voltage phasor of the bus on which the PMU is placed *and* the voltage phasors of the incident buses.

2.5 Multi-Area State Estimation

The power systems on which SE is conducted are generally not islands. They have borders to other systems, and neighbouring power system areas are getting more closely interconnected, both physically and functionally.

Ignoring the neighbouring systems would give a weak SE close to the border because of less measurement redundancy, which may render tie-line power flow estimates unreliable. More importantly, the surrounding areas of the control area must be modelled because the conditions outside but close to an area's borders are part of the input to EMS functions such as contingency analysis and any other function relying on LF [25], e.g., for a contingency on an external border bus.

These external systems could be replaced by simple equivalents, as is discussed in [26]. A more successful approach has been to model a significant part of the network explicitly, assuming the statuses of loads and generation in real-time and solving it either by a new state estimation (for which redundancy in the assumed external system data is allowed) or by a LF [25], or by a sequence of the two [27]. Alternative to making assumptions based on market information, nominal values, best guesses etc., measurement data could be exchanged periodically between neighbouring control areas, such as described in [28] for the Transmission System Operators (TSOs) of Sweden and Norway.

If, for example, SE is employed for the estimation of external system also, the border bus states already estimated are used as measurements with high weights, so that the external system solution adapts to those. In [25], it is reported that erroneous assumptions on the external system close to the borders have the greatest impact on the later analysis, which sounds reasonable. Reference [25] also reports that topology errors in the external system have a more adverse effect than errors in assumed pseudo-measurements, which is much more hard to discover; this calls for a coordination between control areas.

2.5.1 The concept of Multi-Area State Estimation

Market deregulation, increased share of renewables, and the increased need for situational awareness transcending control area borders, all stress the need for a system-wide SE [29]. For this "mega grid estimation", a centralised execution of the conventional SSE is not suitable. Firstly, it entails huge amounts of measurements to be sent to *one* control center (and processed there), which introduces communication latency, demands a lot from the infrastructure, and is computationally demanding. Also, an integrated European system for example, would consist of many TSOs' control areas. Each TSO maintains its grid model, collects measurements and acts

autonomously in a competitive environment, and thus is not willing to share a lot of data of a proprietary nature.

Today's practice of combining the TSOs' incoherent sub-system estimation results gravely limits accuracy of the system-wide estimation. This is largely due to use of different models of the external networks, tailored for the needs of the individual TSO, and also due to the lack of synchronicity of individual SE processes [12]. There is thus a need for Multi-Area State Estimation (MASE) methodology, with adaption of the local processes, to instrument an accurate coordination process, providing a more coherent estimation of the wide-area state, and hence more accurate tie-line power flow estimations.

MASE research was originally driven mostly by speed requirements and communication limitations of a single control centre, to allow near real-time monitoring of a single control area [10] – one control centre is dividing its system into sub-systems. Today, the available computational power already allows for real-time monitoring in a control area. Therefore, the driver for MASE research is rather the improvement of border bus estimations in interconnected power systems, while preserving proprietary data [30], as well as system-wide coherent situational awareness. The sub-systems are already given as the control areas, within an interconnected mega grid.

In the following sections a simple method for MASE is presented, followed by a discussion on the impact of phasor measurements.

2.5.2 A Classical Two-Level Estimator

A hierarchical method, i.e., with a centralised coordination of all sub-systems, is presented conceptually in the following, based on the work of Van Cutsem *et al.* [31]. Classified as a two-level method, the levels can be summarised as:

1. Sub-system level: A SSE is performed in each of the S non-overlapping sub-systems (cf. Fig. 2.4a), with an arbitrary phase angle reference bus.

2. Coordination level: The state variables at the border buses are re-estimated, again using SSE. Additionally, one sub-system reference bus is chosen as system-wide reference, so the phase angles of the sub-system reference buses are also estimated.

For sub-system s, the state vector at sub-system level SE is

$$\boldsymbol{x}_s = \begin{bmatrix} \boldsymbol{x}_s^{int} \\ \boldsymbol{x}_s^{b} \end{bmatrix} \tag{2.29}$$

or equivalently, $\boldsymbol{x}_s = [\boldsymbol{x}_s^{int^T}, \boldsymbol{x}_s^{b^T}]^T$, where state variables $\boldsymbol{x}_s^b$ are of the border buses, from which tie-lines connect to other areas, while $\boldsymbol{x}_s^{int}$ are of internal buses (cf. Fig. 2.4a). The sub-system can not use measurements that are functions of state variables belonging to other sub-systems. Hence, tie-line power flow and border bus power injections cannot be included in the sub-system level SE, but may be used at the coordination level.

At the coordination level, the state variables of the border buses (excluding all internal buses) are re-estimated,

$$\boldsymbol{x}^b = [\boldsymbol{x}_1^{b^T}, \boldsymbol{x}_2^{b^T}, ..., \boldsymbol{x}_S^{b^T}]^T \tag{2.30}$$

along with the phase-angles of sub-system reference buses, commonly referred to as u variables:

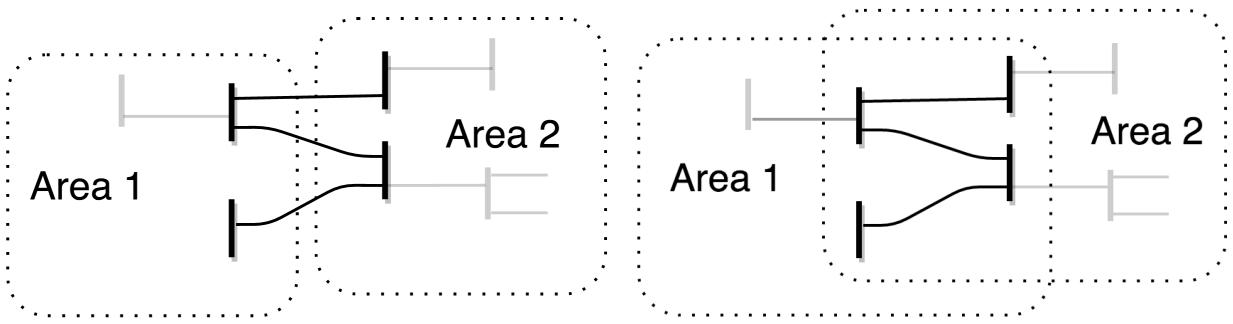
$$\boldsymbol{u} = [u_1, u_2, ..., u_s]^T = [\theta_{ref,2}, \theta_{ref,3}, ..., \theta_{ref,S}]^T \tag{2.31}$$

Together, $\boldsymbol{x}^b$ and $\boldsymbol{u}$ constitute the state vector $\boldsymbol{x}_c$ of the coordination level SE. $\boldsymbol{x}^b$ has already been estimated in the sub-system level, from which the result is denoted as $\bar{\boldsymbol{x}}^b$, which can be taken as pseudo-measurements. The previously not applicable tie-line power flows $\boldsymbol{z}^{t-l}$, can in the coordination level be expressed by state variables of both sides of the border. The border measurement vector $\boldsymbol{z}^b$ is thus simply $\boldsymbol{z}^b = \boldsymbol{z}^{t-l}$. Consequently, the SSE measurement model of the coordination level, $\boldsymbol{z}_c = \boldsymbol{h}_c(\boldsymbol{x}_c) + \boldsymbol{e}_c$, contains:

$$\boldsymbol{x}_c = \begin{bmatrix} \boldsymbol{x}^b \\ \boldsymbol{u} \end{bmatrix} \boldsymbol{z}_c = \begin{bmatrix} \boldsymbol{z}^b \\ \bar{\boldsymbol{x}}^b \end{bmatrix} \qquad \boldsymbol{h}_c(\boldsymbol{x}_c) = \begin{bmatrix} \boldsymbol{h}^b(\boldsymbol{x}^b, \boldsymbol{u}) \\ \boldsymbol{h}^{t-l}(\boldsymbol{x}^b, \boldsymbol{u}) \end{bmatrix} \tag{2.32}$$

2.5.3 Classification

In [32], a set of classification criteria for MASE methods was suggested:



(a) Non-overlapping areas.

(b) Overlapping areas.

Figure 2.4: Two area overlapping levels of MASE. The border buses are drawn in black, the internal buses in gray.

- Area overlapping level: The SE areas may have none or up to several layers of border buses in common (cf. Fig. 2.4). This choice governs which border measurements can be used in the sub-system SE, because their measurement functions must only contain sub-system state variables.

- Computing architecture: In hierarchical systems all the sub-system SE computers communicate only with a centralised computer, which does the coordination. In the decentralised approach, there is no centralised computer or coordinating entity, as all computers communicate only with computers of neighbouring areas.

- Coordination scheme: Coordination at SE level, i.e., after convergence of sub-system SE, or after each sub-system SE iteration. Iteration level coordination guarantees optimality upon convergence.

- Process synchronisation: In an iterative coordination, the centralised computer can either wait for all sub-system estimates, which is synchronous processing, or not, which is asynchronous processing.

- Measurement synchronisation: If the measurement collection from the SCADA system is done at different time instants in the different sub-systems, there will be a time-skew error in the global estimate.

- Solution methodology: The solution methods applied are commonly WLS based, but also a few others have been proposed.

2.5.4 Utilisation of Phasor Measurements

PMUs might be the triggering factor for industrial implementation of MASE, finally, because of their ability to *measure* phase angles in different areas. Earlier, the phase angles in different areas could only be coordinated using the sub-system estimates of border state variables, together with border measurements, through the $\boldsymbol{u}$ vector.

Consider first, for a system with no PMUs, that a pseudo-measurement of zero voltage phase angle is used for one bus. This would be equivalent to setting the bus as the reference bus, because this one angle pseudo-measurement would be a critical measurement. For MASE, having one PMU measurement in each sub-system could rid the need for a reference bus, since the estimated states would all be referenced to the global UTC. Hence, there would be no need for the re-referencing vector $\boldsymbol{u}$ [33].

However, two things must be noted:

- If area s has no PMU measurements available (temporarily or permanently), it would again need a reference bus, and hence also the u_s variable for re-referencing to UTC.

- With only one PMU in a sub-system, the phase measurement error of that PMU would directly affect all phase estimations of that area, because the voltage phasor angle would be a critical measurement in the sub-system SE [19]. Multiple PMUs, or alternatively, keeping a reference bus and using $\boldsymbol{u}$ even by one PMU, is thus beneficial.

In [11] and [34], PMU measurements were processed at the coordination level, which has shown to greatly improve the BD detection. Earlier, the tie-line power measurements would often be critical measurements. With PMUs, not necessarily placed on the border, a good estimate of phase angle differences is already provided. Combined with tie-line measurements, which also depend on phase angle differences, there is good redundancy for estimation improvement and BD detection. Thus, PMUs have a much bigger impact on accuracy of the two-level SE than on the conventional SSE [35].

2.5.5 Multilevel State Estimation

As more and more measurements are available for power system monitoring, it becomes reasonable to question the convenience of centrally processing all raw measurements of a TSO's control area. In this context a multilevel approach, where raw measurements are processed locally, has been proposed [36]. It is a generalisation of the two-level approach for multi-TSO SE, with further subdivision of a TSO area. It can, for example, entail substations performing local SEs, from which the results are used by the TSO's SE, which again provides the wide-area SE measurements.

Chapter 3

The Power Substation

The substation is a point of electrical connection between transmission lines, transformers, generating units and equipment for system monitoring and control [37]. They are the active centres of the physical transmission system, as most of the measurement and control actions are taken within the substations. In the context of SE, the substation is where the measurement data is acquired and sent to the control centre, where the central estimator resides.

Each incoming and outgoing circuit, which could be a transmission line, transformer, capacitor bank etc., terminates at what is called a substation bay [37]. From the bays, the circuits connect to one or more substation bus bars, although not directly, but rather through a series of switching devices such as circuit breakers (CBs) and disconnectors, through protection devices such as lightning arrestors, and through measurement devices.

This chapter gives an overview of some important aspects of the substation, and novelties of the digital substation in particular, relating to opportunities for SE.

3.1 Physical architecture

The substation is a point of connection, and it may perform various functions. Switching substations connect two or more transmission lines on the same voltage level together, with bus bars and high-voltage switches, and possibly including other devices, such as capacitor banks, power flow control devices, measurement devices, etc. Often the connecting lines have different voltage levels, and then one or several power transformers are needed.

The following discussion is aimed at substations in the transmission system, because these are relevant to transmission level SE, which includes transmission substations and (primary) distribution substations (cf. Fig. 3.1). Transmission substation generally denotes substations in which all connecting lines are transmission lines, either between two different voltage levels (in which the substation consists of two buses) or at a single voltage level. Distribution substations are the interface between transmission and distribution systems, and thus have a HV side, usually connecting to several transmission lines, and a Medium Voltage (MV) side, connecting to radial distribution feeders.

This section is meant to give an introductory understanding of the power substation.

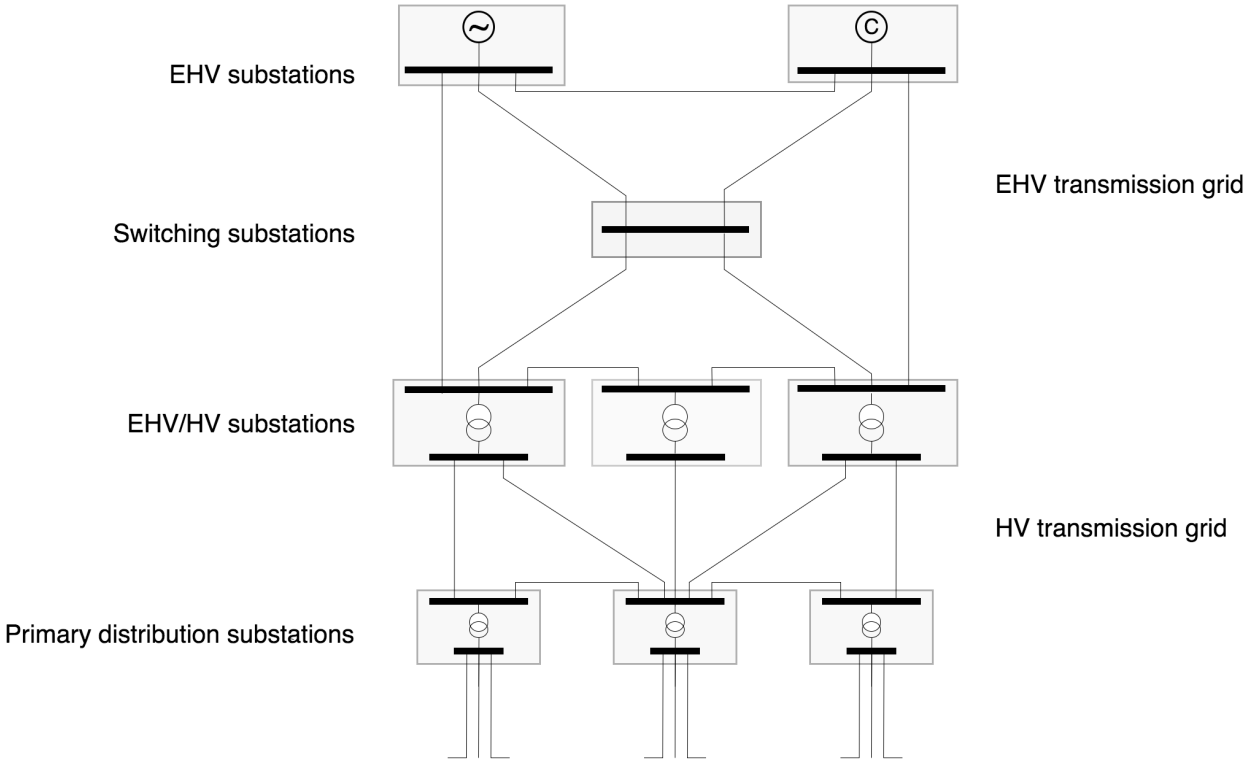


Figure 3.1: The basic structure of power systems, showing the transmission (EHV), subtransmission (HV) and distribution (MV) systems. The grey boxes denote the substations. As seen, the transformer splits the transformer substations in two buses.

3.1.1 Components

A substation usually consists of a control house and an outdoor switchyard. In the switchyard, the following components are found, connected to higher voltages (cf. Fig. 3.2):

- Bus bars
- Circuit breakers
- Disconnectors and grounding switches
- Power transformers
- measurement transformers.
- Other protection equipment (such as surge protection)

Bus bars are the points of connection between different bays. CBs are needed for switching, either for protection purposes (to break a fault current and isolate a fault), operation control purposes (to break operational currents and change system operation) or maintenance (to break operational currents to isolate equipment for maintenance). The disconnectors (disconnecting switches) are used for visual isolation between buses, but only after the circuit breakers are operated, because disconnectors cannot themselves break current (cf. [38]). This visual isolation is important when maintenance is done, to assure technical personnel that the parts are de-energised.

The CB is of critical importance for the reliability of the substation, because of its protection and isolation functions, but also because it has many moving parts that can fail [39]. Therefore, a CB typically has a disconnector on each side, so that the single breaker can be isolated for maintenance without de-energising the whole substation [39].

Other protection devices and their control are important components of a substation, but they are not relevant in the context of SE. Measurement transformers, however, are of great importance, because they provide the electrical measurements, and are therefore treated in depth in the next section.

Secondary equipment, not connected at HV, are also needed in the substation, for example sensors and meters, which receive the low voltage (LV) measurement signals from the measure-

Figure 3.2: An illustrative substation switching yard. Bus bars of the three phases are shown in front, and they are connected through a CB and a measurement transformer to the power transformer in the back.

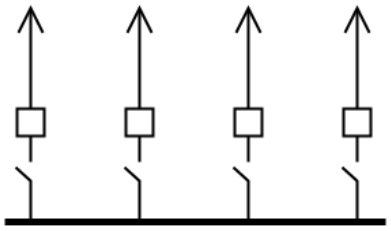
ment transformers; Intelligent Electronic Devices (IEDs), which are processing units executing a wide range of functions, using the sensor data; and also a communication network.

3.1.2 Substation Configurations

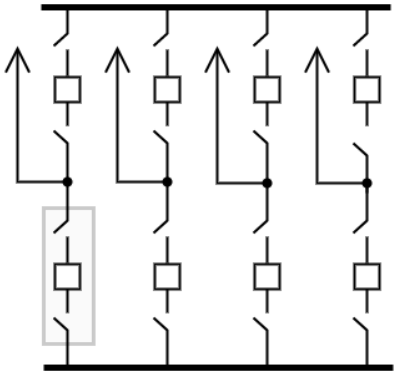
The configuration of breakers/switches and connections in the substation is chosen to achieve high reliability of service, flexibility in operation and to allow for equipment maintenance with minimal interruption of service [38], but must at the same time be cost-effective.

Some common configurations are shown in Fig. 3.3, and they are explained briefly in the following, so as to emphasise the reason for the use of more complex topologies than the single breaker.

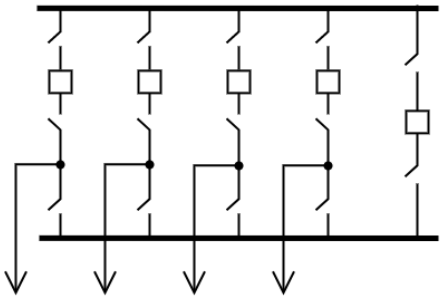
The simplest configuration is the single bus–single breaker arrangement of Fig. 3.3a. Although straight-forward and inexpensive, because only one breaker per line is used, it is limited



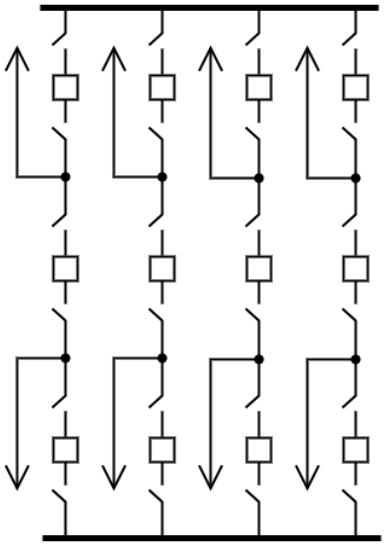
(a) Single breaker



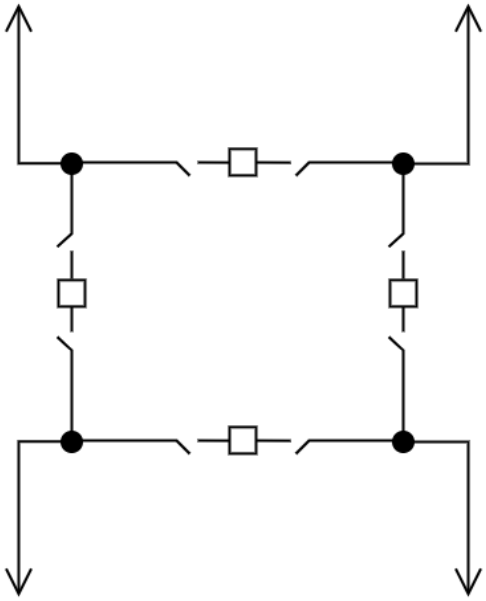
(b) Double breaker



(c) Transfer bus



(d) Breaker-and-a-half



(e) Ring bus

Figure 3.3: Substation configurations. The grey box in figure b shows what could be represented by a so-called logical CB.

in use [40]. When a breaker in this configuration is out for maintenance, or the breaker fails to close, the corresponding line cannot be served from the bus. Even worse, if the breaker fails to open during a fault, or the bus itself experiences a fault, the breakers of all the outgoing lines would have to be opened to clear the fault.

Ring bus configuration (cf. Fig. 3.3e) increases reliability and flexibility drastically, without increasing the number of breakers. The failure of a single breaker has no impact on the system, and a bus section fault only affects the corresponding line. Still, the ring bus configuration has disadvantages, such as complicated protection planning and little opportunities for expansion [41].

With the double breaker configuration (cf. Fig. 3.3b), on the other hand, the failure (or maintenance) of a single breaker or bus section has no impact on the system. In the best case, half the breakers and a bus can fail without disconnection of a single circuit. The double breaker scheme has excellent reliability and flexibility in operation. However, this is also the mostly costly configuration.

The breaker-and-a-half scheme (cf. Fig. 3.3e) takes the advantages of the double breaker configuration, but reducing some of the unnecessary redundancy. It is called so, because two lines share three breakers, reducing the total costs, but complicating the protection planning some [40]. The breaker-and-a-half configuration is still quite expensive, but is quite popular in practice [40].

3.2 Measurement, Communication and Processing

This section gives an introduction to measurement techniques used in power systems.

3.2.1 Conventional measurement set-up

The electrical circuits in the substation bays may be divided into two categories [37]:

- The primary circuit, which is the transmission line, power transformer, bus bars, HV side of VTs and CTs, etc.

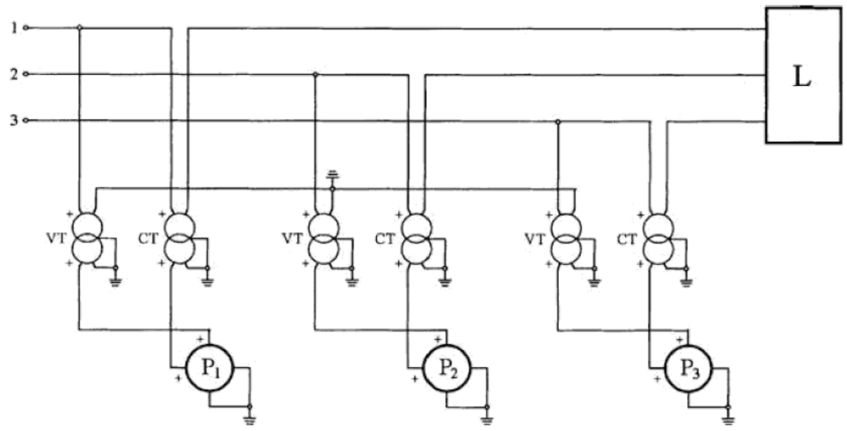


Figure 3.4: Configuration of a three-phase wattmeter, consisting of three current measurements and three voltage measurements [2].

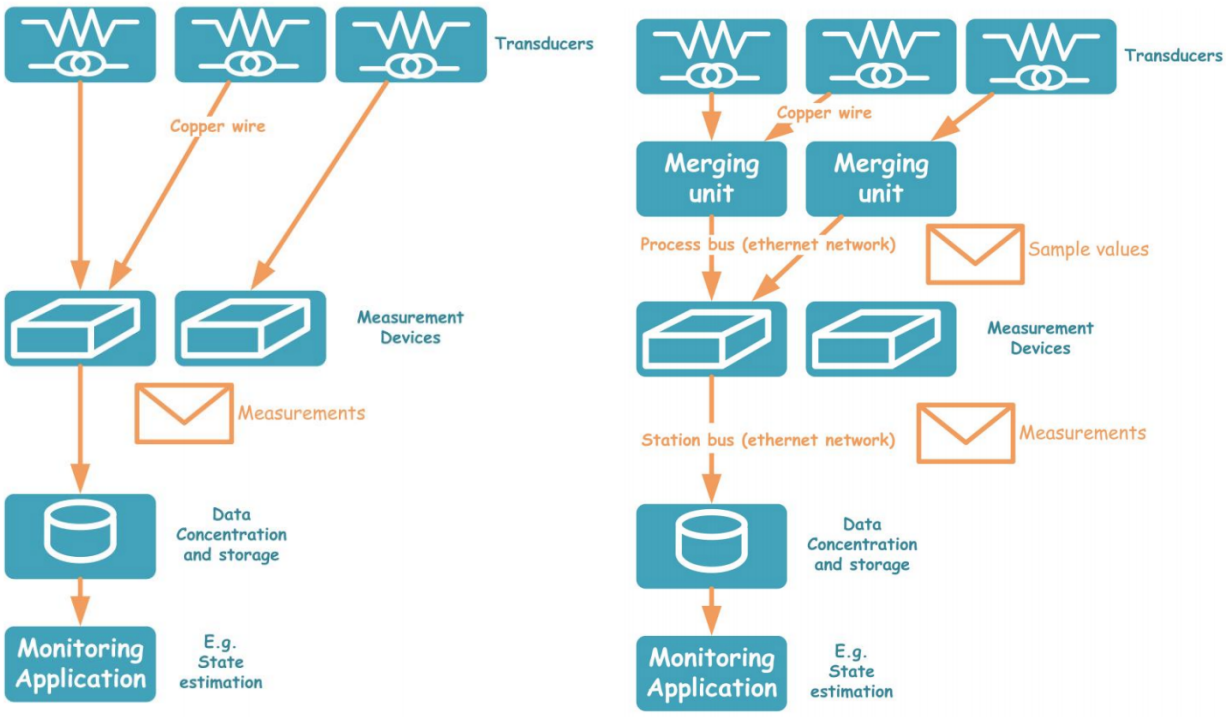
- The secondary circuit, which is the LV side of VTs and CTs (carrying the measurement signals), control circuits and protection circuits.

The conventional measurement scheme in the substations is demonstrated in Fig. 3.5a. Measurement devices, the sensors, are able to measure voltage and current. Since these measurement devices require lower voltages and currents than those in the HV grid, transducers, or measurement transformers, are needed to reduce the scale of voltages and currents. The ideal measurement transformer reproduces the exact same waveform, but with a scaled magnitude by a known factor, i.e., it works as an ideal transformer (cf. [38] for a thorough coverage of transducers).

Transducers are the interface between the HV components and the LV copper wire carrying the transformed, analogue measurement signal. The copper wire transmits the analogue signal from the switchyard into the control building, where the measurement devices use this analogue signal to produce the measurement values, such as rms current or voltage. Transducer can provide several output signals for different measurement devices, and every signal requires a separate copper cable. The measurement device can take as input several signals, as is demonstrated in the wattmeter in Fig. 3.4, which requires both current and voltage signals.

3.2.2 Modern measurement set-up

The modern set-up is shown in Fig. 3.5b. Again, measurement transformers are needed, and some copper wire, carrying the scaled, analogue signal. But, the copper wire only carries the analogue signal a small distance, before the signals are digitised by Analogue to Digital (A/D) converters called merging units, i.e., they are sampled. These samples are single values for voltage and current at the sampling instants, and are communicated into an Ethernet network (carried on fiber-optic cables) called the process bus (cf. Fig. 3.5b). Modern measurement transformers, so-called non-conventional instrument transformers, may even have integrated the two functions of scaling and sampling [2]. Then, only a fiber-optic cable carrying sample values exits the transformer. The devices actually producing the useful measurements, such as rms voltage or current, are also connected to the process bus, and are receiving the digital samples to process them further.



(a) Conventional measurement set-up.

(b) Modern measurement set-up.

Figure 3.5: Measurement paradigms [3].

3.2.3 Digital substation communication

Communication is an essential part of the modern measurement scheme. The digital substation and its communication system is defined in the standard IEC 61850, demonstrated in 3.6. In the following, the communication architecture is explained using this figure.

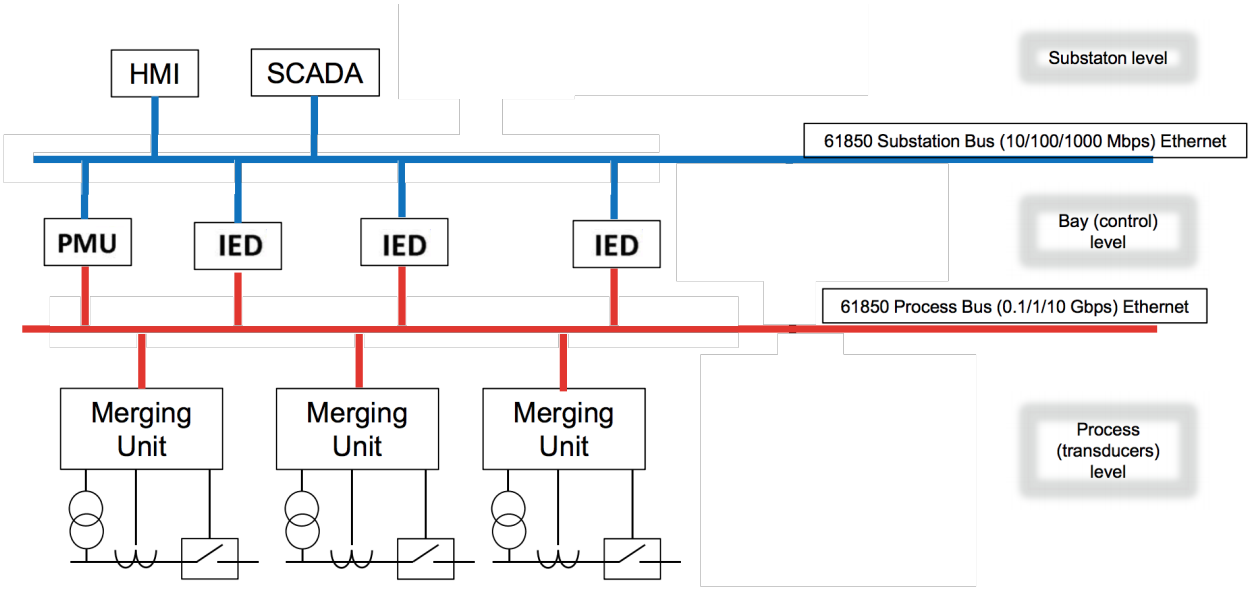


Figure 3.6: The communication structure of a IEC 61580 compliant digital substation [3].

In the process level, copper wires (coloured black in the figure) carry analogue signals, from CTs, CVs, CBs, etc., to the merging unit. Each signal needs its own copper wire going into the merging unit, since it is analogue. The output of the merging unit, on the other hand, is digital samples, i.e., small data packages.

Digital data can be communicated efficiently, so a very fast Ethernet connection called the process bus (coloured red in the figure) is used for the digital samples. The high speed is required to carry samples for time-critical applications, such as digital relays.

The process bus connects the switchyard with the station building. The recipients of the data samples are the family of devices called IEDs, which can be Digital Protective Relays, Digital Fault Recorders, Remote Terminal Units (RTUs), switch position monitors, revenue meters, etc. [42]. These are all located inside the station building.

The IEDs make up the bay level of the communication architecture. For monitoring purposes, the bay level is where the samples are processed to measurements values, such as rms

values of voltage or of active power. The measurement values are then communicated to the SCADA system through the station bus (coloured blue in the Fig. 3.6), a reasonably fast Ethernet network.

3.3 Local Measurement Redundancy

Some advantages of the digital substation that might be relevant for SE, based on [36] and [43]:

- Interoperability: equipment from different vendors can co-operate seamlessly, because communication is standardised.
- Integration: better sharing of information between different functions, because all the sample data are made available on the common process bus.
- Reliability: higher reliability on communication and reduced EMC issues following the use of fiber-optic process bus, leading to less BD from communication errors and less frequent measurement unavailability.

Multiple devices typically collect the same or similar data for different purposes. Currents may often be measured for the monitoring system and the protection system separately. Two different CTs are used, since the CTs for the two different applications are accurate in different ranges of magnitudes [4].

In Fig. 3.7, the typical measurements associated with a transformer is shown, where the redundancy is obvious. In addition to the measurements for monitoring and control, protection relays provide measurements, both electrical magnitudes and phase angles between them [4]. The transformer in Fig. 3.7 might typically have differential protection, providing current magnitude and phase angle measurements on both sides of the transformer. These measurements are available locally (on the substation communication bus) and synchronised.

If the substation data could be collected and utilised, the increased redundancy could enhance BD processing substantially [42].

Synchronisation of power system measurements, necessary to produce synchronised phasor measurements, is difficult because of the remote nature of the different measurement devices over a wide area (cf. discussion on PMUs in Section 2.4). Within a substation, however,

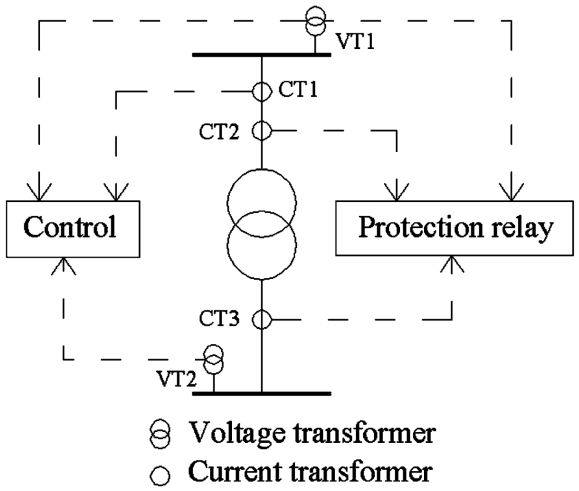


Figure 3.7: Typical measurements associated with a transformer [4].

a local synchronisation could be feasible. The phase measurements of protection relays could utilised, or alternatively, the time-stamped samples could be compared to a local reference by a new IED [36].

Challenges with substation data integration and accessing the local redundancy for BD processing, are summarised as follows:

- There is a great variety of data types from different devices [39]. Some may provide rms values, other peak values and others again phasors. Also, both pre-processed data, such as rms, and simple samples are available.
- Frequency and timing of data arrival can be different.
- Some measurements may be correlated, complicating BD processing and SE. For example, several measurement may be based on samples from a single VT, as is illustrated in Fig. 3.7.

Studying the substation in node-breaker detail, the full amount of measurements actually made in a substation becomes evident: many individual nodes with voltage measurements and many substation-internal branches with current measurements, as demonstrated in Fig. 3.8. The complete redundancy of measurements should be possible to allocate to this model.

In the next chapter, the role of node-breaker model in SE is discussed.

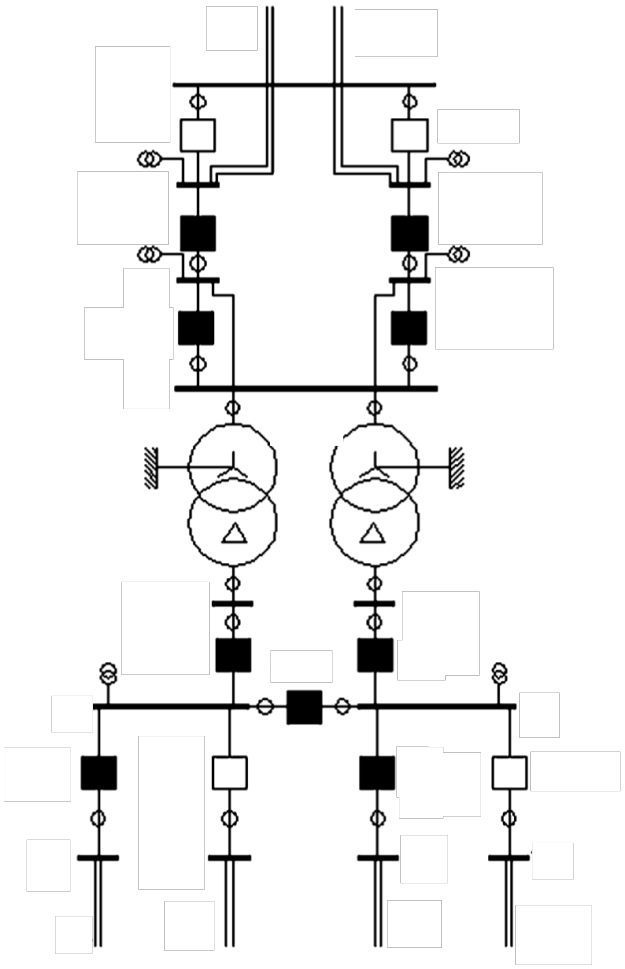


Figure 3.8: Node-breaker model of a 132/66 kV transformer substation, showing individual measurements [4].

Chapter 4

State Estimation using Substation Details

Most SE algorithms today use a bus-branch representation of the network, which is a model constructed by the topology processor. The detailed network model, including substation details, is collapsed into only a few electrical points, namely the *buses* (cf. Fig. 4.1). After this processing, the substation consists of one or more buses, depending on transformer locations and CB statuses.

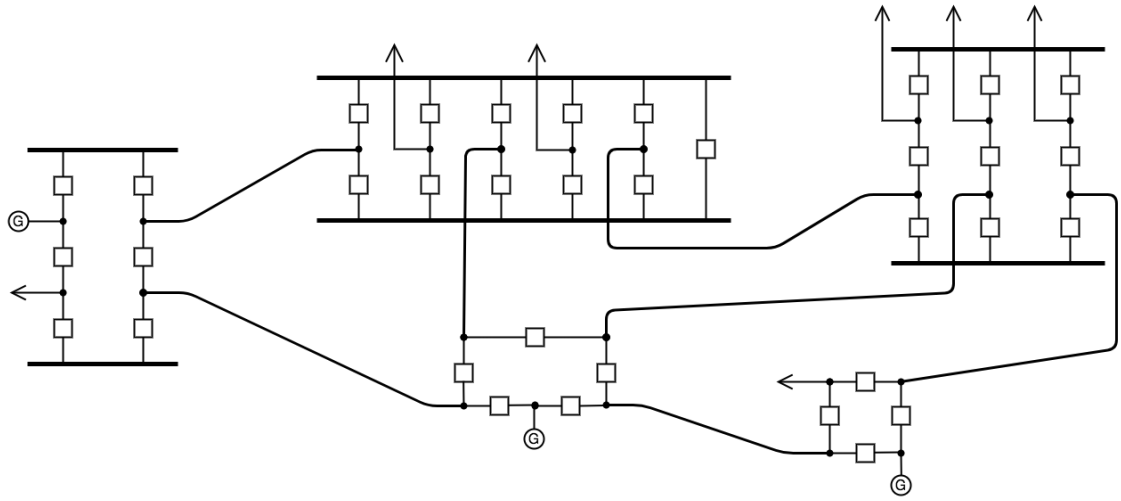
A review of topology processing and topology errors is given in this Chapter. The Generalised State Estimation (GSE), which can be applied directly on node-breaker network models, is also reviewed in this chapter.

The use of GSE may give rise to new and more reliable methods for topology verification. Also, the idea of applying GSE to substations locally will be discussed and a new Substation-Level State Estimation (SLSE) is proposed.

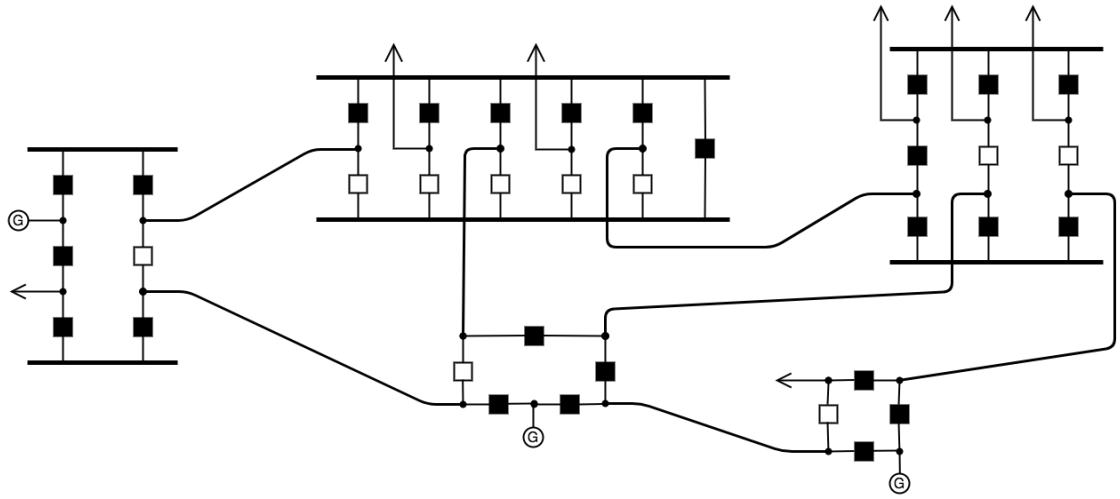
4.1 Topology processing and topology errors

The topology processor uses the static database and the real-time database to construct a network model at the bus-branch level, which is the model that is most usually used by SE and, subsequently, by other functions of the EMS. Since every other EMS function relies on the network model, a correct and efficient topology processor is needed.

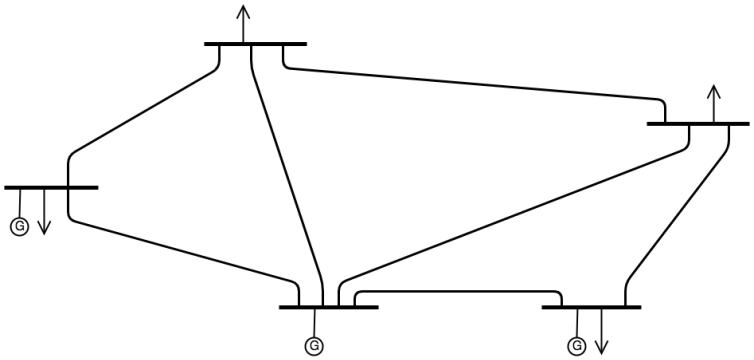
Additionally, for SE, the topology processor has to allocate the measurements to the buses and branches.



(a) Node-breaker model.



(b) Node-breaker including CB statuses.



(c) Bus-branch model.

Figure 4.1: The topology processor collapses the complete node-breaker model to the few buses of the bus-branch model. CBs are denoted as squares, and closed CBs are filled with black [5].

This section discusses the role of the topology processor as well as the problem of topology errors.

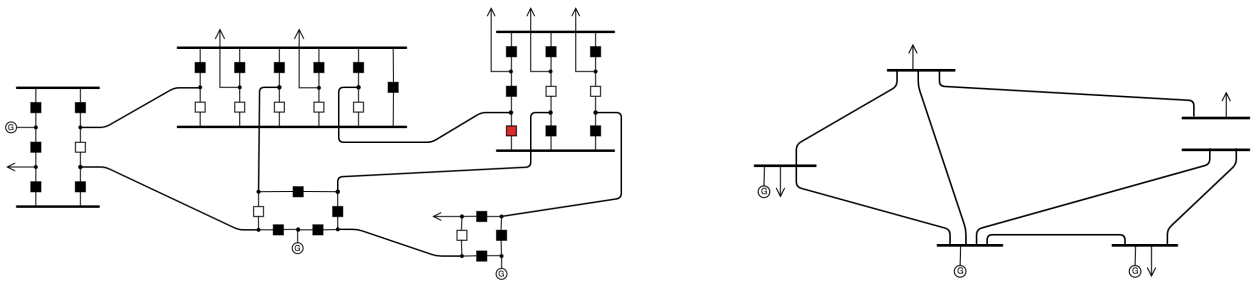
4.1.1 The Conventional Topology Processor

The topology processor has two information sources:

- the static database, containing a list of all the components in the system, such as branches, other connections, CBs, shunt capacitors, generators, transformers, etc.
- the real-time database, containing analogue measurement data, status data (CB statuses) and transformer tap settings, usually telemetered from the devices periodically.

The static information provides all the possible connections in the network, while the real-time data determines how these components are actually connected in the given instant. In Fig. 4.1, the static database provides the complete node-breaker network (Fig. 4.1a), while the real-time database provides the CB statuses (Fig. 4.1b) necessary to construct the bus-branch model (Fig. 4.1c).

Although topology processing might appear an easy task if a closed CB is considered, indicating a disconnection of the corresponding circuit, it is not so straight-forward when considering the spectre of more complex substation topologies (cf. Section 3.1.2), where the opening of a CB can result in either a disconnected circuit, splitting of a bus or have no effect at all, as is demonstrated in Fig. 4.2.



(a) Node-breaker model with erroneous CB status (in red).

(b) Erroneous bus-branch model.

Figure 4.2: A CB status is erroneously reported as open (marked in red), resulting in a topology error, namely a bus split error.

Conventional topology processing could typically involve:

- simple, rule-based telemetry (status and analogue data) checking.
- construction of the bus-branch model, based on the node-branch model and the telemetered CB statuses.
- allocation of analogue measurements to the bus-branch model.

The simple telemetry check, which could include KCL at the nodes, identifying clearly wrong measurements, such as zero power flow on disconnected lines, is used in practice to root out obvious errors [44] [4] [45] [27].

Already the allocation of analogue measurements assumes a correct bus-branch model. It takes all available measurements of the physical node-breaker model and allocates these to positions of the bus-branch model.

If the active power injections of all the nodes comprising a bus (i.e., nodes which are connected through CBs assumed to be closed) are measured, these can be summed to constitute an active power injection measurement of the bus in the bus-branch model. However, if one of the node injection measurements is not available, the bus injection measurement cannot be determined. Correspondingly, power flow measurements of zero-impedance branches may or may not be allocated to branches of the bus-branch model, depending on the node-breaker network topology and the assumed breaker statuses. Also, in substation, voltage measurements usually exists both for bus bars and the substation bays, i.e., for all the nodes of a substation [12]. For communication efficiency, these voltage measurements are merged, through a weighted average, to a single bus-branch voltage measurement [12].

As an example, the substation depicted in Fig. 4.3 is considered. All injections, i.e., of nodes 3, 5 and 7, must be measured for the topology processor to be able to build an injection measurement for the bus-branch model. If the power flow through the CB between node 1 and 5 (CB 1-5) is measured, this can be used when the power injection in node 5 is not available. The same possibility does not exist for the injection at node 3, except if both CB 1-3 and CB 3-4 power flow measurements are available. The same observation can be made for the branch power flows: The branch power flow (outgoing on a transmission line) from node 6 could either be measured at the beginning of the branch or in CB 2-6.

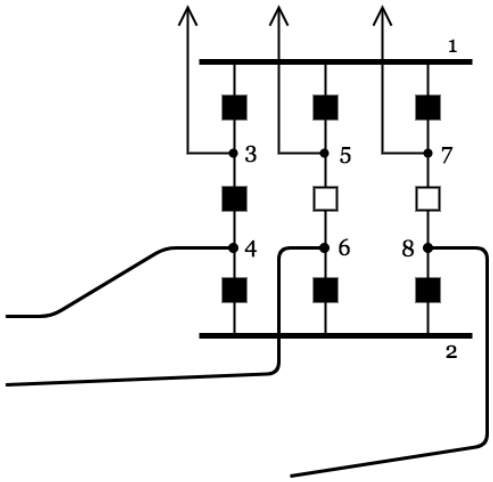


Figure 4.3: Zoom-in of the node-breaker model of a single substation in Fig. 4.1.

So, the topology processor has to allocate the measurements available to branches and buses, even if they are physically originating from an internal connection (between nodes) of a substation. If there are errors in the CB statuses, it does not only alter the SE model (bus-branch model), but it may lead the topology processor allocating wrong measurements to the SE.

4.1.2 Topology Error Processing

If one or several CB statuses are erroneous, the topology processor will produce an incorrect network model, leading to a biased estimated state. Since the CB statuses are not included in the estimation model, the traditional methods "*basically attempt to indirectly infer the presence of topology errors from their impact on analogue measurement's estimation residuals*" [46].

The topology error detection and identification happens only *after* the SE has been executed, when potential errors in the topology processing may already have corrupted the BD processing, biased the estimation adversely and even have lead to convergence issues.

It is not always evident that increased residuals result from a topology error, because it may look like multiple, conforming BD (cf. Section 2.3). Detection may therefore be challenging, and exact identification maybe even impossible, especially since both the topology and possibly the measurements (allocated by the topology processor) may be changed.

Different types of topology errors give different responses in the SE. Topology errors can be classified into the following types [16]:

- Inclusion error: the inclusion in the SE of a branch or other component, when it is in fact disconnected.
- Exclusion error: the falsely exclusion in the SE of a branch or other component, when it is in fact connected.
- Split error: modelling a substation section as several buses in the SE, when it is in fact connected by closed CBs/zero-impedance branches.
- Merging error: modelling a substation section as a single bus in the SE, when it is in fact split into several buses by open CBs.

If topology errors are not corrected, the real-time model used by other EMS analysis functions may be erroneous both in topology and in state, possibly leading the operator to take unoptimal control actions.

Thus topology errors in conventional SE usually biases the estimation and the residual, i.e., the expected value of the residual will no longer be zero (topology errors that do not bias the residual, are not detectable by conventional SE. High values of normalised residuals can therefore indicate the presence of topology errors [47]. Also, if equality-constrained SE methods are applied, the Lagrange multipliers of zero-injections could be applied similarly to residuals: A significant Lagrange multiplier could indicate a topology error close to the equality-constrained zero-injection bus [48]. Unfortunately, high residuals and Lagrange multipliers can also result from BD.

One way to identify substations that may contain a topology error, would be to identify the buses which are incident to the most suspected measurements (with normalised residuals over the threshold) [49], but a secondary method is needed to specify the topology error. For example, the SE of the suspected area could be repeated for all possible combinations of switch statuses, choosing the one with the smallest residuals or best performance index [50]. However, these kinds of methods are not efficient enough for real-time application [51].

Another method which uses the residuals for topology error identification is hypothesis testing, where the residuals resulting from different specific topology errors are estimated and compared with the actual residuals [47].

Values of the state and power flows in earlier time-instants, or any other information available in the EMS, could be utilised for topology error processing. A rule-based method, meant to imitate the investigative process of an engineer, is suggested and implemented in [51] and [52]. In this method, every piece of relevant information is inspected to determine if it is compatible or incompatible for the given CB statuses [51]. Certain patterns in abrupt changes may indicate a switching operation in a certain area. Also, techniques based on artificial intelligence have been suggested for the same knowledge or experience based topology error processor [53].

A practical topology error processor should be able to detect all topology errors, and at the same time be fast, noise-resistant, robust against BD and easily implemented in current SE software [51], and as far as literature can show no single method is widely accepted to satisfy all the criteria.

The commercial, real-life state estimators usually depend solely on pre-filtering plausibility checks for identification of topology errors, since no intrinsic topology error processing exist in the SE algorithm [12], even though, for the German TSOs, correct and timely recording of switching operation is reported to be a main concern [12].

4.2 Generalised State Estimation

The idea of including circuit breaker signals in SE method was first suggested in [45] for validation of measurement data received from the substations, and it has been further developed to a complete GSE methodology [16] [54]. In GSE, the status measurements from CBs are taken as stochastic measurements (and even parameters of lines, transformers, etc.); the estimation is generalised from that of the SSE, which only considers uncertainty in *analogue* measurements.

In conventional SSE the status measurements are initially assumed correct and taken as input for the topology processor. They are not explicitly modelled in the SE algorithm, and the topology error detection happens *after* filtering of the analogue measurements. A topology error might therefore typically result in healthy measurements being suspected as multiple, interacting BD [12].

Although having received a lot of research attention, the post-filtering detection and correction of topology errors is challenging, as discussed in the preceding section, because of the

chaotic influence of topology errors on the state estimate. In GSE the status measurements are included in the measurement vector, and therefore the same proven techniques applied to bad analogue data could possibly be applied to bad status data. Even if other methods are used, with GSE the breakers are included in the problem formulation (possibly implicitly), providing scope for much more direct topology error identification, down to the specific erroneous breaker (depending on the model applied).

GSE is also a generalisation in the sense that it can be applied for not only bus-branch models, but also for physical level, node-breaker models.

4.2.1 Breaker models

A logical circuit breaker, from here on denoted only as a circuit breaker (CB), is a combination of a physical breaker and switches, as shown in Fig. 3.3b. In the closed position, the CB should be modelled as a zero-impedance branch, and in the open position as an infinite impedance branch.

The flow through a closed breaker on a zero-impedance branch is, as opposed to the traditional branches modelled in LF and SE, not controlled by the state variables of the two connecting buses. With the traditional branches, the impedance (as defined by Ohm's law) is the coupling link between terminal voltages and power flow [1]. In a zero-impedance branch the flow of active (or reactive) power is only decided by Kirchhoff's current law, as will be evident from the following discussion. Since the state vector, which per definition should include the necessary and strictly needed quantities to calculate all power flows, cannot be used to calculate power flows of zero-impedance branches, these power flows must be added in an augmented state vector. So, for the zero-impedance branch between k and l, the active power flow, p_{kl}, and the reactive power flow, q_{kl} is added to the state vector of (2.1):

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{v} \\ \boldsymbol{p} \\ \boldsymbol{q} \end{bmatrix} \tag{4.1}$$

Both $\boldsymbol{p}$ and $\boldsymbol{q}$ vectors are of length N_{CB}, which is the number of CBs in the system.

Thus, the inclusion of breakers requires a bigger state vector, which in itself would decrease redundancy. Fortunately, more constraints are being added as breakers are modelled.

For a closed breaker between nodes k and l, the breaker terminals are short-circuited, hence the voltage drop between the terminal nodes should be zero. More specifically, the voltage magnitude and phase angle differences over the branch is zero:

$$v_k - v_l = 0 \tag{4.2}$$

$$\theta_k - \theta_l = 0 \tag{4.3}$$

For an open breaker, the terminals are connected through an infinite impedance, hence guaranteeing that the active and reactive power flows through the breaker are zero:

$$o_{kl} = 0 \tag{4.4}$$

$$q_{kl} = 0 \tag{4.5}$$

These new constraints can be added as pseudo-measurements with a high weightage. The SE result will then normally converge to close to zero power flow for open breakers and zero voltage difference for closed breakers. However, in the case of a bad status signal and hence an incorrect pseudo-measurement, the SE results may differ substantially from the breaker status assumption, revealing the erroneous breaker status (to be discussed further in the following subsections).

Some times the breaker status is considered unknown, because its status signal is suspect or no signal has been received. In this case the constraints (4.2)-(4.5) could be omitted. This, however, often results in the SE converging to an intermediate CB status [55]. Instead, the constraints could be replaced by more general constraints [55]:

$$p_{kl}(\theta_k - \theta_l) = 0 \tag{4.6}$$

$$q_{kl}(v_k - v_l) = 0 \tag{4.7}$$

These constraints will enforce either a zero power flow through the CB or zero voltage difference

over the CB, depending on the other measurement values, because one of the two must be zero for the equation to be zero. Hence, the estimation is forced to converge to either one of the two excluding statuses.

Usually, only CBs on zero-impedance branches are modelled using this formulation (cf. [16] for formulations for CBs on conventional branches). Wherever a breaker is located on a conventional (non-zero impedance) branch, a virtual node must therefore be placed between the breaker and the branch, and this node will naturally be a zero-injection node. GSE problems typically have a lot of zero-injection equality constraints.

It should be noted that the introduction of the new state variables also introduces new measurement equations. For example, if there exist substation-internal measurements of power flow (CB power flow), these will naturally have the same formulation as (4.4) and (4.5): the left side is a CB power flow state variable, according to (4.1), while the right-side value is simply equal to the corresponding measurement. Substation-internal power injection measurements thus equal sums of the connected CBs' state variables.

4.2.2 Solution methods for equality constraints

The originally proposed solution method for GSE was the conventional WLS, where equality constraints were treated as virtual measurements [54], [16]. In that case, the solution procedure is the same as that of the conventional SSE, only with augmented state and measurement vectors.

Virtual measurements in conventional SE are applied to the modelling of zero-injection buses, as well as external network modelling. In GSE, the number of zero-injection buses is much higher, since a lot more nodes are added, many of which are connected only to breakers and lines. Thus, for each breaker/zero-impedance branch, two breaker constraints are added, as described in the previous section.

Thus, GSE has a significant number of equality constraints. For accuracy, it is therefore important that these constraints are enforced quite strictly. There are two main approaches strictly enforcing constraints in SE, namely numerically robust estimation methods and constrained optimisation methods.

4.2.2.1 Numerically robust orthogonal factorisation method

Numerically robust methods do not so easily suffer from ill-conditioning, because they avoid using the very ill-conditioned gain matrix of the normal equation, (2.7). They can therefore be applied to GSE without numerical issues.

The orthogonal factorisation method was suggested in [56], and a review on the development of it is given in [27]. The equations of the method are given in the following.

First, the Jacobian is pre-multiplied by the square of the weighting,

$$\tilde{\boldsymbol{H}}(\boldsymbol{x}) = \boldsymbol{W}^{1/2}\boldsymbol{H}(\boldsymbol{x}) \tag{4.8}$$

To avoid squaring in the gain matrix, $\tilde{\boldsymbol{H}}'\tilde{\boldsymbol{H}}$, which increases the ill-conditioning, the Jacobian is decomposed into an orthogonal matrix, $\boldsymbol{Q}_1$ and an upper triangular matrix, $\boldsymbol{U}$ (Q_2 is not used):

$$\tilde{\boldsymbol{H}} = \boldsymbol{Q}\boldsymbol{R} = \begin{bmatrix}\boldsymbol{Q}_1 & \boldsymbol{Q}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{U}\\ \boldsymbol{0}\end{bmatrix} = \boldsymbol{Q}_1\boldsymbol{U} \tag{4.9}$$

The normal equation (2.7) can be rewritten, by inserting (4.8) and (4.9), as

$$\boldsymbol{U}'\boldsymbol{Q}_1\boldsymbol{Q}_1\boldsymbol{U}\Delta\boldsymbol{x} = \boldsymbol{U}'\boldsymbol{Q}_1\tilde{\boldsymbol{r}}(\boldsymbol{x}) \tag{4.10}$$

for the linear case, $\tilde{\boldsymbol{r}}(\boldsymbol{x})$ being the pre-multiplied measurement vector. $\boldsymbol{Q}_1$ is orthogonal, so $\boldsymbol{Q}_1' = \boldsymbol{Q}_1^{-1}$, which eliminates $\boldsymbol{Q}_1$ from the left side of the equation. Also, $\boldsymbol{U}'$ can be eliminated by multiplying both sides by its inverse. This results in:

$$\boldsymbol{U}\Delta\boldsymbol{x} = \boldsymbol{Q}_1\tilde{\boldsymbol{r}}(\boldsymbol{x}) \tag{4.11}$$

This equation can, for each iteration of the SE, be solved in two simple stages:

$$\boldsymbol{y}_1 = \boldsymbol{Q}_1\tilde{\boldsymbol{r}}(\boldsymbol{x}) \tag{4.12}$$

$$\boldsymbol{U}\Delta\hat{\boldsymbol{x}} = \boldsymbol{y}_1 \tag{4.13}$$

4.2.2.2 Constrained optimisation method

The equality constraints can also be handled explicitly as constraints, using constrained optimisation. Transferring all equality constraints from the measurement vector $\boldsymbol{z}$ to the equality vector $\boldsymbol{c}$, the problem can be formulated as

$$\min J(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{r}'\boldsymbol{W}\boldsymbol{r} \tag{4.14}$$

$$\text{subject to } \boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{0} \tag{4.15}$$

The basic method for solving this is using the Lagrangian function and applying the Karush-Kuhn-Tucker conditions to it, and linearising using Taylor expansion, resulting in [27]

$$\begin{bmatrix} \boldsymbol{H}^T\boldsymbol{W}\boldsymbol{H} & \boldsymbol{C}^T \\ \boldsymbol{C} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{x} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}^T\boldsymbol{W}\Delta\boldsymbol{z}^k \\ -\boldsymbol{c}(\boldsymbol{x}^k) \end{bmatrix} \tag{4.16}$$

where $\boldsymbol{C}$ is the Jacobian matrix of the equality constraints and $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. Equation (4.16) can then be solved iteratively to achieve the optimal state solution.

It can be shown that this constrained estimation is equivalent to applying infinitely high weights to virtual measurements [1], but without virtual measurements to cause ill-conditioning.

The Lagrange multipliers can be seen as expressing the "cost" of enforcing constraints [1]. A lower multiplier value therefore indicates an equality constraint conforming well with the other measurements, while a high multiplier value indicates that the constraint does not fit the rest of the measurement set very well. Lagrange multipliers may therefore be used to identify errors in the virtual measurements, indicating a topology error. In this way, Lagrange multipliers can be used similarly as the normalised residuals [1] [48]. In fact, the normalised residuals for virtual measurements of the unconstrained optimisation, if weightage goes to infinity, equals the Lagrange multipliers obtained using constrained optimisation [48]. Therefore, the normalised residuals using finite weighting can be treated as approximations of the Lagrange multipliers, and conversely, established methods for analogue BD analysis using residuals can be applied to Lagrange multipliers.

Thus, the equality-constrained GSE allows the exact solution to be found, assuming the breaker status signals to be correct, while allowing for easy topology error correction using La-

grange multipliers. The Lagrange multipliers can then be used interchangeably with the normalised residuals.

4.2.3 New topology error processing

Topology errors are inherently much easier processed with GSE than with conventional SE. Since each residual corresponds to a single CB (or rather, to a single logical CB), an erroneous CB status should intuitively show up in the SE as a high normalised residual, as in the case of bad analogue data.

Still, an incorrect CB status could lead to high values of normalised residuals in the vicinity of that CB. One might assume that the effect is similar to the effect of measurement BD in conventional SE, where a single BD will present itself as the highest normalised residual (cf. Section 2.3), which means that the normalised residual test can be applied to bad status data also. But some questions remain unanswered, such as what impact the pairing of CB constraints has on BD/bad status identification, or how the chi-squared test performs. No literature is found that investigates this in detail. However, some discussion can be found in [46] and [57], where hypothesis testing is found effective for GSE topology error processing; it is not made clear, however, how this technique would perform in the presence of BD.

If a CB status is identified as bad, both its constraints should be modified. If the constraints are removed, the estimate may not converge to a fully open or fully closed state [55], so it may have a non-negligible power flow *and* a voltage drop. Instead of removing the constraint, it could either be transformed to the opposite state or it could be replaced by the unknown breaker status constraint.

Also, a potential problem in the topology error processing is that some CBs may satisfy (approximately) both constraints for open and closed CBs, for some particular operational states. A closed CB may accidentally have zero (or close to zero) active and reactive power flow, and two nodes connected by an open CB could by chance end up with the same voltage. In these specific situations, the status of the breakers are unverifiable by GSE [58]. The state estimate is biased by this, but the topology is used for other EMS functions that may be affected.

4.3 Approaches for State Estimation with Substation Models

As with MASE methods, the computing architecture can be recognised as an important characteristic to classify these methods (cf. Section 2.5.3). The three architectures considered here are centralised, two-level (which is partly decentralised), and completely decentralised.

As far as the literature can show, GSE has yet to be implemented in any real-life power system [59]. Only simple rules for checking telemetry have been used in practice [45]. But, a few methods of applying GSE have been suggested.

This section critically reviews earlier approaches for including node-breaker models in SE and proposes a new SLSE method.

4.3.1 Node-Breaker Model transformed to Conventional Model

One straight-forward approach, would be to approximately model closed switches as very low reactance branches and open switches as very high reactance branches, although they in fact have zero and infinite impedances, respectively. While the conventional topology processor uses the reported CB statuses to build the bus-branch system, it would in this case only change the reactance of the zero-impedance lines according to the CB statuses. A negligibly low reactance on a theoretically zero-impedance branch should not alter the LF or SE result in steady state analysis.

This topology processing would result in a bus-branch model, but in which every node is a bus of its own. The conventional static SE algorithm described in Section 2.2 could then be applied, and all node-breaker level measurements could be applied directly without assumptions on the topology. Although no literature has been found testing this approach, having very different impedance values causes serious numerical issues [1], which makes it impractical for real applications. Also, the conventional SE has no mechanism for evaluating the branch parameters, which makes status data error processing close to impossible.

4.3.2 Full Network Generalised State Estimation

The straight-forward application of GSE is possible on a complete network modelled at node-breaker level, such as the system in Fig. 4.1b. However, including the node-breaker model of

substations increases the number of state variables and measurements drastically.

The direct application of GSE would increase the problem size and the resulting computation time dramatically, so that it may not be applicable to real-time SE [5]. In an attempt to make GSE applicable to full-size systems, implicitly constrained substation models were developed in [60], which were shown to efficiently reduce the problem size without losing the capability for topology error processing. However, since the modelling of each single CB is replaced by an implicit model, the topology processing becomes more complicated, as does the construction of the SE model.

4.3.3 Two-Stage Generalised State Estimation

Another centralised approach, which has found wider acceptance, is using the mixed network modelling, since detailed modelling of all substations is computationally too expensive. CB status measurements are, after all, mostly correct. In this approach, first imagined in [45], only models of some suspected substations would be expanded to the node-breaker details. A viable implementation is the two-stage method [1], [12], [13], [49], [54], [57], [61]:

Stage 1 A LAV based SE (cf. Section 2.2.4) is executed for the bus-branch model of the system. If any anomaly (BD or topology error) is detected, in this case any residuals are over a limit, stage 2 will be executed next. The substation/substations to which most of the high residuals belong, are suspected of topology errors. The reason for using a LAV estimator is that it is a so-called robust method, i.e., bad measurements are automatically rejected, and so do not affect the estimate and residuals adversely, and thus the residuals can be used to identify suspected substations (cf. Section 2.2.4).

Stage 2 The models of the suspected substations are expanded to include nodes and breakers, so that the system becomes a mixture of bus-branch and node-breaker systems. A new LAV SE is executed, which must be a GSE to be able to handle the node-breaker modelled areas. It is assumed that all the topology errors (only located in node-breaker modelled substations) are automatically corrected in the SE process, since the LAV estimator rejects bad measurements, including bad CB pseudo-measurements. Therefore, the resulting BD can be identified independently of the previous topology errors.

The biggest advantage of this application is that the problem size is limited, since only a few substations are expanded to node-breaker detail. Alternatively, the problem size could be reduced further by executing the second-stage GSE only for the area surrounding the suspected substations, which is called "zooming" into a "bad data pocket" [27].

A disadvantage of this application is that the first-stage SE may display convergence issues in the presence of topology errors, such as the conventional SE, since the conventional topology processing is used. The LAV based SE is also computationally demanding, possibly reducing the real-time applications [15].

4.3.4 Substation Level State Estimation

One strategy for including substation details in SE is using a two-level method, because it limits the detailed modelling to a local area, allowing a TSO-level SE to use the bus-branch model:

Level 1 Substation level state estimation (SLSE), where the raw-measurements are used for a local SE, using a node-breaker mode in each substation.

Level 2 TSO-level state estimation, where the local level SE results are coordinated. In this context, the TSO-level SE may be executed in the control room of the TSO, utilising a bus-branch model.

Although similar to the two-level method in MASE described in Section 2.5.2, which uses the TSO as the local level, the two-level method in this section uses the TSO as the TSO-level, coordinating local solutions. The two-level methods discussed here are non-iterative (cf. Section 2.5.2), i.e., there is no iteration between local and TSO-level, so the local level can also be seen as a pre-processor for the TSO-level SE.

The idea of local pre-processing of measurement data in the substations is not a new one. But, it may now be practical for implementation, because of the increase in measurement redundancy and the internal communication and computational capabilities of the digital substation, as discussed in the preceding chapter.

The SLSE is envisioned to exploit the full potential in modern transmission system substations. In addition to the conventional SCADA measurements, phasor measurements and protection IEDs could be applied to a local SE, which is made possible by the use of node-breaker

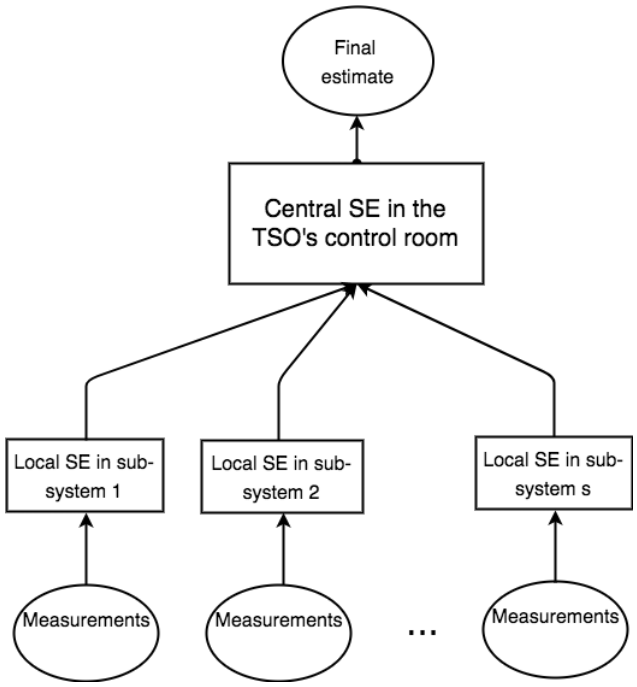


Figure 4.4: Two-level concept for node-breaker substation level SE. Each local SE is executed for one substation (with various definitions for the substation).

network model and local utilisation (cf. Section 3.3). Also, some measurements are not usable by the bus-branch model of the conventional SE (as demonstrated in 4.1.1), which increases the redundancy of the node-breaker model further.

The application of the detailed, physical-level network with a lot of extra measurements is not feasible for the entire network, because of the problem size; the problem size of the SLSE, however, is naturally limited. Also, the communication bandwidth would limit the possibilities of sending the extended measurement set to the TSO estimator, and it would lead to greater delays.

The advantage of exploiting all measurements is that instead of raw measurements, verified pseudo-measurements are sent to the EMS. Even more importantly, topology errors and BD could be detected, identified and removed locally, utilising the redundancy provided by the extra measurements [12]. The time-demanding BD processing of the conventional SE has to rerun the complete estimation for every identified and removed BD. If performing satisfactorily, the local correction of these errors would entail only rerunning the small substation problem.

To summarise, some advantages of using a SLSE are:

- The small size of the system allows for use of the node-breaker model, possibly allowing for much more accurate BD and topology error processing.
- Increasing the number of measurements, because of the use of node-breaker model and because it is not limited by communication capabilities (i.e., communication delay from substation to control centre), since the measurement set is used locally. This increased redundancy has the potential to better BD and topology error processing, for better estimate reliability, and for increased accuracy of the filtering.

The two points above make local BD and topology error processing feasible. In such a case, rerunning the complete system after removal of BD can be avoided, because only the much smaller local problem must be rerun.

4.3.4.1 A classification criterion

The node-branch model of a single substation generally consists of linear connections (the zero-impedance branches with CBs) and non-linear conductors and transformers (cf. Section 3.1.2). If the substation is separated at transformer terminals, every voltage level may be considered as a separate substation and the transformers can be considered as external branches (which is done in LF analysis). Using this definition for the substation, i.e., that each voltage level is a separate substation, the system consists of linear substations connected by non-linear branches (including transformers).

Thus, the local substation level SE can be planned to be linear or non-linear depending on the sub-system choice. Similarly, the TSO-level SE can be planned to be linear or non-linear depending on the choice of output of the local level, or, equivalently, the input to the TSO-level.

If the output of the SLSE of each substation is a voltage phasor for each bus, the TSO-level SE can be made linear, equivalent to the two-stage inclusion of phasor measurements in Section 2.2.5. If, instead, power flows or other non-linear measurements are added to the inputs, the TSO-level becomes non-linear.

As with the two-level SE in MASE (cf. Section 3.1.2), there may also be a question of which measurements should be utilised at which level.

From the above discussion on sub-system choices, different structures for the local SLSE can be imagined:

- Local SE for each linear part of substation, resulting in the output of bus voltage magnitudes and external power flow estimates.
- Local SE for each linear part of substation, resulting in the output of bus voltage magnitudes and phase angles, which is only possible with PMUs [62] [63], rendering a linear TSO-level SE possible.
- Local SE for the whole substation, resulting in the output of bus voltage magnitudes and external power flow estimates, and possibly with voltage phase angles.
- Local SE for the whole substation, resulting in the output of bus voltage magnitudes and phase angles, rendering a linear TSO-level SE possible.

4.3.4.2 Alternative methods

The first SLSE proposed was presented in [64], in which the active power flows of the linear substation were validated using Linear Programming (LP) optimisation. It should be noted, however, that this was proposed as a pre-estimation validation in the control centre of analogue and status data, and not for a two-level SE, but it may still be interesting. Open CBs are modelled as zero-flow measurements, and usage of LP makes it an implicit LAV method (cf. Section 2.2.4), so bad analogue and status data are automatically rejected. The rejected data is assigned high residuals, which is the basis for the identification and correction of the database.

In [4] (also used in [12]), a SE is utilised for complete, non-linear substations, with the aim of exploiting the full measurement redundancy. This method *is* a GSE, using the constrained optimisation for solution of the full measurement set (cf. Section 4.2.2).

The discussion of [4] is, unfortunately, limited to local level. Its position in a two-level SE was left for later work, and this should be discussed and tested; the discussion is not complete without it. Also, the consequence of including non-linear elements in the local SE should be investigated, in terms of speed and convergence. Overall, the method seems to perform well and provide all the advantages of GSE.

A simpler, linear SLSE, not based on GSE, was suggested in [65]. As it is a proper two-level SE method, the extended measurement set of power flows, including any measured CB power flows, is used to estimate the smaller set of external branch power flows, i.e., power flows going out on the branches connected to the substation. These branch power flows are the input for the TSO-level SE.

An important characteristic of linear SLSE is that the quantities of active power, reactive power, voltage magnitude and voltage angle (and possibly current magnitude and angle) are decoupled. The impedance is, through Ohm's law, the connection between power flows and voltages. Thus, no mathematical connection between these quantities can exist if there are only branches with zero impedance, which will have two important consequences:

- The linear estimation may be done separately for P, Q, and possibly θ. This further reduces the matrices involved.
- Observability and redundancy is considered individually for each of these quantities, which means that a redundancy in active power measurements cannot compensate for a lack of redundancy in voltage magnitude measurements.

The SLSE in [65] only pre-processes the raw-measurements, the TSO-level SE software needs minimal adaptation, and the network can be a mix of SLSE and non-SLSE substations. It should be noted that the pre-processed measurements are correlated, so that the weighting matrix no longer will be diagonal, and the resulting weighting matrix can be calculated [65]. If the off-diagonal elements of the weighting matrix are not neglected, a simple adaption of the TSO-level SE software would be required.

The main disadvantage of the SLSE in [65], is that it is not a GSE. The simplistic modelling only works on radially operated substations, which is not discussed in [65]. Also, without GSE, CB statuses are deterministic: lines with closed CBs are omitted from the formulation, rendering the validation of their reported statuses impossible. In summary, the simplicity of [65] produces an efficient and mathematically intuitive formulation, but also limits the application to radial substations and inhibits the topology error capabilities.

4.3.5 Proposing a Substation Level State Estimation

The non-linear GSE-based SLSE, from [4], seems a promising candidate for a two-level SE scheme. But, it is expected that quite a few transmission substations will only contain linear elements. This occurs either if the substation only has a single voltage level (switching substation) or if it has only one voltage level that belongs to the transmission system (a primary distribution substation, in which the HV section is included in SE, while the distribution transformer is modelled simply as a load).

The problem of power and voltage decoupling that occurs in linear substations, which changes the rules for redundancy and observability, was discussed in the preceding section for linear SLSEs. However, this problem also exists for non-linear SLSEs, i.e., in all the linear substations, an aspect that was not discussed in [4]. Also, another aspect that was not discussed is that in a nearly-linear substation, with only one or two non-linear elements, the impedances of the non-linear elements would be the only coupling factors. That could make the estimation erroneous because of the parameter errors of those elements.

These new observations lead to the proposed two-level SE in this thesis: A linear GSE is executed locally, sending pseudo-measurements of power flow and voltage magnitudes to the TSO-level, which consists of a conventional SE. The substation GSE simply pre-processes the raw measurements, sending a few, but verified, pseudo-measurements to the TSO. In this way, the established SE software in the control centres would only need minor adaptions.

For a completely general model, the selection of GSE state variables can be complicated [45], but the substations are much more similar, and hence a simple state variable selection method could be utilised. In the linear substation, all internal branches are zero-impedance branches. Each internal branch has two state variables associated with it – the active and reactive component of power flow through that branch. Additionally, voltages at every node must be included in the state vector. Phase angles are not included, when phasor measurements are assumed locally. The CBs with unknown statuses cannot be modelled with (4.6), because the pseudo-measurement equation is non-linear, and so a most likely status must rather be assumed and corrected.

If phasor measurements are available in the system, either from PMUs or protection IEDs, local utilisation seems difficult in the linear regime, because current and angle are two new de-

coupled quantities and cannot help in the estimation of power or voltage, and hence cannot enhance local BD or topology processing, which definitely is a drawback. Thus, TSO-level utilisation of PMUs is more likely.

Finally, the advantage of a linear model is that no iterations are needed, and hence no issue of convergence, and that the Jacobian is constant, so long as the measurement set (analogue and status data) does not change. This makes for an efficient and reliable pre-processing.

The next chapter explains the proposed estimator in detail.

Chapter 5

Implementation

The preceding chapter ended with a suggestion for a new linear, GSE-based SLSE. This chapter describes in detail a proposed implementation logic and design choices, for the SLSE in a two-level scheme and also the test program. The test program includes LF, error generator, and other functions needed for conceptual testing.

5.1 Test System

The test system considered is based on the expanded 14-bus IEEE test system shown in Fig. 5.1, because this expanded network has been considered in similar studies [5]. The expansion to node-breaker detail was done in [5] by using some of the typical substation topologies presented in Section 3.1.2, i.e., the ring bus, double breaker, breaker-and-a-half and transfer bus configurations.

To make the conceptual tests easier, only the high voltage substations are considered, i.e., substations 1 to 5 in Fig. 5.1. To keep test conditions as close to the standard operating conditions as possible, border power flows between high voltage and low voltage buses are included as loads in substations 4 and 5, which are modelled as PQ-constrained buses in the load flow. The load and generation of each substation sums up to the injected power of each bus as in the original 14-bus system.

The final test system is shown in Fig. 5.2, with numbering of substations from 1 to 5 and with numbering of nodes starting with 1 in every substation. It should be noted that with the CB

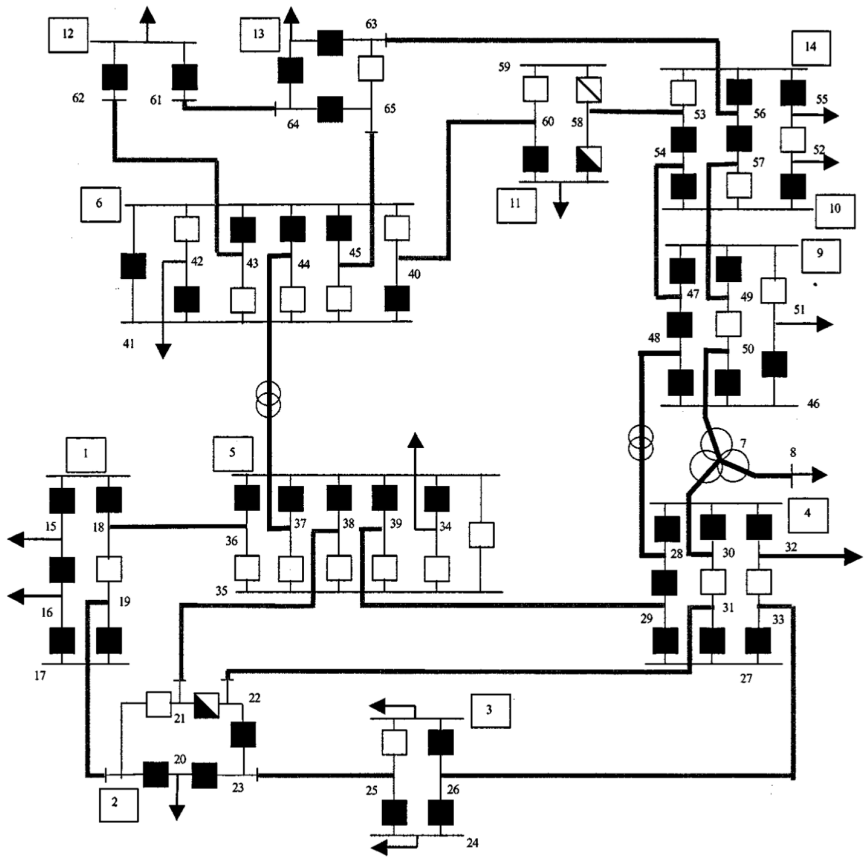


Figure 5.1: The expanded 14-bus IEEE test system, as used in [5]. A different node numbering is used in this thesis.

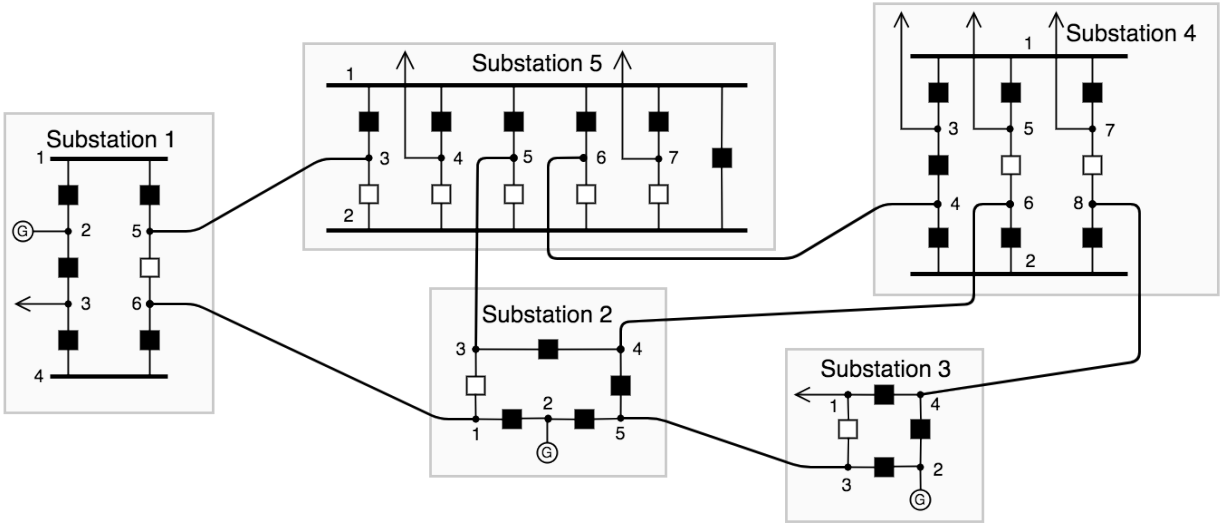


Figure 5.2: The chosen 5-substation test system, based on bus 1 to 5 of the test system in Fig. 5.1.

statuses as shown in Fig. 5.2, each substation is connected as one bus, i.e., with all nodes connected through closed CBs. Therefore, the terms substation and bus are used interchangeably in the following.

Since a two-level approach is applied, the substations are used as subsystems in the SLSE. The structure of the two-level scheme is illustrated in Fig. 5.3.

The Excel file for the system data is given in Appendix C.

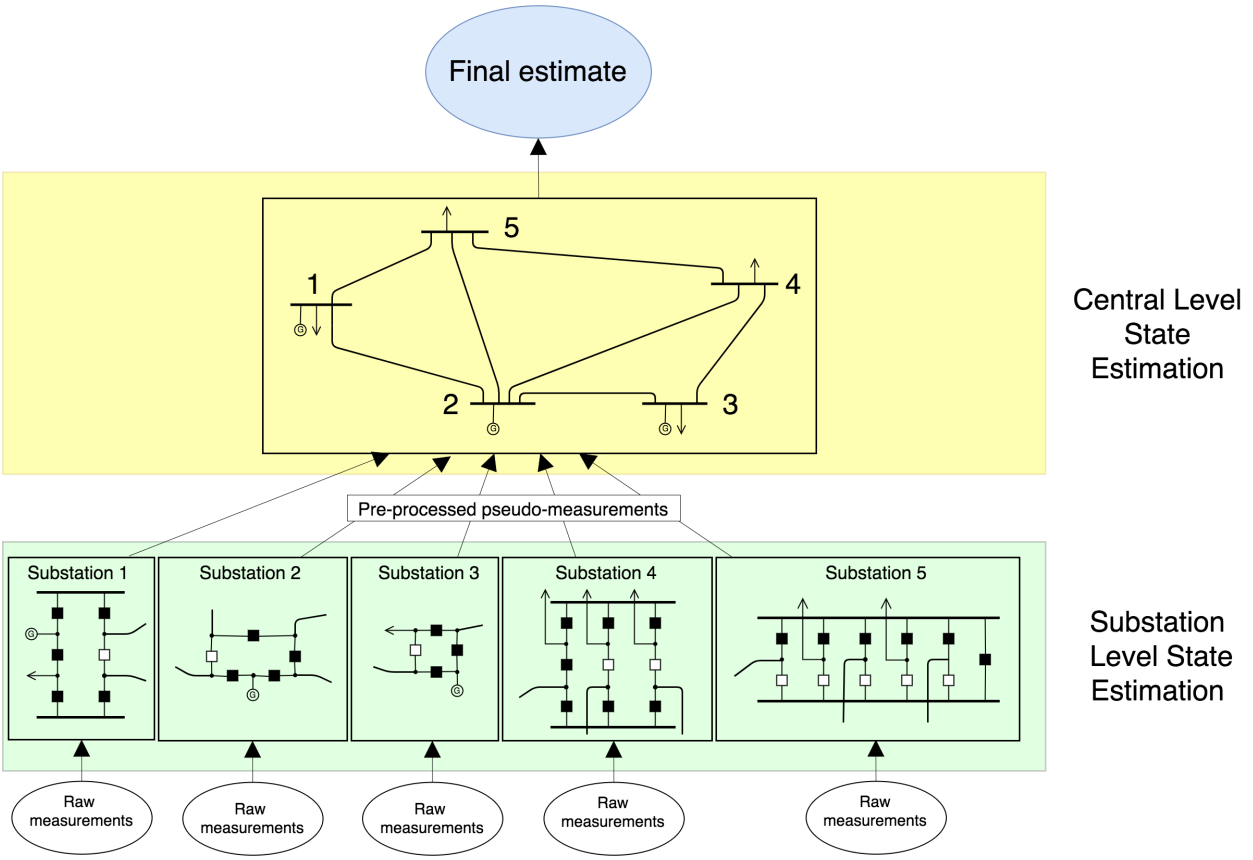


Figure 5.3: Principle of the two-level scheme, showing subsystems.

5.2 Test Program Structure

The SLSE is to undergo conceptual tests, according to the goals in Section 1.2. The structure of the test program is illustrated in Fig. 5.4.

In Fig. 5.4, the LF and measurement generator create measurements, at the node-breaker level. The local estimator is the SLSE, taking raw-measurements as input and building verified

measurements for the TSO-level. The TSO-level state estimator applies these measurements using a conventional WLS SE. A conventional SE is also run for the raw measurements, for comparison, which makes a topology processor necessary for allocating the raw-measurements to the bus-branch model. The following sections describe these programs, as well as the validation testing conducted.

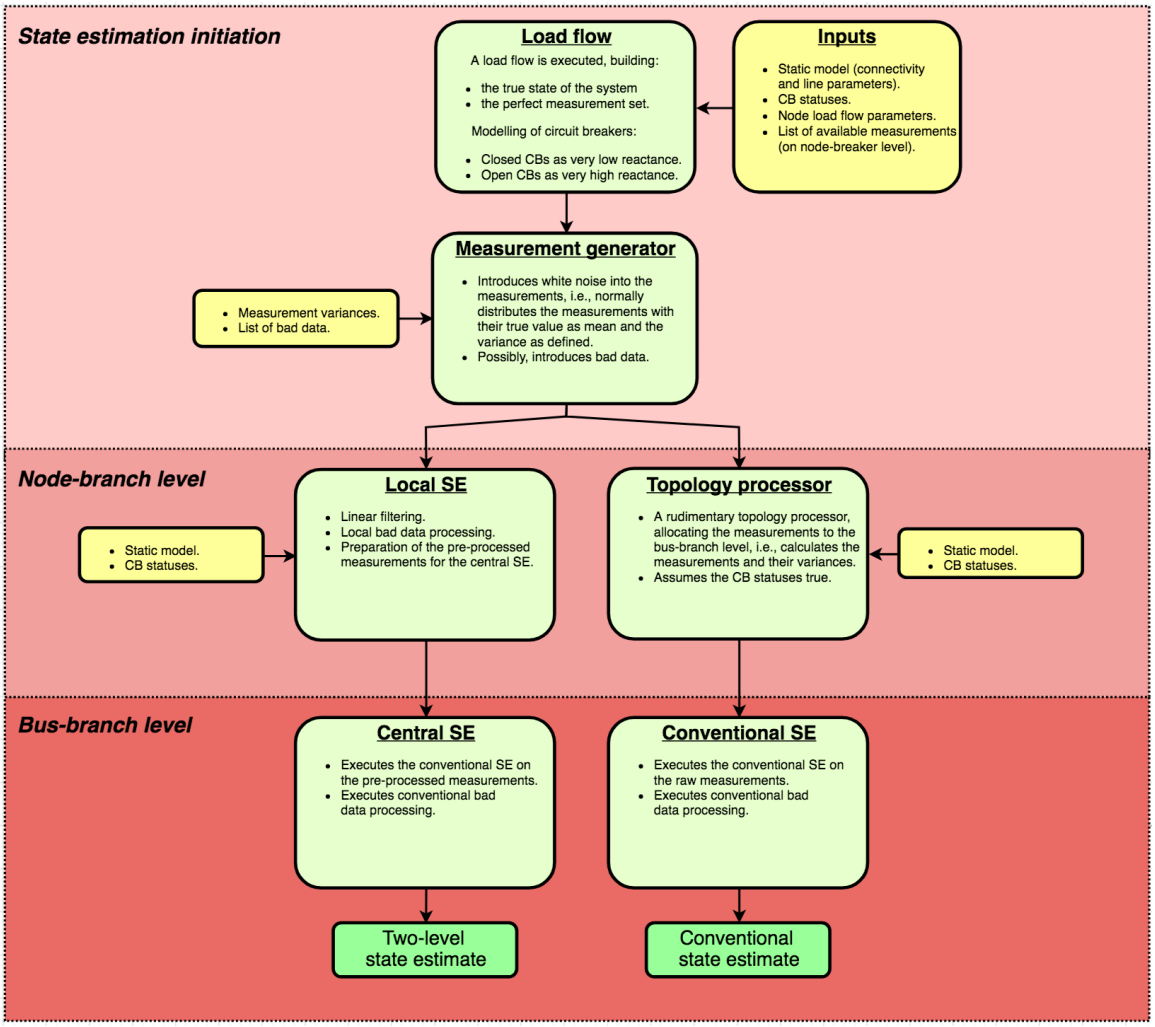


Figure 5.4: The organisation of the test program.

5.3 Measurement generator

The raw measurements are composed of the load flow results with uncorrelated white noise with fixed variance, according to the assumptions of SE described in Section 2.2.3. BD can be

introduced, overwriting the actual measurement. Any other sources of error, such as measurement time skew and transients, are not considered explicitly but assumed incorporated in the noise.

Inputs for power flow model and measurements were given in Excel files. This solution was chosen because it gives a well-arranged presentation and any changes can be made easily. Also, Excel files are easily read into MATLAB. The files are given in Appendix C.

The *lines.xlsx* file contains the list of all lines, including zero-impedance lines within the substation. The zero impedance lines are all assumed to have a breaker, and if nothing else is mentioned, the breaker is assumed to be closed.

In *operation.xlsx* every node in each substation is specified. It should be noted that every node is assumed to have only one purpose: it is either connected to a load, a generator, a transmission line or simply a connection between other nodes (zero-injection node). The following node types are used:

1. Load node, used as a PQ bus in the LF and an injection in the SE, e.g., node 3 in substation 1.

2. Generator node, used as a PV bus in the LF and an injection in the SE, e.g., node 2 in substation 3.

3. Slack node, used as a $P\delta$ bus in the LF and an injection in the SE, e.g., node 2 in substation 1.

4. Zero-injection node, used as a PQ bus with zero-injections in the LF and as a certain zero-injection in the SE, e.g., nodes 1 and 4 in substation 1.

5. Bay node, i.e., node connected to a transmission line, used as a PQ bus with zero injections in the LF and as a certain zero-injection in the SE, e.g., nodes 5 and 6 in substation 1.

Matpower is chosen to conduct the load flow, so as to enable load flow within the MATLAB platform. Although zero-impedance branches are not handled by Matpower load flow, closed CBs are modelled as a very small reactance and open CBs as a very high reactance (both with zero resistance), resulting in negligible voltage drop over closed CBs and negligible power flow

through open CBs. This is not a very robust approach, since load flow programs can reach numerical instability under very different branch impedances, but in the application of creating measurements, no problems have been noticed.

The third Excel file acting as input is *measurement.xlsx*. It contains, for each substation, the list of what measurements should be included. No value or variance has to be specified, since the measurement values are taken from LF results and the program automatically assigns variance after measurement type. However, if a value is specified, e.g., to simulate a BD, the true measurement is overwritten. Variances can also be assigned individually in the Excel file. Also specified in the *measurement.xlsx* file are the CBs that are open. This is used to determine if a very low or very high reactance is to be used for the zero-impedance branches in the LF, and is also used to produce the required pseudo-measurements.

In the SLSE, high-weight pseudo-measurements are generated. The zero-injection nodes and bay nodes, node types 4 and 5 above, are both assigned zero active and reactive power measurements. For each closed and open CB, the appropriate pseudo-measurement from Section 4.2.1 is constructed.

5.4 Local state estimator

The local state estimator is a linear GSE. The estimations of voltage magnitude, active power and reactive power are, as discussed, decoupled problems in the linear formulation, and can therefore be solved individually, to a certain degree. Using substation 1 in the test system as an example, the local state estimator is explained in the following.

5.4.1 Voltage Measurement Jacobian

The voltage estimation is quite trivial. The state vector consists of the voltages of all the substation nodes:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} \tag{5.1}$$

These correspond to the columns of the Jacobian.

The measurement set consists of the voltage magnitude measurements, in addition to the high-weight pseudo-measurements of zero voltage magnitude difference over closed CBs. A voltage measurement, e.g., of node 4, is simply

$$v_4^m = x_4 = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix} \boldsymbol{x} = \boldsymbol{J}_{v_4} \boldsymbol{x} \tag{5.2}$$

making $\boldsymbol{J}_{v_4} = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}$ the corresponding row of the Jacobian matrix, $\boldsymbol{H}$, and v_4^m the corresponding measurement value in $\boldsymbol{z}$.

A closed CB pseudo-measurement (zero voltage difference), e.g., of the CB between node 2 and 3, gives

$$0 = v'_{2,3} = x_4 = \begin{bmatrix} 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{x} = \boldsymbol{J}_{v'_{2,3}} \boldsymbol{x} \tag{5.3}$$

making $\boldsymbol{J}_{v'_{2,3}}$ the corresponding row of the Jacobian matrix, $\boldsymbol{H}$, and setting the corresponding measurement value in $\boldsymbol{z}$ to 0.

5.4.2 Active and Reactive Power Measurement Jacobian

The estimation of active and reactive power is identical, so the following discussion on the active sub-problem is directly transferable to the reactive sub-problem.

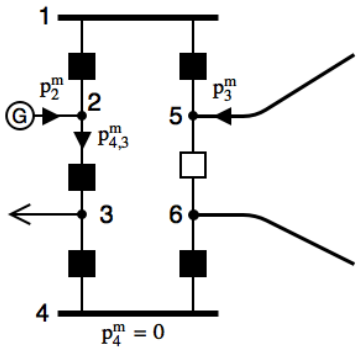


Figure 5.5: Some example active power measurements in substation 1.

The state variables are all the internal power flows, which in the case of substation 1 are

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} p_{1,2} \\ p_{1,5} \\ p_{2,3} \\ p_{3,4} \\ p_{4,6} \\ p_{5,6} \end{bmatrix} \tag{5.4}$$

where $p_{1,5}$ is the internal power flow from node 1 to 5.

The measurements are then used to build the Jacobian matrix, $\boldsymbol{H}$. The Jacobian for the local level consists of only ones and zeros. Any measurement of the CB power flow can naturally be represented by only one non-zero element. For example, a measurement of the power flow from node 4 to node 3 is simply equal to negative of the corresponding state variable, x_4, resulting in the following row in the Jacobian:

$$p_{4,3}^m = -p_{3,4} = -x_4 = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix} \boldsymbol{x} = \boldsymbol{J}_{p_{4,3}} \boldsymbol{x} \tag{5.5}$$

Also, open CBs introduce high-weight pseudo-measurements for zero power flow through CBs, and thus the formulation for these is identical.

The Jacobian for the power injection at node 2 is built in the same manner, giving the follow-

ing row in the Jacobian,

$$p_2^m = -p_{1,2} + p_{2,3} = -x_1 + x_3 = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{x} = \boldsymbol{J}_{p_2} \boldsymbol{x} \tag{5.6}$$

The zero-injection virtual measurements applied for zero-injection nodes, e.g., nodes 1 and 4 in substation 1, are naturally formulated in the same manner. Also the Jacobian for power injections and for external power flows are identically built. Since these are both locally viewed as injections, the formulation of the Jacobian is done identically for nodes 5 and 6. However, it is imperative to be wary of the signs, so that the injection in node 5 is equal to the negative of the measured outgoing power flow, i.e., $p_5^m = -P_{1,5}^m$ (uppercase letters denote external power flow, with the substation numbers as subscript).

5.4.3 Estimation procedure

In Section 4.2.2 two appropriate solution methods were reviewed, namely a numerically robust estimation method and a constrained optimisation method. The numerically robust method of orthogonal transformations is chosen in this program, because of the easy adaptation needed for application. Also, the appropriate function is available in the MATLAB environment.

The implementation is quite straightforward. First, the pre-multiplied Jacobian matrix and measurement vectors are calculated:

$$\tilde{\boldsymbol{H}} = \boldsymbol{W}^{0.5} \boldsymbol{H} \tag{5.7}$$

$$\tilde{\boldsymbol{z}} = \boldsymbol{W}^{0.5} \boldsymbol{z} \tag{5.8}$$

The MATLAB function $qr(\tilde{\boldsymbol{H}})$ returns the full matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, from which the sub-matrices $\boldsymbol{Q}_1$ and $\boldsymbol{U}$ are extracted. The state vector is then found in two stages, without iteration, using the linear version of (4.12) and (4.13), i.e., first calculating

$$\boldsymbol{y}_1 = \boldsymbol{Q}_1 \tilde{\boldsymbol{z}} \tag{5.9}$$

and thereafter solving

$$\boldsymbol{U} \hat{\boldsymbol{x}} = \boldsymbol{y}_1 \tag{5.10}$$

for $\boldsymbol{x}$, using the Cholesky factorisation from Section 2.2.4 for efficiency. Since the problem is linear, no iterations are needed.

5.4.4 Assumptions and simplifications

The transmission network is designed as a three-phase system. In analysis and planning, most methods rely either on single-phase equivalents, when both network and load/generation are balanced, or on sequence analysis, when load/generation is unbalanced. Both these methods assume that network is symmetrical and that the three phases have no mutual coupling. In situations where this is not the case, a full three-phase analysis may be needed.

When incorporating the full measurement set of a substation, there would be three different approaches [4]:

- Single-phase model, assuming the single-phase measurements being of the same quantity, so that there are three measurements available for the same quantity.
- Single-phase model, using the three single-phase measurements to calculate the direct sequence measurement, which is then applied in the SE.
- Three-phase model, assuming each singe-phase measurement error independently and unaggregated.

The advantage of using the three-phase model, as in the non-linear SLSE in [4], is that all kinds of measurements can be utilised; unbalances and mutual coupling are also taken into consideration. However, for a linear network, such as the focus here is, there is no difference between using a single-phase model for each phase separately and using a three-phase model, the phases are decoupled anyway. Therefore, a single-phase model is applied in the testing.

Since the implementation and the testing are done in software only, with LF-based measurements, some of the assumptions worth mentioning are (cf. Section 2.2.3):

- Balanced three-phase system, steady-state and perfectly sinusoidal signal.
- Observability is guaranteed, for voltage, active power and reactive power individually, through utilisisation of the full measurement set.

- Perfect synchronisation of measurements, i.e., all measurements are made in the same time-instant.
- Perfect knowledge of line parameters.
- Measurements are uncorrelated, with white noise errors of a known and fixed variance.

The assumption of fixed error variance means that the errors generated for small quantity measurements in relative terms are very high, which is a necessary but maybe unrealistic assumption. One consequence of this is that the relatively small reactive power flows, and the coupled voltage phase angles, are harder to estimate accurately.

5.4.5 Local bad data processing

BD, both in analogue measurements and CB status, is considered. The conventional BD methods, presented in Section 2.3, are adapted to the linear scheme, using the following matrices, for the active sub-problem:

$$\boldsymbol{r}_P = \boldsymbol{z}_P - \boldsymbol{H}_P \boldsymbol{x}_P \tag{5.11}$$

$$\boldsymbol{\Omega}_P = \boldsymbol{W}_P^{-1} - \boldsymbol{H}_P \boldsymbol{G}_P^{-1} \boldsymbol{H}_P^T \tag{5.12}$$

The normalised residuals are then

$$r_j^N = \frac{r_j}{\sqrt{\Omega_{jj}}} \tag{5.13}$$

and the chi-squared detection and normalised residual identification is applied (cf. Section 2.3). Since the active and the reactive estimations are decoupled, the chi-squared limit of each sub-problem is calculated using only the degree of freedom in that sub-problem, i.e., the number of measurements minus the number variables. Also, a 99 % detection limit is used, i.e., the theoretical probability of a false detection is set to 1 %.

The BD processing of the active and the reactive sub-problems should not be implemented independently, because they are connected by the CB constraints. For example, if a closed CB pseudo-measurement is identified as BD in the active sub-problem ($p_{kl} = 0$ for the closed CB between node k and l), then the equivalent pseudo-measurement in the reactive sub-problem

($q_{kl} = 0$) should be removed. The BD processing of the voltage sub-problem and the active and reactive power sub-problems are coupled, since, in the correction of a CB status, all three sub-problems are changed: in the correction from open CB status to closed, the voltage pseudo-measurement should be removed and the active and the reactive pseudo-measurements of the closed CB should be added.

in the active and the reactive problems as well as the addition of an closed CB pseudo-measurement in the voltage sub-problem.

Removal of bad statuses in the right order is important for the normalised residuals method, since the second largest residual often may not be BD when the largest residual measurement is removed. The fact that pseudo-measurements couple BD processing of the active, reactive and voltage sub-problems, has led to the decision of removing only the highest normalised residual of the combined system, if BD is detected in multiple sub-problems.

Fig. 5.6 gives the structure of the SLSE implementation, including the error processing.

5.4.6 Building the measurement set for the TSO-level SE

The result of the SLSE in this two-level scheme is the estimated state, i.e., the voltage magnitudes and the active and reactive power flows, but also, indirectly, the verified CB statuses. First, the verified CB statuses are used to determine which set of nodes can be aggregated into buses: in the test system considered in Section 5.1, each substation is aggregated to a single bus.

The SLSE states and the topology will be used to build pre-processed measurements that are used at the TSO-level. The process of building these measurements is described in the following, explaining how the measurements are allocated to the elements of the bus-branch model, and how their measurement values and corresponding variances are calculated.

5.4.6.1 Building bus voltage measurements

For voltage magnitude, the weighted mean (using the calculated covariance matrix) of the node voltages, of the nodes constituting the bus, is taken as measurement at the TSO-level. In substations consisting of several buses, this calculation is not straight-forward. In general, statistical theory gives, for bus i [66]

$$\sigma^2_{V_i} = (\boldsymbol{D}^T \ \boldsymbol{\Sigma}_{v^{s_i}} \ \boldsymbol{D})^{-1} \tag{5.14}$$

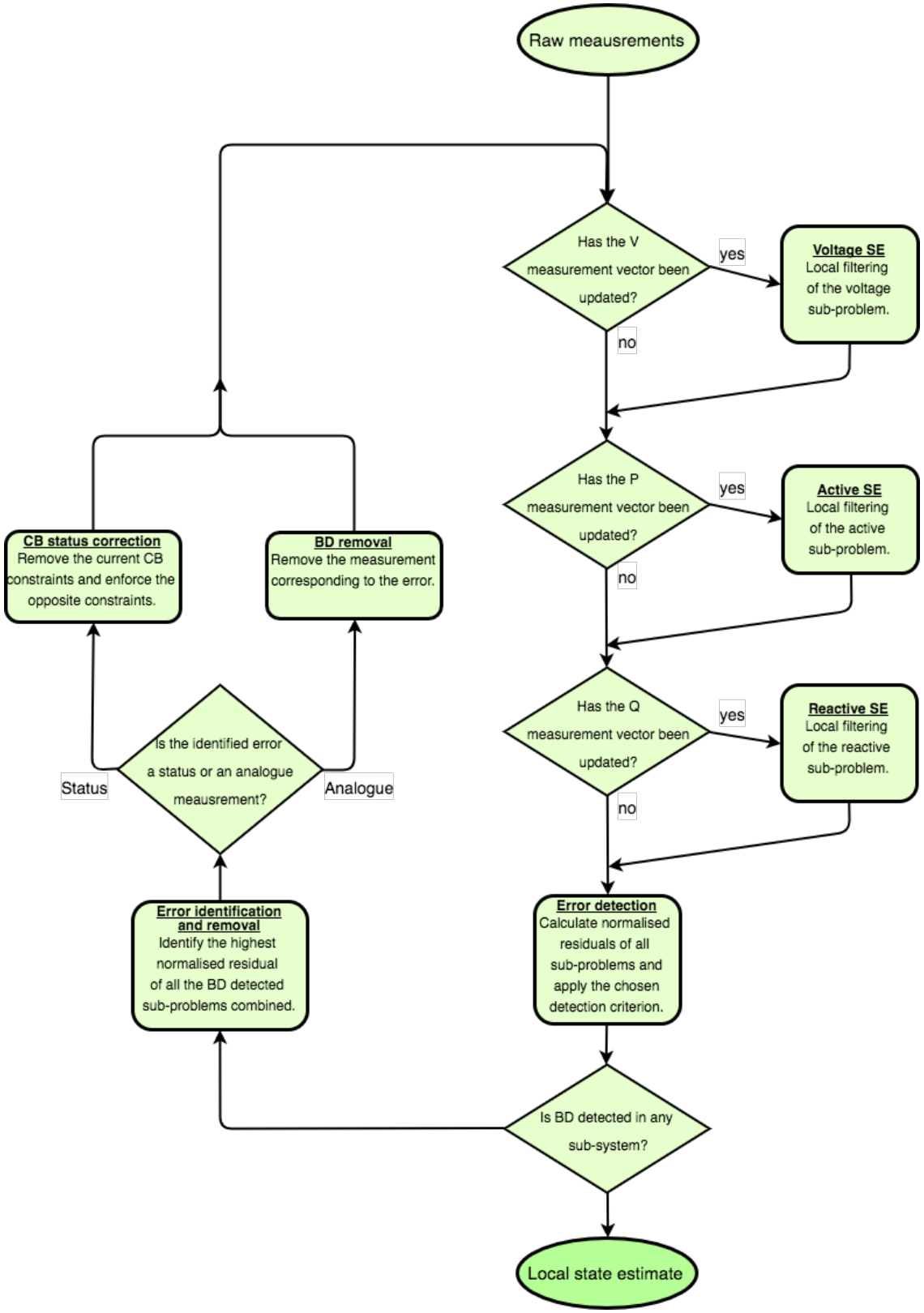


Figure 5.6: The workflow of the substation level state estimation, including the local BD processing.

and

$$V_i = \sigma^2_{V_i}(\boldsymbol{D}^T \ \ \boldsymbol{\Sigma}_{v^{s_i}} \ \ \hat{\boldsymbol{v}}_{s_i}) \tag{5.15}$$

where

- i is the bus number for which the voltage pseudo-measurement at the TSO-level is being calculated;
- s_i is the set of nodes in the substation which make up a bus, i.e., the set of nodes connected by closed CBs;
- $D = [1, 1, ..., 1]^T$ is a vector with the same length as s_i;
- $\hat{\boldsymbol{v}}_{s_i}$ is the subset of the the SLSE voltage estimate ($\hat{\boldsymbol{v}}$) for s_i, the nodes constituting bus i; and
- $\boldsymbol{\Sigma}_v$ is the covariance matrix of the estimated node voltage vector $\hat{\boldsymbol{v}}$, which makes $\boldsymbol{\Sigma}_{v^{s_i}}$ the sub-matrix for elements s_i, the nodes constituting bus i.

The implementation has shown that an easier formulation for building the voltage measurement is possible, in which ignoring the covariances produces the exact same answer. From simple statistics and the mean of uncorrelated, normally distributed values, the measurement is calculated as [67]:

$$V_i = \frac{1}{n_{s_i}} \sum_{s_i} v^{s_i} \tag{5.16}$$

$$\sigma^2_{V_i} = \frac{1}{n_{s_i}} \sum_{s_i} \sigma^2_V \tag{5.17}$$

5.4.6.2 Building power measurements

For every substation, the transformation matrix $\boldsymbol{H}_{P_{inj}}$ is calculated, which transforms the power state vector, consisting of all the substation-internal power flows, to the estimated power injected at all the nodes:

$$\boldsymbol{p}_{inj} = \boldsymbol{H}_{p_{inj}} \ \boldsymbol{x} \ \boldsymbol{H}^T_{P_{inj}} \tag{5.18}$$

The corresponding covariance matrix follows as

$$\boldsymbol{\Sigma}_{\boldsymbol{p}_{inj}} = \boldsymbol{H}_{p_{inj}} \, \boldsymbol{\Sigma}_x \, \boldsymbol{H}_{p_{inj}}^T \tag{5.19}$$

where $\boldsymbol{\Sigma}_x$, the variance of the (active power) state vector estimate can be computed as the inverse of the gain: $\boldsymbol{\Sigma}_x = [\boldsymbol{H}^T \boldsymbol{W} \boldsymbol{H}]$.

For example, consider for substation 1,

$$\boldsymbol{p}_{inj} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{bmatrix} \tag{5.20}$$

where p_1 and p_6 will be close to zero (since they are zero-injection nodes), p_2 and p_3 are the actual power injections from load and generation, and $p_5 = -P_{1,5}$ and $p_6 = -P_{1,2}$ are injected powers from the external branches.

Bus power injections may consist of several elements of the injection vector $\boldsymbol{\Sigma}_{\boldsymbol{p}_{inj}}$, as in substation 1, where both node 2 and node 3 are load/generation nodes. The injected power measurement sent to the TSO-level is the sum of estimated injection measurements of the nodes constituting the bus, e.g., for substation 1:

$$P_1 = p_2 + p_3 \tag{5.21}$$

The variance is more complicated to calculate. If the bus has two nodes with load and/or generation, great care must be taken in calculating the variance of the TSO-level injection measurement. This is because the local state variables are correlated, i.e., the co-variance matrix $\boldsymbol{\Sigma}_{\boldsymbol{p}_{inj}}$ is a full matrix, including co-variances in the off-diagonal terms.

For uncorrelated variables, the variance of their sum is simply the sum of the individual variances. For two correlated variables, however, the variance of their sum is known in statistics

[67] as

$$\sigma^2_{P_{inj}} = \sigma^2_{p_1} + \sigma^2_{p_2} + 2 * \sigma_{p_1}\sigma_{p_2} \tag{5.22}$$

where $\sigma^2_{p_i^m}$ is the diagonal element of the co-variance matrix $\mathbf{\Sigma}_{\boldsymbol{p}_{inj}}$. In general, for more than two injection nodes, the variance is

$$var(\sum_{i}^{N} X_i) = \sum_{i}^{N}\sum_{j}^{N} cov(X_i, X_j) \tag{5.23}$$

which is the formula utilised for calculating the variance of bus power injection pseudo-measurements for the TSO-level.

A small error in the calculated variances for the TSO-level power measurements is introduced by the fact that virtual measurements and CB pseudo-measurements are assigned a variance, but are not subject to noise as the regular measurements. If this artificial variance were to be significant, it would alter the variance, and hence the weighting, of the TSO-level measurements, leading to a non-optimal solution. As as can be seen in the next chapter, the variance is very much negligible and thus does not carry any significance.

Also, it could be noted that only the variance of the pseudo-measurements is calculated, although the pseudo-measurements of a bus will be correlated because of which some covariance is inherent. The pseudo- measurement co-variances are neglected to avoid off-diagonal elements in the TSO-level SE, so as to keep the TSO-level SE software intact. The effect of this approximation will be discussed duly.

5.5 TSO-level State Estimator

The TSO-level estimator is a conventional estimator, as described in Chapter 2, which is a strength of the two-level approach. The TSO's SE software can be kept intact, with the SLSE simply acting as a pre-processor.

The basic techniques and the formulae for measurement equations and measurement Jacobian are given in Appendix A. The equations for phasor measurements are also included in the program.

The basis for the WLS estimation algorithm was adapted from a script publicly available on

exchange communities [68], especially regarding the structure of data. The full matrix set is recalculated for each iteration, something that is also often done in practical implementation [12], although more efficient techniques could be applied. The normal equation is solved using Cholesky-factorisation (cf. Section 2.2), which is more efficient than direct inversion for sparse systems.

The BD processing is basically the conventional method, as described in Section 2.3. After each full estimation, BD detection is done with the chi-squared method, and the BD, when present, is identified and removed from the measurement set. The WLS solver is then applied again for the new measurement set.

5.6 Topology Processor and Conventional State Estimator

To be able to run a parallel conventional SE, a conventional state estimator has been implemented. To ensure the comparability of the conventional SE and the two-level SE, the measurement set for the conventional estimator is built on the same measurements as the two-level estimator.

Since the raw-measurements are, as in the real power system, made in the node-breaker model, a rudimentary topology processor has been implemented to transform the measurements to the bus-branch level. This is a simple process, described in the following for the different measurement types.

Power flows on external branches are transferred directly. Bus injections are only included only if measurements of the all the individual nodes (except nodes of type 4 and 5, which have zero injection) in the substation are available, according to the discussion in Section 4.1.1. The bus injections are calculated as the sum of node injections and their variances are simply the sum of node injection variances, since the errors are uncorrelated.

For voltage measurements, the measurement value is simply the (weighted) mean of the node measurements, and the variance is the sum of variances divided by the number of voltage measurements.

The conventional SE is the same program as the TSO-level SE, which was discussed in the preceding section.

Chapter 6

Simulation and Testing

This chapter demonstrates the application of the SLSE and the two-level SE described in the preceding chapter. Validation and conceptual testing of the SLSE, and also comparison of the two-level SE with the conventional SE, are performed.

6.1 Parameters and Measurements

Table 6.1 presents the list of measurements used and the standard deviations assigned to them. The mark (') denotes zero-injection virtual measurements and CB pseudo-measurements. In the representation p_{kl} and q_{kl}, the lower case letters are used to represent measurements of the node-breaker model, with k and l being nodes; in this case these are power flows through the CB between k and l. Their uppercase equivalents, P_{ij} and Q_{ij}, are measurements of the bus-branch model; in this case these are transmission line power flows from sending bus i towards receiving bus j.

The standard deviations listed provide weighting in the SE, but since normally distributed noise and perfect knowledge of variance are assumed (cf. Section 5.4.4), these are also the standard deviations of the noise added to the ideal measurements. Naturally, no noise is added to the zero-injection virtual measurements and the CB pseudo-measurements.

A main objective of the two-level SE is the possibility of exploiting the local redundancy with the SLSE. Stemming from protection VTs and CTs, the availability of CB power flow measurements is assumed to be utilised by the SLSE here (given in Table 6.1 as p_{kl} and q_{kl}). These

Table 6.1: The standard set of measurements used in the tests, where i and j are bus numbers and k and l are node numbers.

Measurement type	**Measurement standard deviation**
v_k	0.005
p_k	0.01
q_k	0.01
P_{ij}	0.01
Q_{ij}	0.01
p'_k	10^{-8}
q'_k	10^{-8}
p'_{kl}	10^{-8}
q'_{kl}	10^{-8}
θ'_{kl}	10^{-8}
V'_{kl}	10^{-8}
p_{kl}	0.05
q_{kl}	0.05

measurement are less accurate, but should provide a slightly increased redundancy at the local level. It should, however, be noted that the increased standard deviation of the protection measurements by a factor of 5, results in increased variance by a factor of 25, i.e., the regular measurements are weighted 25 times higher in the local SE than the protection measurements. Two measurement cases are considered in the analysis, and their excel files are also given in Appendix C:

Case 1 Good bus-branch redundancy: voltage measurements at all the nodes, active and reactive injection measurements at all the relevant nodes (nodes connected to load or generator), active and reactive injection measurements on both sides of every transmission line (branches of non-zero impedance), and additionally, only for the local SE, the less accurate protection measurements of active and reactive power in all the CBs. The measurement set includes:

- voltages at all nodes;
- all transmission line power flows, from both terminals;
- all injected powers, and;

- all CB power flows.

Case 2 Limited bus-branch redundancy: reduced number of measurements for voltages, power injections and power flows compared to case 1. It should be noted that the loss of node injections in substations 1 and 4 make the bus injection measurements unavailable for the conventional SE. The measurement set includes:

- voltages at two nodes per substation;
- one transmission line power flow per substation;
- all node injections except in two nodes, and;
- all CB power flows.

For the local level, the GSE pseudo-measurements, i.e., zero-injections and CB constraints, are also added to the local measurement set.

6.2 Purely noisy conditions

The two-level SE is demonstrated with purely noisy measurements in the following. Also, the two-level SE is tested and compared with the conventional SE.

6.2.1 Demonstrating the Substation Level State Estimator

The local SLSE is executed, with the case 1 measurement set. The only source of measurement error is noise.

The results of the active power sub-problem of an arbitrary execution are shown in Table 6.2. Rows 1 through 5 correspond to the measurements directly transferable to the bus-breaker level. Measurements 6 through 10 correspond to protection measurements. These CB power flows also make up the local state vector, according to the GSE-formulation. The last row shows the virtual CB measurements, based on the (reported) open status of the CBs.

In the left part of Table 6.2, the columns show the true power flows, the raw measurements, where noise has been added, and the error of the raw measurements. In the right part, the pre-processed measurements and their errors are shown.

Table 6.2: Errors in raw measurements (left) and pre-processed measurements (right).

Measurement	z^{ideal}	z^m	$\lvert z^m - z^{ideal}\rvert \times 10^{-3}$
P_2	0.183	0.182	1.0
$P_{2,1}$	-0.940	-0.963	22.6
$P_{2,5}$	0.195	0.193	1.7
$P_{2,4}$	0.318	0.323	5.3
$P_{2,3}$	0.611	0.605	5.5
$p_{1,2}$	0.940	0.999	59.0
$p_{1,3}$	0.000	-0.039	39.0
$p_{2,5}$	1.123	1.101	22.0
$p_{4,5}$	-0.512	-0.408	104.4
$p_{3,4}$	-0.195	-0.271	76.3
$p'_{1,3}$	0	0	0

Measurement	$\hat{z}$	$\lvert \tilde{z} - z^{ideal}\rvert \times 10^{-3}$
P_2	0.175	8.0
$P_{2,1}$	-0.957	17.2
$P_{2,5}$	0.197	2.5
$P_{2,4}$	0.324	6.5
$P_{2,3}$	0.611	0.2
$p_{1,2}$	0.957	17.2
$p_{1,3}$	-0.000	0.0
$p_{2,5}$	1.132	9.2
$p_{4,5}$	-0.521	9.0
$p_{3,4}$	-0.000	0.0
$p'_{1,3}$	-0.000	0.0

Elements 1 through 5 of the $\tilde{z}$ column are the pseudo-measurements for the TSO-level. It can be seen that in this specific case, the bigger error of raw measurement $P_{2,1}$ is reduced, but at the cost of a slight increase in error in the other pseudo-measurements.

Though this demonstration is interesting for insight into the estimation procedure, bigger samples must be used to be able to draw any detailed conclusions.

The SLSE is executed using 100,000 runs of Monte Carlo Simulations (MCS), with measurement case 1 and without BD or BD correction. In Fig. 6.1, the box-plots of pre-processed and raw measurements are presented. This illustrates the local estimation, but should not be interpreted as two-level method being more accurate than the conventional method, but rather that some part of the estimation successfully takes place in the substation. Also, the mean is zero,

Table 6.3: Pre-processed measurement set sent to the TSO-level SE.

Measurement	z^{ideal}	z	σ_{z^m}
P_2	0.183	0.182	0.0088
$P_{2,1}$	-0.940	-0.963	0.0087
$P_{2,5}$	0.195	0.193	0.0087
$P_{2,4}$	0.318	0.323	0.0088
$P_{2,3}$	0.611	0.605	0.0089

Measurement	z^{ideal}	z^m	$\tilde{\sigma}_{\tilde{z}}$
P_2	0.183	0.182	0.01
$P_{2,1}$	-0.940	-0.963	0.01
$P_{2,5}$	0.195	0.193	0.01
$P_{2,4}$	0.318	0.323	0.01
$P_{2,3}$	0.611	0.605	0.01

confirming that no errors in the algorithm bias the results.

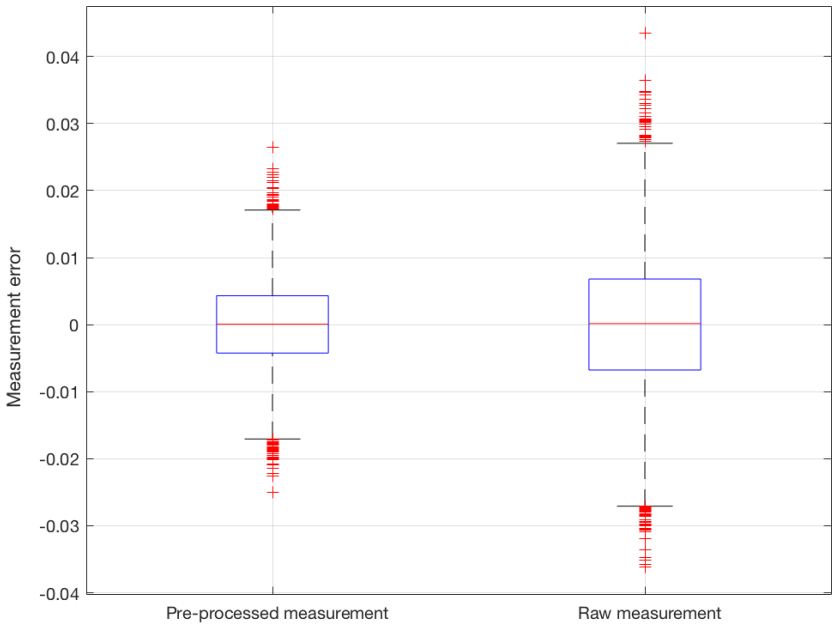


Figure 6.1: Absolute errors of the active power flow from bus 2 to bus 1, resulting from 100,000 runs of MCS.

The predicted variance of the pre-processed measurements, as calculated in Section 5.4.6, are used as weights in the TSO-level SE. These should therefore be as accurate as possible. As displayed in Table 6.4, the simulated variances (given in the table as standard deviations) after 100,000 runs validate approximately the statistics from which the variances were calculated.

Table 6.4: Comparison between calculated standard deviation, used for weighting in the TSO-level SE, and the actual standard deviation, calculated from the simulations, in the pre-processed measurements from substation 2.

Measurement	σ^m	$\sqrt{var(z^m)}$
P_2	0.0088	0.0089
$P_{2,1}$	0.0087	0.0088
$P_{2,5}$	0.0087	0.0088
$P_{2,4}$	0.0088	0.0089
$P_{2,3}$	0.0089	0.0089

6.2.2 Testing the Two-level State Estimation

Fig. 6.2 compares the errors of power flows estimated with the two-level SE and the conventional SE. When noise is the only source of error, the two-level SE seems to behave much like the conventional SE. The main difference between the two-level scheme and the conventional scheme under this purely noisy regime is the extra redundancy that the protection measurements can provide.

In the box plots, shown in Fig. 6.2, the top and the bottom of each box are the 25th and 75th percentiles of the samples, respectively, while the median is shown as the middle line.

6.2.2.1 Impact of protection measurements

The impact of the protection measurement on the accuracy of two-level SE can be investigated using Fig. 6.3, in which the standard deviation of the estimated power flow $\hat{P}_{1,2}$ has been plotted for different accuracies of the protection measurements.

When the protection measurements are as accurate as the regular measurements, i.e., with a standard deviation of $\sigma = 0.01 pu$, the standard deviation of $\hat{P}_{1,2}$ is reduced by 24 % using the two-level method. However, decreasing the protection measurement accuracy quickly reduces the advantage. The accuracy improvement of the two-level SE over the conventional SE is negligible for the standard deviation chosen for protection measurements in cases 1 and 2, $\sigma = 0.05$.

For even less accurate protection measurements, the two-level SE actually produces a slightly worse estimate than the conventional method. This probably has nothing to do with the protection measurements themselves, since their impact is almost negligible at those weights. The two-level method's approximation of neglecting the covariances of the TSO-level pseudo- mea-

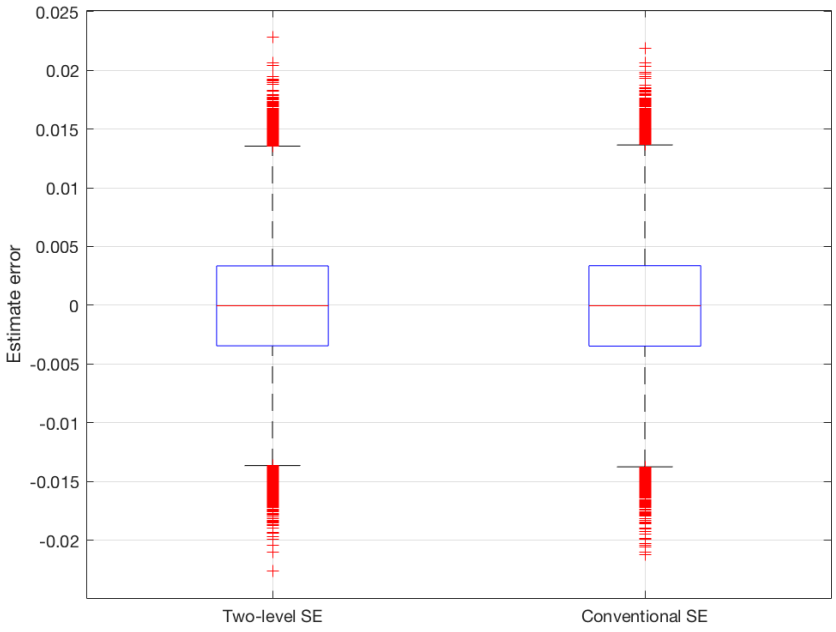


Figure 6.2: Absolute errors of estimated active power flow from bus 2 to bus 1, resulting from 100,000 runs of MCS.

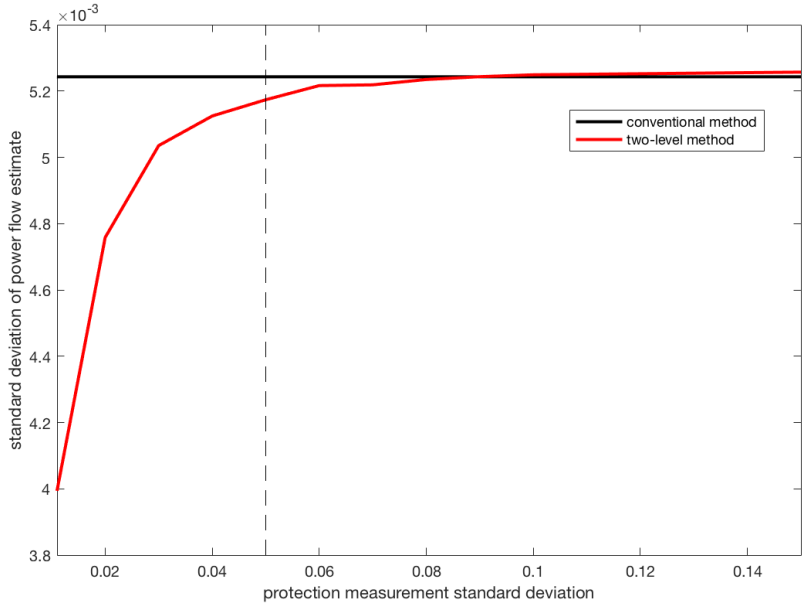


Figure 6.3: Standard deviation of the estimated $\hat{P}_{2,1}$, for different accuracies of the protection measurements, using measurement case 1 and 100,000 runs of MCS.

surements, which has been discussed already, is the likely source of this disadvantage. However, the two-level scheme is based on the concept of local redundancy. Fig. 6.3 shows that only for an unlikely scenario will the two-level SE lead to loss in accuracy.

The same testing is applied to measurement case 2, which has a lower redundancy. The result is shown in Fig. 6.4.

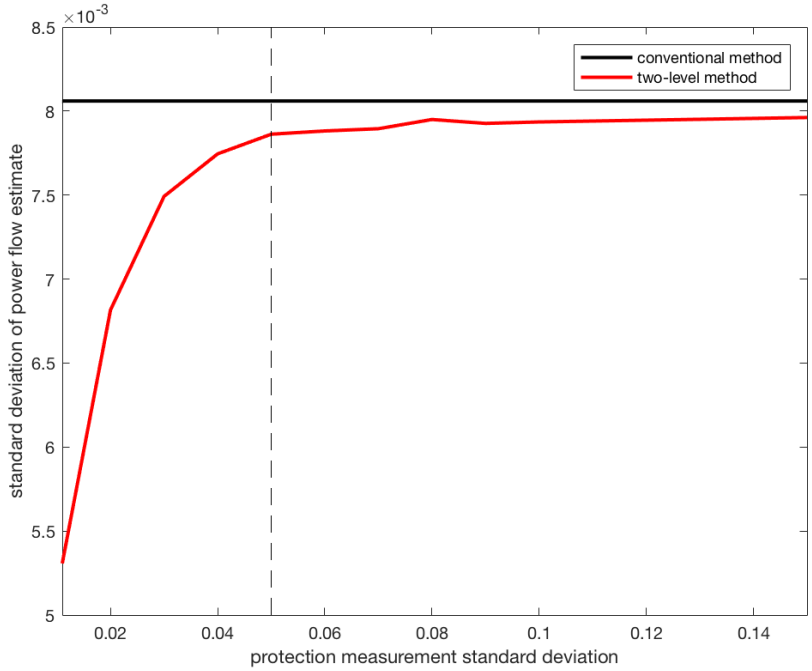


Figure 6.4: Standard deviation of the estimated $\hat{P}_{2,1}$, for different accuracies of the protection measurements, using measurement case 2 and 100,000 runs of MCS.

With lower redundancy, the protection measurements seem to have a bigger impact on accuracy. Even for very uncertain protection measurements, the two-level method gives an accuracy improvement. This is most likely mainly due to the way case 2 is defined, where some node injection measurements are not available, making the bus injection measurements of buses 1 and 4 unavailable in the conventional method, but still applicable to the node-breaker model of the SLSE.

6.3 Erroneous data conditions

6.3.1 SLSE demonstration under bad data

An execution of the two-level SE, or rather, its active sub-problem, with an arbitrary measurement set containing both noise and BD, is described in the following.

The measurement of active power injected from substation 2 into the transmission line heading to substation 1, $P_{2,1}$, is changed to introduce a BD. With the change, the error is too big to be remotely plausible within the assumed normal distribution; thus, it must be a gross error and $P_{2,1}$ is branded BD. The interesting question is whether the SLSE in substation 2 is able to correctly detect and identify the BD. Examples of the SLSE succeeding and failing are demonstrated in this section.

6.3.1.1 Noise-free measurements

First, the case where all other measurements are error-free is considered, so that the only error in the measurement set is the gross error in $P_{2,1}$. When the local SLSE in substation 2 is executed, the first estimation yields a biased estimate. The significant normalised residuals for a 15 % error and a 7.5 % error are shown in Table 6.5. The bad measurement has achieved the highest normalised residual, and would therefore be identified as BD. However, the residual of node 2 injection measurement, p_2, is almost as high, especially with the smaller error of 7.5 %. The small margin indicates that in the case of non-perfect, noisy measurements, the 7.5 % error might not always be locally identifiable; random noise can cause other measurements to achieve higher residuals, thus being erroneously identified as BD.

6.3.1.2 Successful identification

The SLSE in substation 2 is then tested with both noise and BD containing a 15 % error. The results are mostly similar to the results without noise from Table 6.5. The whole measurement set as well as the result of its execution, is shown in Table 6.6.

The normalised residual of the BD, $P_{2,1}$, is the highest for this arbitrary run, but when compared to Table 6.5, it is evident that noise has reduced the margin between the two highest residuals considerably.

Table 6.5: The normalised residuals of the first iteration (before BD is removed) of the SLSE in substation 2, when all measurements except the BD are perfect. The BD in the two considered cases contains 15 % and 7.5 % errors, respectively.

Measurement	$\boldsymbol{r_{N,1}}$, 15 % error	$\boldsymbol{r_{N,1}}$, 7.5 % error
$-p_2 = P_2$	5.96	3.20
$-p_1 = P_{2,1}$	7.12	3.49
$-p_3 = P_{2,5}$	3.90	2.45
$-p_4 = P_{2,4}$	4.77	2.67
$-p_5 = P_{2,3}$	5.08	2.92

Table 6.6: Successful execution of the substation 2 SLSE, with BD containing a 15 % error.

-	z^{ideal}	z^m	$\hat{z}_1$	$r_{N,1}$	$\hat{z}_2$	$r_{N,2}$
p_2	0.183	0.182	0.150	6.86	0.180	< 3
$P_{2,1}$	-0.940	-1.011	-1.047	6.94	0.941	-
$P_{2,5}$	0.195	0.193	0.217	4.96	0.193	< 3
$P_{2,4}$	0.318	0.323	0.345	4.74	0.320	< 3
$P_{2,3}$	0.611	0.605	0.634	6.15	0.607	< 3
$p_{1,2}$	0.940	0.999	1.047	< 3	0.941	< 3
$p_{1,3}$	0.000	-0.039	0.000	< 3	0.000	< 3
$p_{2,5}$	1.123	1.101	1.196	< 3	1.120	< 3
$p_{4,5}$	-0.512	-0.408	-0.563	3.17	-0.514	< 3
$p_{3,4}$	-0.195	-0.271	-0.217	< 3	-0.193	< 3
$p'_{1,3}$	0	0	0.000	< 3	0.000	< 3

Still, for most runs, when using the 15 % error, a successful identification is achieved. As is shown in Table 6.7, the actual BD is correctly identified in 91 % of the runs. The margin between the two biggest normalised residuals, however, varies.

Table 6.7: The outcomes of local BD processing in substation 2.

BD error	15 %	7.5 %
Correct identification of actual BD	91 %	76 %
Incorrect identification of other measurements	10 %	23 %

6.3.1.3 Failed local identification

The SLSE in substation 2 is run again, with the exact same measurement set as in Table 6.6, but now the BD is assumed to contain a reduced 7.5 % error. With this smaller error, the BD does

not stand out as clearly, making the identification harder.

Table 6.8 shows an SLSE execution of substation 2, for the same measurement set as in Table 6.6, but with a 7.5 % BD error. BD is detected, irrespective of whether the chi-squared test or the normalised residual test is used as the underlying criterion. However, the actual BD is not the measurement achieving the highest normalised residual, and so measurement P_2 is the erroneously identified as BD and removed from the measurement set. The result is a further aggravated second estimation, in which BD is no longer detected, but where the estimate is as bad or worse as the first estimate.

Table 6.8: Execution of the SLSE in substation 2, with BD.

-	z^{ideal}	z^m	$\hat{z}_1$	$r_{N,1}$	$\hat{z}_2$	$r_{N,2}$
P_2	0.183	0.182	0.165	3.69	0.104	-
$P_{2,1}$	-0.940	-1.011	-0.994	3.46	-1.011	< 3
$P_{2,5}$	0.195	0.193	0.206	< 3	0.192	< 3
$P_{2,4}$	0.318	0.323	0.333	< 3	0.319	< 3
$P_{2,3}$	0.611	0.605	0.620	< 3	0.605	< 3
$p_{1,2}$	0.940	0.999	0.994	< 3	1.011	< 3
$p_{1,3}$	0.000	-0.039	0.000	< 3	0.000	< 3
$p_{2,5}$	1.123	1.101	1.159	< 3	1.114	< 3
$p_{4,5}$	-0.512	-0.408	-0.539	< 3	-0.510	< 3
$p_{3,4}$	-0.195	-0.271	-0.206	< 3	-0.192	< 3
$p'_{1,3}$	0	0	0.000	< 3	0.000	< 3

Since measurement case 1 is utilised in this analysis, all power flows of outgoing transmission lines and all injected powers are measured. When one of these is erroneous, KCL for the substation as a whole is violated, and so the residuals increase. On studying node 1, it can be seen that the CB active power flow between nodes 1 and 2 must equal the power injected on the line from node 1. If the CB measurement was closed to its expected value, it would be able to support the correct measurement, i.e., P_2, even though protection measurements carry little weight. However, the inaccurate CB measurement unfortunately has a 6.3 % error, purely from noise, in the same direction as the BD, so it actually supports the erroneous identification. A reduction of the CB measurement error to 3.5 % would have sufficed to achieve correct BD identification.

This demonstration clearly shows how the local BD detection can fail on account of noise. It can also be inferred that CB measurement accuracy is vital for the successful identification of

BD.

In general, with the lower 7.5 % BD error, the rate of successful identification is reduced to 76 %, according to Table 6.7.

By comparing the raw measurement column (z^m) and the pre-processed measurement column ($\hat{z}$) of Table 6.8 with the value of the real load flow (z^{ideal}), it can be seen that in case of local BD failure, the pre-processing actually degrades the measurements. All the active power measurements in the substations are biased by the undiscovered BD, and also the redundancy is reduced, because of the removal of a valid measurement.

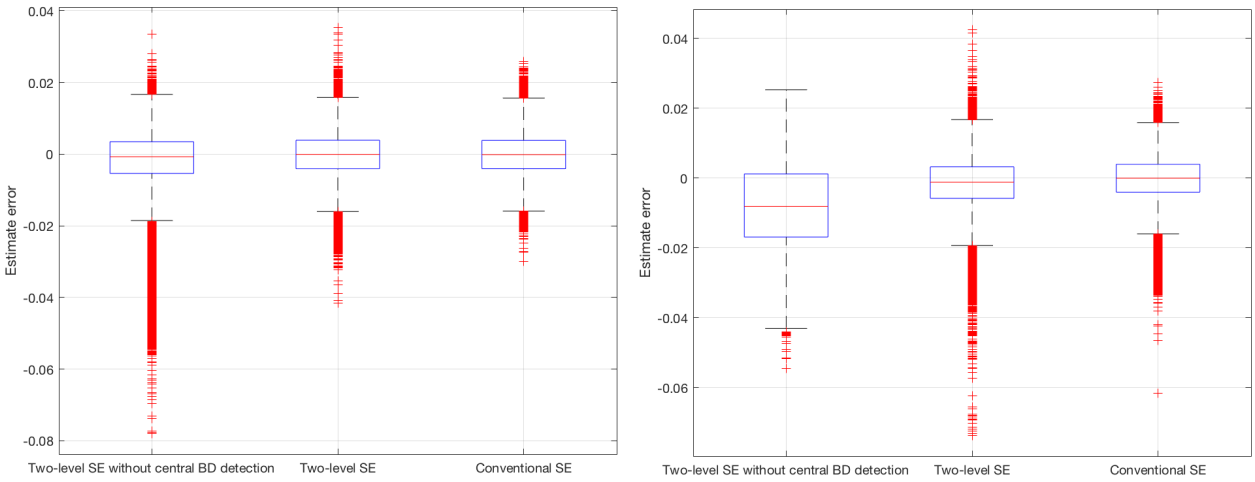
Thus, the pre-processed measurements are smeared from the BD, but the errors are for the most part contained in $P_{2,1}$ and P_2 (the new, locally estimated injection pseudo-measurement), which have a deviation of 7.1 MW and 7.9 MW, respectively. It should be noted how the correct injection measurement was eliminated and replaced by an erroneous pseudo-measurement, according to the algorithm for building the TSO-level measurement set in Section 5.4.6. In this case, BD is (or at least, can be) detected at the TSO-level, and so the bad power flow measurement is successfully identified and removed. The smeared injection measurement, however, is not detected, because of which the estimate loses accuracy on account of the smeared injection measurement.

6.3.2 Presence of a single analogue bad data

As seen in the demonstration above, BD left undetected by the SLSE leads to smearing of the TSO-level measurement set, i.e., the single erroneous measurement moves the local state vector to fit itself. This smearing makes the TSO-level BD detection challenging, since strongly correlated substation measurements appear as multiple conforming data, which are difficult to detect and correct (cf. Section 2.3). For example, as seen in the previous demonstration, the original BD is successfully removed at the TSO-level, but the smeared measurements were not. This indicates that the two-level SE scheme should mainly be based on local identification and removal of BD, leaving only noisy effects to be handled by the TSO-level. Investigating whether or not the SLSE is able to ensure that, is done in the following.

The BD containing a 15 % error is again introduced in the measurement set, with the result shown in Fig. 6.5a. Without the TSO-level BD processing, the two-level SE produces a slightly

biased estimate under these BD conditions. However, the TSO-level BD processing reduces the bias to virtually zero. The complete two-level approach does have a slight increased estimate variance, but the accuracy of the two is approximately the same.



(a) With BD containing 15 % error.

(b) With BD containing 7.5 % error.

Figure 6.5: Errors of the estimated active power flow from bus 2 to bus 1, resulting from 100,000 runs of MCS, when the original active power flow measurement is bad.

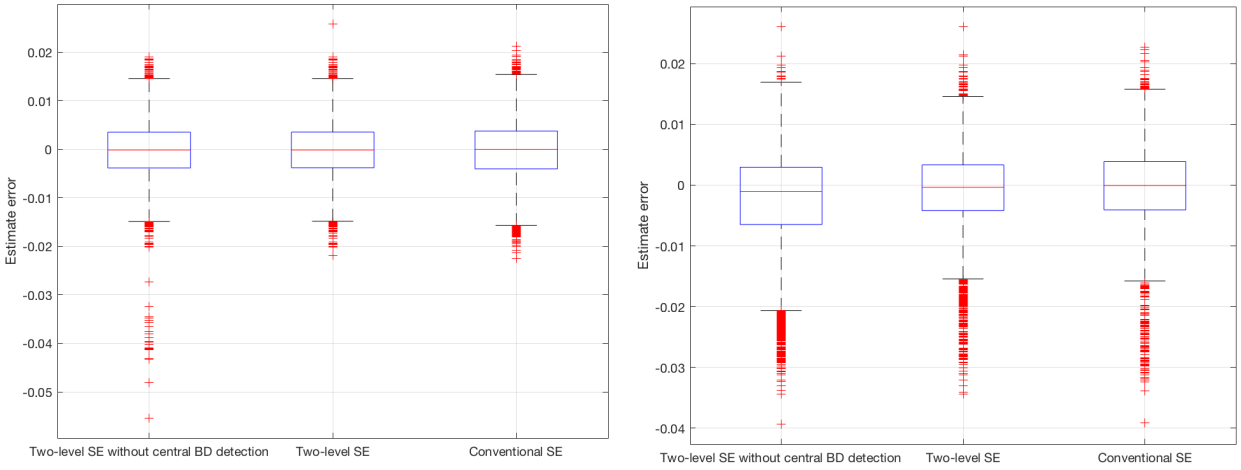
When the BD error is reduced to 7.5 %, the same effect is seen, only more extremely. Fig. 6.5b shows that the local BD processing is less effective (with a BD identification rate of 76 %, according to Table 6.7), which seems to produce a small bias, 0.19 MW, in the two-level approach.

To demonstrate again the importance of protection measurement accuracy, Fig. 6.5a and Fig. 6.5b are reproduced, but with the protection measurement standard deviation reduced to 0.025 – half that of the original value. The results, in Fig. 6.6a and Fig. 6.6b, show that with this improved measurement accuracy, the local BD processing is much more effective. With the 15 % BD error in Fig. 6.6a, the local BD processing apparently identifies 99.7 % of the BD locally.

When the error is further increased to 30 % in Fig. 6.7, the two-level SE performs better than the conventional SE, even without TSO-level BD processing.

6.3.3 Presence of bad status data

In conventional SE, topology error processing is no trivial task, which is one of the main motivations for the development of GSE (cf. Chapter 4). The SLSE should, as a scaled down GSE,



(a) With BD containing 15 % error.

(b) With BD containing 7.5 % error.

Figure 6.6: Errors of the estimated active power flow from bus 2 to bus 1, resulting from 100,000 runs of MCS, when the original active power flow measurement is bad and the protection measurement standard deviation is reduced to 0.025.

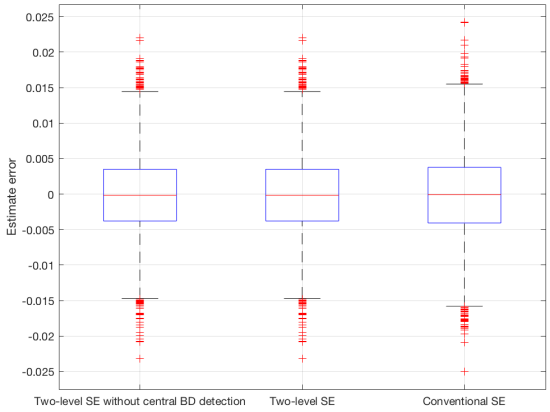
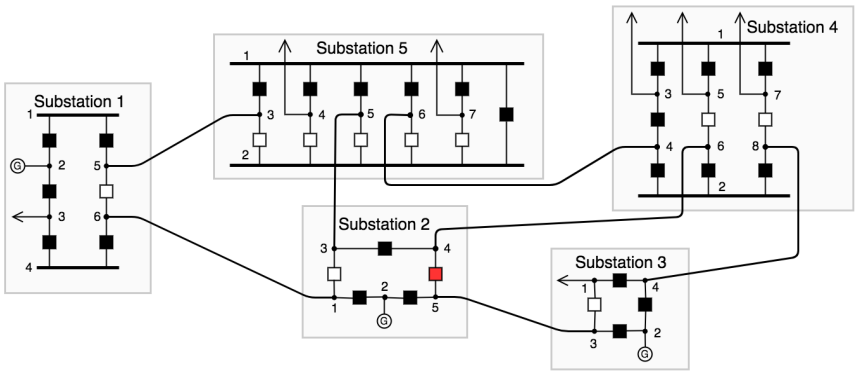


Figure 6.7: Errors of the estimated active power flow from bus 2 to bus 1, resulting from 100,000 runs of MCS, when the original active power flow measurement is bad. The protection measurement standard deviation is reduced to 0.025 and the BD error increased to 30 %.

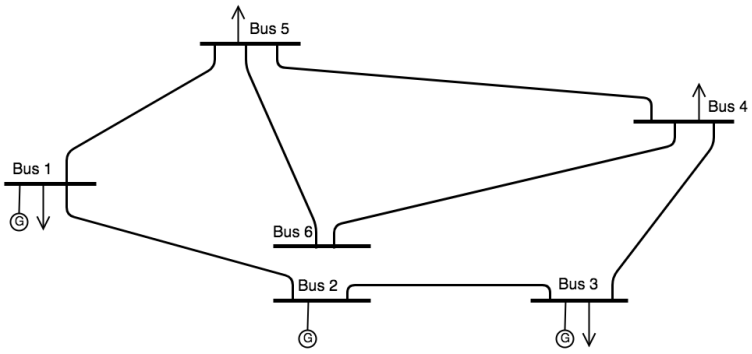
retain this same capability for topology error processing. Correct topology error identification is important for the estimation to remain unbiased, and also so that the topology in the real-time model, used by contingency analysis and other EMS functions, can be corrected.

The handling of bad status data in GSE is very similar to the handling of analogue errors. The subject is therefore treated in a brief matter in this report, in the form of a simple demonstration.

The demonstration is done for measurement case 1, but with one erroneous input: in sub-



(a) Node-breaker model with erroneous CB status (in red).



(b) Erroneous bus-branch model.

Figure 6.8: The resulting bus branch model, when CB between node 4 and 5 in substation 2 is erroneously assumed to be open. As seen, the erroneous CB status leads to a bus split error, where substation 2 is split into bus 2 and bus 6.

station 2, the CB connecting nodes 4 and 5 is reported as open, although it is actually closed. This leads to a bus-split topology error, as shown in Fig. 6.8, as the conventional topology processor splits the substation into two buses – bus 1 and bus 6.

In the subsequent execution of the conventional SE, the measurement and Jacobian equations are not correct, resulting in multiple significant residuals, as discussed in Section 4.1.1. In the two-level SE, however, the local SLSE in substation 2 can detect topology errors in the same way as BD.

The open breaker status can be thought of as a very accurate zero power flow measurement, according to (4.2) and (4.3), in the same way that a closed breaker status can be seen as a zero voltage difference measurement. A status error in a CB which, by chance, has zero power flow and zero voltage difference is therefore undetectable.

Also, the bigger the power flow is in reality, the easier the detection of CB erroneously re-

ported as open will be. Assuming the zero-flow pseudo-measurement as BD, it was shown in the previous section that the size of the BD error has a great impact on the SLSE's ability to correctly identify the BD.

In the case considered here, the real flow is 51.24 MW. Compared to the biggest BD error considered in Section 6.3, which was 14.1 MW (15 %), this error is very high. Naturally, the bad status error is easily detected locally.

6.4 Discussion and evaluation

Under purely noisy conditions, the SLSE has shown to produce pre-processed measurements with significantly lower variance than the raw-measurements. Comparing the two-level SE with the conventional SE, assuming measurement case 1, the former is shown to perform as good as the latter, but with no big improvement in accuracy. The protection measurements simply are not accurate enough, with the assumed standard deviation, to improve estimation accuracy under these conditions. However, an improvement in the accuracy of protection measurements would provide a substantial improvement in the estimate, even under noisy conditions. Also, the two-level method was shown to provide a small accuracy improvement using measurement case 2, which provides favourable conditions for the SLSE, because of limited bus-branch redundancy.

A demonstration is done for BD conditions, both for successful and failed executions of the local BD processing. It is also shown that unidentified BD in the SLSE degrades the estimate of the two-level method somewhat, but the TSO-level BD processing limits this problem.

The demonstration and the following testing revealed two factors decisive for the SLSE's ability to correctly identify the BD:

- the size of the error;
- the protection measurement accuracy.

Under satisfying conditions, e.g., with protection measurement standard deviation of 2.5 % and a BD error of at least 15 %, the local BD processing succeeds almost every time. When, additionally, the error is increased, say, to 30 %, the two level SE performs better than the conventional

SE.

A bad status measurement is also considered, but only briefly, since it follows the same logic as with BD. An erroneous open CB in a CB where the power flow is high would correspond to a big BD error, and is thus easily identifiable.

In the simulations, the local redundancy was represented by the availability of protection measurements of power flow through all the CBs. As discussed in Chapter 3, many local measurements, of voltage and current magnitudes and phase angles (locally synchronised), could be made available for the SLSE. Thus, the power flows would have to be calculated from these quantities; if just a separate CT is used for protection, together with VT for monitoring, the protection and monitoring measurements would in fact be correlated.

The assumption of the availability of CB power flows was assumed in this case to retain the original formulation of the GSE problem. If current magnitudes were used as measurements in the linear GSE, the decoupling would require that the CB current magnitudes be included in the local state vector. An alternative formulation of the SLSE, when digital substations and modern IEDs are assumed to be in place, so that current and voltage are available individually instead of only as power, is to use a voltage and current based GSE model, i.e., taking all monitoring and protection measurements in terms of their voltage and current measurements. The pseudo-measurements for the TSO-level could be kept in the conventional form by calculating power flows from the estimated currents and voltages, so that the conventional, TSO-level SE could be kept intact. The problem, however, is that the power flows would be correlated with the other measurements.

There is a great redundancy available locally, and their application has not yet been fully exploited. Large substations with multiple transformers for which the limitations of the non-linear SLSE exist, according to the discussion in Section 4.3.5, should therefore be solved using the non-linear SE, so as to provide the best possible estimate in these substations.

Chapter 7

Conclusions

7.1 Summary and concluding remarks

Modelling power systems with substation details in State Estimation (SE) has many advantages. An increased measurement redundancy is possible, because the measurements can be evaluated individually, instead of being aggregated to the bus-branch level, especially when the aggregation of all measurements is not practically feasible. The topology processor does not need to aggregate measurements and topology to the bus-branch level based on unvalidated Circuit Breaker (CB) statuses, in which the following deteriorating consequences of incorrect CB status data can be avoided: Incorrect bus-branch network model or erroneous aggregation of measurements. Also, Bad Data (BD) is never combined with healthy measurement data, which would otherwise cause the whole aggregate measurement to be bad. The high level of detail, combined with the increased redundancy, provides opportunities for improved BD processing.

Topology error processing is difficult in conventional SE, traditionally using measurement residual analysis. The effect of topology errors on the measurement residuals is chaotic, especially when noise and BD also might be present.

Generalised State Estimation (GSE) was developed to solve SE problems at the node-breaker level, by modelling every single CB in the substation. Since the reported CB status is modelled as pseudo-measurements in GSE, a topology error, resulting from errors in one or multiple CBs, can be processed as easily as BD, using residual analysis with the CBs' pseudo-measurements.

However, the GSE and the detailed modelling increases drastically both the number of mea-

surements (including zero-injection virtual measurements and CB pseudo-measurements) and the number of state variables, thus increasing the problem size and the solution time. The direct application of GSE on the full network is therefore not feasible for real-time monitoring.

Several attempts have been made to retain the advantages of the GSE while avoiding the full problem size, e.g., the two-stage method. Solving a robust SE in the first stage, i.e., an estimation not as easily influenced by errors, residual analysis is used to identify substations suspected of containing BD or bad status data; the suspected substations can then be expanded with node-breaker details in the second stage, repeating the estimation. The two-stage method is a popular in literature, but the robust estimation utilised is computationally expensive.

The Substation Level State Estimation (SLSE) is a newer application of node-breaker models and GSE, and has been the main subject of this thesis. The SLSEs are usually considered in a two-level scheme; the local estimator, the SLSE, pre-processes the measurements, sending a few, but verified, measurements to the TSO-level, where a conventional SE is executed. In SLSE methods, the problem size is naturally limited by the substation size, which is a computationally non-demanding solution, even when the detailed node-breaker model is utilised. In this way, a two-level scheme attempts to exploit the advantages of the GSE locally, especially in terms of BD and topology processing, and the accuracy improvement of the (full-network) conventional SE.

The trend of substation digitalisation provides new opportunities for SLSE. The new standard for substation-internal communication and the interoperability of Intelligent Electronic Devices (IEDs) may provide a lot of new measurements, such as those from protection devices, applicable at the node-breaker level. Restrictions in substation-external communication and in the problem size, render a centralised application of this extended measurement set improbable, so it is said to have a *local redundancy*.

The central question raised in this thesis was whether or not an SLSE, restricted in size, would be able to obtain the same advantages as the GSE, in terms of accuracy, and BD and topology error processing. *The answer is a reasonable "yes"*. Estimators are generally more accurate when applied for big areas and a high number of measurements; hence, the full-network GSE, if feasible, would provide the best estimation. However, the extended measurement set offered by the digital substation might compensate the SLSE for its limitations.

Earlier SLSE methods have been classified, reviewed and critiqued in this thesis. The most important classification criterion is the sub-system choice. Non-linear SLSE covers the complete substation, including any non-linear elements, typically transformers; thus, the standard GSE is executed as in the full-network case. The linear SLSE, however, separates the substations at the transformer terminals, i.e., every voltage level is considered as a separate substation. The linear substation thus consists only of substation nodes (bus-sections) connected together through CBs, with attached power lines, generators, loads, etc. The most important characteristic of linear SLSEs is that the electrical quantities, i.e., active power, reactive power, voltage magnitude, voltage angle, current magnitude, etc., are all decoupled. The impedance is, through Ohm's law, the connection between power flows and voltages/currents. Thus, no mathematical connection between these quantities can exist if there are only branches with zero impedance, and thus they cannot assist in the estimation of each other.

The most promising earlier SLSE approach is the non-linear GSE-based SLSE, in which all types of local measurements could be taken into account. In literature, good results were demonstrated for a big, non-linear substation, but two key questions remained unanswered. First, how would linear substations, not modelled with transformers, be handled, and how would they perform, taking the above discussion into account? Secondly, in a *nearly linear* substation, e.g., with one non-linear element, the parameters of that element would be the sole coupling factor; would that make the estimation erroneous because of the parameter errors of that element?

On that basis, a new method has been proposed: the linear, GSE-based SLSE in a two-level scheme. A linear GSE is executed locally, sending pseudo-measurements of power flow and voltage magnitudes to the TSO-level, where a conventional SE is performed. The linearity provides a non-iterative solution, avoiding issues of convergence and a constant Jacobian matrix; this makes for an efficient and reliable pre-processing.

The proposed implementation logic and design choices for the SLSE have been presented and implemented in MATLAB, including the program needed to perform the conceptual testing. Decoupled estimators have been implemented for the active power, reactive power, and voltage magnitude sub-problems; thus only measurements of these quantities are applicable. BD processing of voltages and active and reactive powers is linked by the fact that CB status error affects all the three quantities, and therefore a BD processing logic has been suggested in

this thesis.

The (pseudo-)measurements for the TSO-level SE are calculated from the local state variables. Mathematical expressions for the measurement value and the weighting variance are suggested, based on the local state variables and the state covariance matrix, respectively; and both have been validated by Monte Carlo Simulations (MCS).

Under purely noisy conditions, the SLSE is shown to reduce the variance of the measurements significantly. When BD is present, two factors were revealed as decisive for the SLSE's ability to correctly identify the BD: the size of the error and the protection measurement accuracy. If these two factors are sufficiently large, the local BD processing of the SLSE performs as well as that of the TSO-level SE. When errors are even bigger, the two-level SE even outperforms the conventional SE in accuracy, although not by much. Bad status measurements are processed in the same way as BD, and SLSE's capabilities for topology error processing were confirmed.

Local redundancy is critical for the performance of the SLSE. There is a great redundancy available locally in the substations, but no application of these have been found for a linear SLSE, except the protection measurements of power flow. The non-linear SLSE is not fitting for all substations, but could be employed in collaboration with linear SLSE. Large substations with multiple transformers could be solved using the non-linear SE, so as to exploit the redundancy and provide the best possible estimate in these important substations.

The SLSE can be the basis of visions for the future monitoring system. A linear and local SE can be solved very fast, and could therefore be applied with a much higher update rate than the TSO-level SE. With a high frequency, applying forecasting-aided techniques [23] could increase the redundancy and improve BD and bad status processing, by utilising the data of previous times-steps.

With an increased deployment of PMUs, a wide-area PMU-based SE could be implemented [22] where the SLSE result could be used to improve observability and redundancy and to verify topology. The TSO-level PMU-based SE could be run at very high rates, because of the high update frequency of PMUs. The SLSE could also be run with a higher update rate, but only updating the pseudo-measurements occasionally, to preserve communication with the super-TSO control centre. During events, such as switching or CB tripping, the SLSE could automatically validate or correct the topology locally, then reporting any changes to the TSO-level. Such a

MASE method could combine the wide-area features of the PMUs for speed and accuracy, and the local features of the SLSE for topology validation and observability.

7.2 Achievements

A literature review, shedding light on the node-breaker models and their applications in power systems, has been performed, including an overview of the power substation. Earlier approaches for SLSEs have been reviewed and critiqued, highlighting issues previously undetected and addressing a select few.

A new SLSE not described in literature has been developed, and with it a framework applicable to other SLSE implementations. The most important results are of a theoretical nature, but conceptual tests are also conducted; mainly for demonstration, but also for preliminary testing.

7.3 Recommendations for future work

Utilising the full local redundancy is vital for the SLSE concept. The proposed linear SLSE shows limitations for employing typical protection measurements. A current-based GSE formulation for SLSE should be explored, using current in the place of power flow at the local level. Also, for a real or at least a realistic power grid, the application of a hybrid two-level SE, in which large substations have a non-linear SLSE implemented, achieving a higher local redundancy in these substations, should be investigated.

Assuming that a successful SLSE can be realised, the idea of a wide-area PMU-based two-level SE could be explored as a natural extension of this thesis work. The SLSEs could provide occasional pseudo-measurements, to ensure the observability of the system and to validate the topology.

Bibliography

[1] Ali Abur and Antonio Gomez Exposito. *Power system state estimation: theory and implementation*. CRC press, 2004.

[2] John G Webster. *Electrical measurement, signal processing, and displays*. CRC Press, 2003.

[3] Antonello Monti. Lecture notes in measurement techniques and distributed intelligence for power systems, May 2016.

[4] Adl V Jaen, Pedro Cruz Romero, and Antonio Gómez Exposito. Substation data validation by a local three-phase generalized state estimator. *IEEE Transactions on Power Systems*, 20 (1):264–271, 2005.

[5] A Gómez Expósito and Antonio de la Villa Jaen. Reduced substation models for generalized state estimation. *IEEE Transactions on Power Systems*, 16(4):839–846, 2001.

[6] Fred C Schweppe. Power system static-state estimation, part iii: Implementation. *IEEE Transactions on Power Apparatus and systems*, (1):130–135, 1970.

[7] Pieter Schavemaker and Lou Van Der Sluis. *Electrical power system essentials*. John Wiley & Sons, 2008.

[8] Arun G Phadke and John Samuel Thorp. *Synchronized phasor measurements and their applications*. Springer Science & Business Media, 2008.

[9] Arun G Phadke, John S Thorp, and KJ Karimi. State estimation with phasor measurements. *IEEE Transactions on Power Systems*, 1(1):233–238, 1986.

[10] Th Van Cutsem and M Ribbens-Pavella. Critical survey of hierarchical methods for state estimation of electric power systems. *IEEE Transactions on Power Apparatus and Systems*, (10):3415–3424, 1983.

[11] Jun Zhu, Ali Abur, MJ Rice, GT Heydt, and Sakis Meliopoulos. Enhanced state estimators. Technical report, PSerc, 2006.

[12] A Gómez-Expósito. Deliverable D2.1 Part 1: Algorithms for state estimation of ETN. Technical report, PEGASE (Pan European Grid Advanced Simulations And State Estimation), 2011.

[13] Shan Zhong. *Measurement calibration/tuning & topology processing in power system state estimation.* PhD thesis, Texas A&M University, 2005.

[14] Antonio Gómez-Expósito, Antonio J Conejo, and Claudio Cañizares. *Electric energy systems: analysis and operation.* CRC Press, 2016.

[15] DM Falcao and SM De Assis. Linear programming state estimation: error analysis and gross error identification. *IEEE Transactions on Power Systems*, 3(3):809–815, 1988.

[16] Alcir Monticelli. *State estimation in electric power systems: a generalized approach*, volume 507. Springer Science & Business Media, 1999.

[17] SC Tripathy, DS Chauhan, and G Durga Prasad. State estimation algorithm for ill-conditioned power systems by Newton's method. *International Journal of Electrical Power & Energy Systems*, 9(2):113–116, 1987.

[18] Mukhtar Ahmad. *Power system state estimation.* Artech House, 2013.

[19] JS Thorp, AG Phadke, and KJ Karimi. Real time voltage-phasor measurement for static state estimation. *IEEE Transactions on Power Apparatus and Systems*, (11):3098–3106, 1985.

[20] Ming Zhou. *Advanced system monitoring with phasor measurements.* PhD thesis, Virginia Tech, 2008.

[21] S Chakrabarti, E Kyriakides, G Ledwich, and Arindam Ghosh. Inclusion of PMU current phasor measurements in a power system state estimator. *IET generation, transmission & distribution*, 4(10):1104–1115, 2010.

[22] Ali Abur. Observability and dynamic state estimation. In *2015 IEEE Power & Energy Society General Meeting*, pages 1–5. IEEE, 2015.

[23] Milton Brown Do Coutto Filho, Julio Cesar Stacchini de Souza, and Ronaldo Sérgio Freund. Forecasting-aided state estimation — Part II: Implementation. *IEEE Transactions on Power Systems*, 24(4):1678–1685, 2009.

[24] MB Do Coutto Filho, JCS de Souza, RSG Matos, and M Th Schilling. Revealing gross errors in critical measurements and sets via forecasting-aided state estimators. *Electric Power Systems Research*, 57(1):25–32, 2001.

[25] Anjan Bose and Kevin A Clements. Real-time modeling of power networks. *Proceedings of the IEEE*, 75(12):1607–1622, 1987.

[26] Felix Fulih Wu and A Monticelli. Critical review of external network modelling for online security analysis. *International Journal of Electrical Power & Energy Systems*, 5(4):222–235, 1983.

[27] A Monticelli. Electric power system state estimation. *Proceedings of the IEEE*, 88(2):262–282, 2000.

[28] Yu Tong. *Information Exchange for co-operative State Estimation*. PhD thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2009.

[29] Vassilis Kekatos and Georgios B Giannakis. Distributed robust power system state estimation. *IEEE Transactions on Power Systems*, 28(2):1617–1626, 2013.

[30] Ali Abur. Distributed state estimation for mega grids. *Proc. of the 15th PSCC Liege*, pages 22–26, 2006.

[31] Th Van Cutsem, Jean-Luc Horward, and M Ribbens-Pavella. A two-level static state estimator for electric power systems. *IEEE Transactions on Power Apparatus and Systems*, (8): 3722–3732, 1981.

[32] Antonio Gómez-Expósito, Antonio de la Villa Jaén, Catalina Gómez-Quiles, Patricia Rousseaux, and Thierry Van Cutsem. A taxonomy of multi-area state estimation methods. *Electric Power Systems Research*, 81(4):1060–1069, 2011.

[33] Antonio Gomez-Exposito, Ali Abur, Patricia Rousseaux, Antonio de la Villa Jaen, and Catalina Gomez-Quiles. On the use of PMUs in power system state estimation. In *Proc. 17th Power Systems Computation Conference*, volume 22, page 26, 2011.

[34] Liang Zhao and Ali Abur. Multi area state estimation using synchronized phasor measurements. *IEEE Transactions on Power Systems*, 20(2):611–617, 2005.

[35] MY Patel and AA Girgis. Two-level state estimation for multi-area power system. In *Power Engineering Society General Meeting, 2007. IEEE*, pages 1–6. IEEE, 2007.

[36] Antonio Gomez-Exposito, Ali Abur, Antonio de la Villa Jaen, and Catalina Gomez-Quiles. A multilevel state estimation paradigm for smart grids. *Proceedings of the IEEE*, 99(6):952–976, 2011.

[37] Jan Machowski, Janusz Bialek, and Jim Bumby. *Power system dynamics: stability and control*. John Wiley & Sons, 2011.

[38] Stanley H Horowitz and Arun G Phadke. *Power system relaying*, volume 22. John Wiley & Sons, 2008.

[39] Sasa Jakovljevic and Mladen Kezunovic. Advanced substation data collecting and processing for state estimation enhancement. In *Power Engineering Society Summer Meeting, 2002 IEEE*, volume 1, pages 201–206. IEEE, 2002.

[40] Paul M Anderson. *Power system protection*. Wiley, 1998.

[41] Leonard L Grigsby, James H Harlow, and John D McDonald. *Electric power engineering handbook*. CRC press, 2007.

[42] Sasa Jakovljevic. *Data collecting and processing for substation integration enhancement.* PhD thesis, Texas A&M University, 2004.

[43] RE Mackiewicz. Overview of iec 61850 and benefits. In *Power Systems Conference and Exposition, 2006. PSCE'06. 2006 IEEE PES*, pages 623–630. IEEE, 2006.

[44] MR Irving, RC Owen, and MJH Sterling. Power-system state estimation using linear programming. In *Proceedings of the Institution of Electrical Engineers*, volume 125, pages 879–885. IET, 1978.

[45] A Monticelli. Modeling circuit breakers in weighted least squares state estimation. *IEEE Transactions on Power Systems*, 8(3):1143–1149, 1993.

[46] Elizete Maria Lourenco, A Simões Costa, and Kevin A Clements. Bayesian-based hypothesis testing for topology error identification in generalized state estimation. *IEEE Transactions on Power Systems*, 19(2):1206–1215, 2004.

[47] Felix F Wu and W-HE Liu. Detection of topology errors by state estimation (power systems). *IEEE Transactions on Power Systems*, 4(1):176–183, 1989.

[48] Anders Gjelsvik. The significance of the lagrange multipliers in wls state estimation with equality constraints. In *Proceedings of the 11th Power Systems Computation Conference*, pages 619–625, 1993.

[49] A Abur, Hongrae Kim, and MK Celik. Identifying the unknown circuit breaker statuses in power networks. *IEEE Transactions on Power Systems*, 10(4):2029–2037, 1995.

[50] HJ Koglin, D Oeding, and KD Schmitt. Identification of topological errors in state estimation. In *2nd PSMC IEE Durham*, pages 140–144, 1986.

[51] N Singh and H Glavitsch. Detection and identification of topological errors in online power system analysis. *IEEE Transactions on Power Systems*, 6(1):324–331, 1991.

[52] Nisheeth Singh and Fritz Oesch. Practical experience with rule-based on-line topology error detection. *IEEE Transactions on Power Systems*, 9(2):841–847, 1994.

[53] JCS Souza, AM Leite da Silva, and AP Alves de Silva. Online topology determination and bad data suppression in power system operation using artificial neural networks. *IEEE Transactions on Power Systems*, 13(3):796–803, 1998.

[54] O Alsac, N Vempati, B Stott, and A Monticelli. Generalized state estimation. *IEEE Transactions on Power Systems*, 13(3):1069–1075, 1998.

[55] A de la Villa Jaén and A Gómez Expósito. Modeling unknown circuit breakers in generalized state estimators. In *Power Tech Proceedings, 2001 IEEE Porto*, volume 3, pages 4–pp. IEEE, 2001.

[56] Robert AM van Amerongen. On the exact incorporation of virtual measurements on orthogonal-transformation based state-estimation procedures. *International Journal of Electrical Power & Energy Systems*, 13(3):167–174, 1991.

[57] A Simões Costa, Elizete Maria Lourenço, and Fábio Vieira. Topology error identification for orthogonal estimators considering a priori state information. In *15th Power Systems Computation Conference*, volume 1, pages 1–6, 2005.

[58] Kevin A Clements and A Simões Costa. Topology error identification using normalized lagrange multipliers. *IEEE Transactions on Power Systems*, 13(2):347–353, 1998.

[59] Mladen Kezunovic, Papiya Dutta, Santiago Grijalva, and A Roy. The smart grid needs: Model and data interoperability and unified generalized state estimator. *PSERC Technical Report*, 2012.

[60] Antonio De La Villa Jaen and Antonio Gomez-Exposito. Implicitly constrained substation model for state estimation. *IEEE Transactions on Power Systems*, 17(3):850–856, 2002.

[61] S Zhong and A Abur. Implementation of a modified state estimator for topology error identification. In *Power Engineering Society General Meeting, 2003, IEEE*, volume 2, pages 765–770. IEEE, 2003.

[62] Mingyu Zhai, Ting Zhang, Haibin Zhang, Miaomiao Qi, Chengming Gong, and Lei Li. A new substation linear state estimation method. In *Power & Energy Society General Meeting, 2015 IEEE*, pages 1–5. IEEE, 2015.

[63] Tao Yang, Hongbin Sun, and Anjan Bose. Transition to a two-level linear state estimator—part i: Architecture. *IEEE Transactions on Power Systems*, 26(1):46–53, 2011.

[64] MR Irving and MJH Sterling. Substation data validation. In *IEE Proceedings C (Generation, Transmission and Distribution)*, volume 129, pages 119–122. IET, 1982.

[65] Antonio Gomez-Exposito and Antonio de la Villa Jaen. Two-level state estimation with local measurement pre-processing. *IEEE Transactions on Power Systems*, 24(2):676–684, 2009.

[66] Wikipedia. Weighted arithmetic mean: Accounting for correlations, 2016. URL `http://en.wikipedia.org/wiki/Weighted_arithmetic_mean`. Accessed 04.06.2017.

[67] Ronald E Walpole, Raymond H Myers, Sharon L Myers, and Keying Ye. *Probability and statistics for engineers and scientists*, volume 5. Macmillan New York, 1993.

[68] P G Praviraj. Power System State Estimation using WLS, 2009. URL `http://se.mathworks.com/matlabcentral/fileexchange/23052-power-system-state-estimation-using-wls`. Accessed 19.01.2017.

Appendix A

The Complete WLS Solution Methodology for SSE

The WLS method is by far the most deployed technique for SE. This appendix presents a thorough, mathematical presentation of SSE, as background material for the example of Section 2.2.

A.1 Step-by-step solution

A solution manual for WLS SSE, based on the method from [1], follows:

1. Start iterations, $k = 0$, and initialise $\boldsymbol{x_k}$ with all bus voltages at flat start $V_i = 1pu$, $\theta_i = 0$.

2. Calculate the measurement function, $\boldsymbol{h_k}$, and measurement Jacobian, $\boldsymbol{H_k}$, using the latest state estimate, $\boldsymbol{x_k}$.

3. Solve the NE (2.7),

   $\Delta \boldsymbol{x_k} = [\boldsymbol{H}_k^T \boldsymbol{W} \boldsymbol{H}_k]^{-1} \boldsymbol{H}_k^T \boldsymbol{W} [\boldsymbol{z} - \boldsymbol{h_k}]$ (either by direct inversion of the gain or some sort of decomposition), to obtain the state update $\Delta \boldsymbol{x_k}$.

4. Test for convergence, e.g., that the biggest update voltage magnitude and angle is less than the tolerance limits: $max(|\Delta \boldsymbol{\theta_i}|) < \epsilon_\theta$ and $max(|\Delta \boldsymbol{V_i}|) < \epsilon_V$.

5. If no, update $\boldsymbol{x_{k+1}} = \boldsymbol{x_k} + \Delta \boldsymbol{x_k}$ and $k = k + 1$, and go to step 2. Else, stop, with $\hat{\boldsymbol{x}} = \boldsymbol{x_k}$.

A.2 Power system model and measurement equations

The power system is modelled using the bus admittance ($\boldsymbol{Y}_{bus}$) representation known from LF studies, with the equations of bus power injections. The rectangular form of $\boldsymbol{Y}_{bus}$ is used here, where $\boldsymbol{G}$ is the conductance and $\boldsymbol{B}$ is the susceptance. It should be noted that V_i is used as shortened notation for $|V_i|$, for simplicity, and likewise, I_{ij} is used for $\left|I_{ij}\right|$.

The measurement functions for injected real and reactive power, $\boldsymbol{P_{inj}}$ and $\boldsymbol{Q_{inj}}$, are the same ones used in LF studies (where $\theta_{ij} = \theta_i - \theta_j$, the voltage angle difference between two buses):

$$P_i = V_i \sum_{j=1}^{N} V_j [G_{ij} cos(\theta_{ij}) + B_{ij} sin(\theta_{ij})] \tag{A.1}$$

$$Q_i = V_i \sum_{j=1}^{N} V_j [G_{ij} sin(\theta_{ij}) - B_{ij} cos(\theta_{ij})] \tag{A.2}$$

In addition, there are measurements of branch power flows. Therefore, the branch impedances (also called primitive impedances), from which the $\boldsymbol{Y}_{bus}$ is constructed, have to be known. For a line, using the pi-model (cf. Fig. A.1), the equations are

$$P_{ij} = V_i^2 (g_{si} + g_{ij}) - V_i V_j (g_{ij} cos\theta_{ij} + b_{ij} sin\theta_{ij}) \tag{A.3}$$

$$Q_{ij} = -V_i^2 (b_{si} + b_{ij}) - V_i V_j (g_{ij} sin\theta_{ij} - b_{ij} cos\theta_{ij}) \tag{A.4}$$

The measurement function for voltage magnitude is simply:

$$V_i = V_i \tag{A.5}$$

The current magnitude measurement function, is more complicated, given as

$$I_{ij} = \frac{\sqrt{P_{ij}^2 + Q_{ij}^2}}{V_i} \tag{A.6}$$

where P_{ij} and Q_{ij} is given from (A.3) and (A.4).

A.3 Measurement function vector and its Jacobian

The measurement function vector, $\boldsymbol{h}(\boldsymbol{x})$, has the following structure:

$$\boldsymbol{h}(\boldsymbol{x}) = \begin{bmatrix} \\ V_1(\boldsymbol{x}) \\ \\ \vdots \\ \\ P_1(\boldsymbol{x}) \\ \\ \vdots \\ \\ \vdots \\ \\ P_{ij}(\boldsymbol{x}) \\ \\ Q_1(\boldsymbol{x}) \\ \\ \vdots \\ \\ \vdots \\ \\ Q_{ij}(\boldsymbol{x}) \\ \\ \vdots \\ \\ \end{bmatrix} \tag{A.7}$$

All the measurement functions are described in the above review. The Jacobian, $\boldsymbol{H}(\boldsymbol{x})$, is derived by differentiation of the given functions in $\boldsymbol{h}(\boldsymbol{x})$, with respect to $\boldsymbol{x}$:

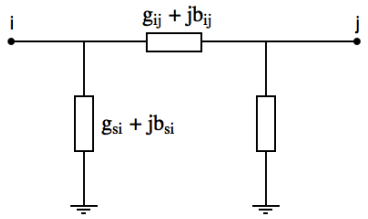


Figure A.1: Pi-model of a transmission line

$$H = \begin{bmatrix} \frac{\partial P_{inj}}{\partial \theta} & \frac{\partial P_{inj}}{\partial V} \\ \frac{\partial P_{flow}}{\partial \theta} & \frac{\partial P_{flow}}{\partial V} \\ \frac{\partial Q_{inj}}{\partial \theta} & \frac{\partial Q_{inj}}{\partial V} \\ \frac{\partial Q_{flow}}{\partial \theta} & \frac{\partial Q_{flow}}{\partial V} \\ \frac{\partial I_{mag}}{\partial \theta} & \frac{\partial I_{mag}}{\partial V} \\ \mathbf{0} & \frac{\partial V_{mag}}{\partial V} \end{bmatrix} \tag{A.8}$$

The elements in these equations are themselves matrices. The matrix $\frac{\partial P_{inj}}{\partial \theta}$ will have one row per measurement of active power injection P_i in the measurement set. For real power injections, the element matrix is

$$\frac{\partial P_{inj}}{\partial \theta} = \begin{bmatrix} \frac{\partial P_1}{\partial \theta_2} & \cdots & \frac{\partial P_1}{\partial \theta_N} \\ \vdots & & \vdots \\ \frac{\partial P_N}{\partial \theta_2} & \cdots & \frac{\partial P_N}{\partial \theta_N} \end{bmatrix} \tag{A.9}$$

All the other matrices in (A.8) are developed in a similar fashion.

The elements in these sub-matrices of $\boldsymbol{H}$ are given by differentiation of the measurement functions with respect to the state variables. For completeness, they are given in the following, from [1].

Active power injections:

$$\frac{\partial P_i}{\partial \theta_i} = V_i \sum_{j=1}^{N} V_j[-G_{ij}sin(\theta_{ij}) + B_{ij}cos(\theta_{ij})] - V_i^2 B_{ii} \tag{A.10}$$

$$\frac{\partial P_i}{\partial \theta_j} = V_i V_j[G_{ij}sin(\theta_{ij}) - B_{ij}cos(\theta_{ij})] \tag{A.11}$$

$$\frac{\partial P_i}{\partial V_i} = \sum_{j=1}^{N} V_j[G_{ij}cos(\theta_{ij}) + B_{ij}sin(\theta_{ij})] + V_i G_{ii} \tag{A.12}$$

$$\frac{\partial P_i}{\partial V_j} = V_i[G_{ij}cos(\theta_{ij}) + B_{ij}sin(\theta_{ij})] \tag{A.13}$$

Reactive power injections:

$$\frac{\partial Q_i}{\partial \theta_i} = V_i \sum_{j=1}^{N} V_j[G_{ij}cos(\theta_{ij}) + B_{ij}sin(\theta_{ij})] - V_i^2 G_{ii} \tag{A.14}$$

$$\frac{\partial Q_i}{\partial \theta_j} = V_i V_j[-G_{ij}cos(\theta_{ij}) - B_{ij}sin(\theta_{ij})] \tag{A.15}$$

$$\frac{\partial Q_i}{\partial V_i} = \sum_{j=1}^{N} V_j[G_{ij}sin(\theta_{ij}) - B_{ij}cos(\theta_{ij})] - V_i B_{ii} \tag{A.16}$$

$$\frac{\partial Q_i}{\partial V_j} = V_i[G_{ij}sin(\theta_{ij}) - B_{ij}cos(\theta_{ij})] \tag{A.17}$$

Active power flows:

$$\frac{\partial P_{ij}}{\partial \theta_i} = V_i V_j[g_{ij}sin(\theta_{ij}) - b_{ij}cos(\theta_{ij})] \tag{A.18}$$

$$\frac{\partial P_{ij}}{\partial \theta_j} = -V_i V_j[g_{ij}sin(\theta_{ij}) - b_{ij}cos(\theta_{ij})] \tag{A.19}$$

$$\frac{\partial P_{ij}}{\partial V_i} = -V_j[g_{ij}cos(\theta_{ij}) + b_{ij}sin(\theta_{ij})] + 2(g_{ij} + g_{si})V_i \tag{A.20}$$

$$\frac{\partial P_{ij}}{\partial V_j} = -V_i[g_{ij}cos(\theta_{ij}) + b_{ij}sin(\theta_{ij})] \tag{A.21}$$

Reactive power flows:

$$\frac{\partial Q_{ij}}{\partial \theta_i} = -V_i V_j [g_{ij} cos(\theta_{ij}) + b_{ij} sin(\theta_{ij})] \tag{A.22}$$

$$\frac{\partial Q_{ij}}{\partial \theta_j} = V_i V_j [g_{ij} cos(\theta_{ij}) + b_{ij} sin(\theta_{ij})] \tag{A.23}$$

$$\frac{\partial Q_{ij}}{\partial V_i} = -V_j [g_{ij} sin(\theta_{ij}) - b_{ij} cos(\theta_{ij})] - 2(b_{ij} + b_{si}) V_i \tag{A.24}$$

$$\frac{\partial Q_{ij}}{\partial V_j} = -V_i [g_{ij} sin(\theta_{ij}) - b_{ij} cos(\theta_{ij})] \tag{A.25}$$

Branch current magnitudes, when ignoring the shunt admittance:

$$\frac{\partial Q_i}{\partial \theta_i} = \frac{g_{ij}^2 + b_{ij}^2}{I_{ij}} V_i V_j sin(\theta_{ij}) \tag{A.26}$$

$$\frac{\partial Q_i}{\partial \theta_j} = -\frac{g_{ij}^2 + b_{ij}^2}{I_{ij}} V_i V_j sin(\theta_{ij}) \tag{A.27}$$

$$\frac{\partial Q_i}{\partial V_i} = \frac{g_{ij}^2 + b_{ij}^2}{I_{ij}} [V_i - V_j cos(\theta_{ij})] \tag{A.28}$$

$$\frac{\partial Q_i}{\partial V_j} = \frac{g_{ij}^2 + b_{ij}^2}{I_{ij}} [V_j - V_i cos[\theta_{ij})] \tag{A.29}$$

Bus voltage magnitudes:

$$\frac{\partial V_i}{\partial \theta_i} = 0 \tag{A.30}$$

$$\frac{\partial V_i}{\partial \theta_j} = 0 \tag{A.31}$$

$$\frac{\partial V_i}{\partial V_i} = 1 \tag{A.32}$$

$$\frac{\partial V_i}{\partial V_j} = 0 \tag{A.33}$$

Appendix B

MATLAB code

The MATLAB code for execution of the two-level SE and the conventional SE are given in the Appendix.

B.1 measurementGenerator.m

The *measurementGenerator.m* executes a load flow to generate the measurements that are the inputs to the SE. It takes for itself the Excel files given in Appendix C as input. The measurements are saved in a file accessed by the *central.m* script, so the *measurementGenerator.m* script must only be run once for every time the input of the Excel files change. Matpower must included in MATLAB or be kept in the same folder as this file. Matpower is available online (*http://www.pserc.cornell.edu/matpower/*).

```
% This script collects the inputs from excel files, executes a load, and
% prepares the measurements on the node-breaker level. It also provides
% linedata and true state. The output is stored in a file, so that this only
% has to be run every time the inputs change.

clear; clc;

%% Setting testing parameters and data source
toperror = false; %if true, a bus split topology error is introduced in substation 2.
measurementdatabase = 'database/meas_database_case1.xlsx';

```

```
12
13 %% Set measurement parameters
14 rii_zeroinj = 0.00000001^2; %10^-20;
15 rii_CBvolt = 0.0000001^2; %10^-20;
16 rii_CBangl = 0.0000001^2; %10^-20;
17 rii_V = 0.005^2;
18 rii_Pinj = 0.01^2;
19 rii_Qinj = 0.01^2;
20 rii_P = 0.01^2;
21 rii_Q = 0.01^2;
22 rii_CB_P = 0.0000001^2; %10^-20;
23 rii_CB_Q = 0.0000001^2; %10^-20;
24
25 MVAbase = 100;
26
27 %% Preparing cd.branch (Matpower input)
28 cd.version = '2';
29 cd.baseMVA = MVAbase;
30
31 data.lines = xlsread('database/lines_database.xlsx');
32
33 n_sub = max(max(data.lines(:,1:2)));
34 a = size(data.lines,1);
35 cd.branch = [data.lines(:,3:4), data.lines(:,5:7), zeros(a, 5), ones(a,1), -360*ones(a,1)
       , 360*ones(a,1)];
36
37 n_nodes = zeros(n_sub,1);
38 nodenum = zeros(n_sub,1);
39 i_nodes = cell(n_sub,1);
40 n=1;
41 for s = 1:n_sub
42     nodenum(s) = sum(n_nodes);
43     n_nodes(s) = max(max(data.lines(data.lines(:,1)==s & data.lines(:,2)==s,3:4)));
44     i_nodes{s} = (1+nodenum(s)):(nodenum(s)+n_nodes(s));
45 end
46
```

```
cd.branch(:,1) = nodenum(data.lines(:,1))+data.lines(:,3);
cd.branch(:,2) = nodenum(data.lines(:,2))+data.lines(:,4);

data.CB = xlsread(measurementdatabase, 'OpenCBs');
b = size(data.CB,1);
openCBs = zeros(b,2);
openCBs(:,1) = nodenum(data.CB(:,1))+data.CB(:,2);
openCBs(:,2) = nodenum(data.CB(:,1))+data.CB(:,3);

for i = 1:b
    cd.branch((cd.branch(:,1) == openCBs(i,1) & cd.branch(:,2) == openCBs(i,2)) | ...
        (cd.branch(:,1) == openCBs(i,2) & cd.branch(:,2) == openCBs(i,1)) ,4) = 10000;
end
cd.branch(cd.branch(:,4)==0, 4) = 0.00001;

%% Preparing cd.bus and cd.gen (Matpower input)
ntot_nodes = sum(n_nodes);
cd.bus = [zeros(ntot_nodes,6), ones(ntot_nodes, 1), ones(ntot_nodes,1), zeros(ntot_nodes,
    2), ones(ntot_nodes,1), ...
    1.06*ones(ntot_nodes, 1), 0.94*ones(ntot_nodes, 1)];
cd.gen = [];

for s = 1:n_sub
    data.operation{s} = xlsread('database/operation_database.xlsx', num2str(s));
    nodetype{s} = data.operation{s}(:,1:2);

    for i = 1:n_nodes(s)
        n = i_nodes{s}(i);
        cd.bus(n,1) = n;
        cd.bus(n,2) = data.operation{s}(i,2);

        if data.operation{s}(i,2) == 1 %load bus
            cd.bus(n,3:4) = -data.operation{s}(i,3:4);

        elseif data.operation{s}(i,2) == 4 || data.operation{s}(i,2) == 5 %zero-injection
    bus
```

```
81              cd.bus(n,2) = 1;
82              cd.bus(n,3:4) = [0,0];
83
84          elseif data.operation{s}(i,2) == 2 %generator bus
85              gen = [n, data.operation{s}(i,3), 0, 1000, -200, ...
86                  data.operation{s}(i,5), 0, 1, 1000, 0, zeros(1,11)];
87              cd.gen = [cd.gen; gen];
88
89          elseif data.operation{s}(i,2) == 3 %slack bus
90              gen = [n, 0, 0, 1000, -200, ...
91                  data.operation{s}(i,5), 0, 1, 1000, 0, zeros(1,11)];
92              cd.gen = [cd.gen; gen];
93              cd.bus(n,8) = data.operation{s}(i,5);
94          end
95      end
96  end
97
98  pfres = runpf(cd); %runs Matpower load flow
99
100 %% Preparing the measurement set
101 data.meas = cell(n_sub,1);
102 z = cell(n_sub,1);
103 for s=1:n_sub
104
105     %Find conventional measurements
106     data.meas{s} = xlsread(measurementdatabase, num2str(s));
107
108     [n_meas, temp] = size(data.meas{s});
109     data.meas{s} = [data.meas{s}, nan(n_meas, 5-temp)];
110
111     %Find zero-injection nodes
112     zeroinj_nodes = data.operation{s}(data.operation{s}(:,2) == 4,1);
113     n_zeroinj = length(zeroinj_nodes);
114
115     %Find CB signals
116     lines = data.lines(data.lines(:,1) == s & data.lines(:,2) == s,3:4);
```

```
117     openCBs = data.CB(data.CB(:,1)==s,2:3);
118     n_openCBs = size(openCBs,1);
119     n_closedCBs = size(lines,1)-n_openCBs;
120     closedCBs = zeros(n_closedCBs,2);
121     count = 1;
122     for i = 1:size(lines,1)
123         if max(sum((lines(i,:) == openCBs),2)) ~=2
124             closedCBs(count,:) = lines(i,:);
125             count = count + 1;
126         end
127     end
128 
129     if s == 2 && toperror
130         openCBs = [openCBs; 4,5];
131         n_openCBs = n_openCBs + 1;
132         closedCBs(4,:)=[];
133         n_closedCBs = n_closedCBs - 1;
134     end
135 
136     %Pre-allocating measurement matrix
137     z{s} = zeros(n_meas, 5);
138 
139     %Conventional
140     for m = 1:n_meas
141         meas = data.meas{s}(m,:);
142         node = i_nodes{s}(meas(1));
143         if meas(3) == 1
144             z{s}(m,:) = [meas(1),  0,  1,  pfres.bus(pfres.bus(:,1)==node, 8),  rii_V];
145         elseif meas(3) == 2
146             if cd.bus(node,2) == 1
147                 z{s}(m,:) = [meas(1),  0,  2,  -pfres.bus(pfres.bus(:,1)==node, 3)/
    MVAbase,  rii_Pinj];
148             else
149                 z{s}(m,:) = [meas(1),  0,  2,  pfres.gen(pfres.gen(:,1)==node, 2)/
    MVAbase,  rii_Pinj];
150             end
```

```
151         elseif meas(3) == 3
152             if cd.bus(node,2) == 1
153                 z{s}(m,:) = [meas(1), 0, 3, -pfres.bus(pfres.bus(:,1)==node, 4)/
    MVAbase, rii_Qinj];
154             else
155                 z{s}(m,:) = [meas(1), 0, 3, pfres.gen(pfres.gen(:,1)==node, 3)/
    MVAbase, rii_Qinj];
156             end
157         elseif meas(3) == 4
158             if isnan(meas(2)) || meas(2) == 0 %if node injection is from incoming line
159                 index1 = find(data.lines(:,1) == s & data.lines(:,2)~=s & data.lines(:,3)
     == meas(1));
160                 index2 = find(data.lines(:,2) == s & data.lines(:,1)~=s & data.lines(:,4)
     == meas(1));
161                 if isempty(index2)
162                     t_node = i_nodes{data.lines(index1,2)}(data.lines(index1,4));
163                 else
164                     t_node = i_nodes{data.lines(index2,1)}(data.lines(index2,3));
165                 end
166                 if sum(pfres.branch(:,1)==node & pfres.branch(:,2)==t_node)==0
167                     z{s}(m,:) = [meas(1), 0, 4, -pfres.branch(pfres.branch(:,1)==t_node
     & pfres.branch(:,2)==node, 16)/MVAbase, rii_P];
168                 else
169                     z{s}(m,:) = [meas(1), 0, 4, -pfres.branch(pfres.branch(:,1)==node
    & pfres.branch(:,2)==t_node, 14)/MVAbase, rii_P];
170                 end
171             else
172                 t_node = i_nodes{s}(meas(2));
173                 if sum(pfres.branch(:,1) == t_node & pfres.branch(:,2)==node) == 0
174                     z{s}(m,:) = [meas(1), meas(2), 4, pfres.branch(pfres.branch(:,1)
    ==node & pfres.branch(:,2)==t_node, 14)/MVAbase, rii_P];
175                 else
176                     z{s}(m,:) = [meas(1), meas(2), 4, pfres.branch(pfres.branch(:,1)
    == t_node & pfres.branch(:,2)==node, 16)/MVAbase, rii_P];
177                 end
178
```

```
179             end
180         elseif meas(3) == 5
181             if isnan(meas(2)) || meas(2) == 0 %if node injection is from incoming line
182                 index1 = find(data.lines(:,1) == s & data.lines(:,2)~=s & data.lines(:,3)
    == meas(1));
183                 index2 = find(data.lines(:,2) == s & data.lines(:,1)~=s & data.lines(:,4)
    == meas(1));
184                 if isempty(index2)
185                     t_node = i_nodes{data.lines(index1,2)}(data.lines(index1,4));
186                 else
187                     t_node = i_nodes{data.lines(index2,1)}(data.lines(index2,3));
188                 end
189                 if sum(pfres.branch(:,1)==node & pfres.branch(:,2)==t_node)==0
190                     z{s}(m,:) = [meas(1),  0,  5, -pfres.branch(pfres.branch(:,1)==t_node
    & pfres.branch(:,2)==node, 17)/MVAbase,    rii_Q];
191                 else
192                     z{s}(m,:) = [meas(1),  0,  5,  -pfres.branch(pfres.branch(:,1)==node
    & pfres.branch(:,2)==t_node, 15)/MVAbase,    rii_Q];
193                 end
194             else
195                 t_node = i_nodes{s}(meas(2));
196                 if sum(pfres.branch(:,1) == t_node & pfres.branch(:,2)==node) == 0
197                     z{s}(m,:) = [meas(1),  meas(2),  5,   pfres.branch(pfres.branch(:,1)
    ==node & pfres.branch(:,2)==t_node, 15)/MVAbase,    rii_Q];
198                 else
199                     z{s}(m,:) = [meas(1),  meas(2),  5,   pfres.branch(pfres.branch(:,1)
    == t_node & pfres.branch(:,2)==node, 17)/MVAbase,    rii_Q];
200                 end
201
202             end
203         end
204         if ~isnan(meas(4)) %if a value is given manually in the excel file
205           z{s}(m,4) = meas(4);
206         end
207         if ~isnan(meas(5)) %if a variance is given manually in the excel file
208           z{s}(m,5) = meas(5)^2;
```

```
209         end
210     end
211
212     %Preparing zero-injection virtual measurements
213     z{s} = [z{s};
214         zeroinj_nodes, zeros(n_zeroinj,1), 2*ones(n_zeroinj,1), ...
215         zeros(n_zeroinj,1), rii_zeroinj*ones(n_zeroinj,1)];
216     z{s} = [z{s};
217         zeroinj_nodes, zeros(n_zeroinj,1), 3*ones(n_zeroinj,1), ...
218         zeros(n_zeroinj,1), rii_zeroinj*ones(n_zeroinj,1)];
219
220     %Preparing closed CB virtual measurements
221     z{s} = [z{s};
222         closedCBs(:,1), closedCBs(:,2), 10*ones(n_closedCBs,1), ...
223         zeros(n_closedCBs,1), rii_CBangl*ones(n_closedCBs,1)];
224     z{s} = [z{s};
225         closedCBs(:,1), closedCBs(:,2), 11*ones(n_closedCBs,1), ...
226         zeros(n_closedCBs,1), rii_CBvolt*ones(n_closedCBs,1)];
227
228     %Preparing open CB virtual measurements
229     z{s} = [z{s};
230         openCBs(:,1), openCBs(:,2), 12*ones(n_openCBs,1), ...
231         zeros(n_openCBs,1), rii_CB_P*ones(n_openCBs,1)];
232     z{s} = [z{s};
233         openCBs(:,1), openCBs(:,2), 13*ones(n_openCBs,1), ...
234         zeros(n_openCBs,1), rii_CB_Q*ones(n_openCBs,1)];
235 end
236
237
238 %% Prepare linedata
239 linedata = data.lines;
240 Vreal = pfres.bus(nodenum+1,8);
241 delreal = pfres.bus(nodenum+1,9);
242
243 %% Save linedata
244 save('busbranchmeasurements', 'linedata', 'z', 'Vreal', 'delreal', 'nodetype');
```

```
245 save('realestimate', 'Vreal', 'delreal')
```

B.2 central.m

The *central.m* script executes a two-level SE and a conventional SE based on the same measurement set. The measurements are automatically collected from the file produced by *measurementGenerator.m*. The results are printed and plotted if the variable *display* is declared true.

```
1  % This script executes a two-level SE and a conventional SE based on the
2  % same measureent set. The measurements are automatically collected from
3  % the file produced by measurementGenerator.m. The results are printed and
4  % plotted if the variable display is set to true.
5
6  %% Input data
7  load('busbranchmeasurements'); %input file
8  display = false; %if true, results will be printed and plotted
9
10 %% Introducing measurement noise
11 for s =1:n_sub
12     z{s}(z{s}(:,5) > 0.0001^2, 4) = normrnd(z{s}(z{s}(:,5) > 0.0001^2, 4), sqrt(z{s}(z{s
       }(:,5) > 0.0001^2, 5)));
13 end
14
15 %% Executing local SE
16
17 %Classifying lines:
18 n_sub = size(z,1);
19 i_extbranches = find(linedata(:,1) ~= linedata(:,2)); %substation-external branches
20 zcentral = cell(n_sub,1); %pre-processed measurement set for the central level
21 i_intbranches = cell(1,n_sub); %subsation-internal branches (zero-impedance)
22 extlinedata = cell(1,n_sub); %array of external branches connected to each substation
23
24 if display
25     disp('Executing two-level SE')
26 end
27
```

```
for s = 1:n_sub
    extlinedata{s} = linedata(linedata(:,1) ~= linedata(:,2) & (linedata(:,1) == s |
    linedata(:,2) == s ),:);
    localbranches = linedata((linedata(:,1) == s & linedata(:,2) == s),3:4);
    zlocal = z{s};

    [zcentral{s}, H] = local(zlocal, localbranches, extlinedata{s}, nodetype{s}, s,
    display);
end

%% Execute central SE and BD processing
[ybus, bsh, g, b] = getGridParameters2(linedata(i_extbranches, [1,2,5,6,7]));

zdata = cat(1, zcentral{[1,2,3,4,5]});
zdata = zdata(~isnan(zdata(:,4)),:);
zdata = sortrows(zdata, [3,1,2]);

%Executing first SE:
[stateestimate, h, r, r_N, J, x] = wlsfunction(zdata, ybus, bsh, g, b);

Del0=stateestimate(:,3); %first estimate
V0=stateestimate(:,2); %first estimate

[r_N_max, i] = max(r_N);

%Bad data processing and rerunning of the SE:
while r_N_max > 3 && J>chi2inv(0.99,size(zdata,1)-n_sub*2+1)
    if display
        disp('Removed bad data in central level SE:')
        disp(zdata(i,:));
    end
    zdata(i,:)=[];
    [stateestimate, h, r, r_N, J, x] = wlsfunction(zdata, ybus, bsh, g, b, x);
    [r_N_max, i] = max(r_N);
end

```

```
%Final two-level estimate:
Del=stateestimate(:,3);
V=stateestimate(:,2);

%% Prepare conventional SE
z2 = z; %node-breaker measurement set for the conventional SE
zdata2 = []; %bus-branch measurement set for the conventional SE

for s = 1:n_sub

    %Removing extra redundancy from protection measurements and pseudo-meaurements;:
    z2{s}(z2{s}(:,5)>0.010001^2 | z2{s}(:,5)<0.000000001 , :) = [];

    %Adding voltage measurement:
    zdata2 = [zdata2; s, 0, 1, mean(z2{s}(z2{s}(:,3)==1, 4)), mean(z2{s}(z2{s}(:,3)==1,
    5))/length(z2{s}(z2{s}(:,3)==1, 5))];

    %Adding power injections:
    i_inj = find(nodetype{s}(:,2)== 1 |  nodetype{s}(:,2)== 2 | nodetype{s}(:,2)== 3);
    if isempty(i_inj)
        zdata2 = [zdata2;   s, 0, 2, 0, 10^-10;   s, 0, 3, 0, 10^-10];
    end
    if isequal(i_inj, sort(z2{s}(z2{s}(:,3) == 2,1)))
        zdata2 = [zdata2; s, 0, 2, sum(z2{s}(z2{s}(:,3) == 2, 4)), sum(z2{s}(z2{s}(:,3)
    == 2, 5))];
    end
    if isequal(i_inj, sort(z2{s}(z2{s}(:,3) == 3,1)))
        zdata2 = [zdata2; s, 0, 3, sum(z2{s}(z2{s}(:,3) == 3, 4)), sum(z2{s}(z2{s}(:,3)
    == 3, 5))];
    end

    %Adding power flows:
    i_flow = find(nodetype{s}(:,2)== 5); %indicates bay nodes
    for i = 1:length(i_flow) %active power flow
        i_meas = find(z2{s}(:,3) == 4 &  z2{s}(:,1) == i_flow(i)); %active power flow and
     belongs to indexed nodes in i_flow
```

```
94          if i_meas %is not empty
95              to_sub = max([extlinedata{s}(  extlinedata{s}(:,1) == s & extlinedata{s}(:,3)
    == i_flow(i), 2), ...
96                          extlinedata{s}(  extlinedata{s}(:,2) == s & extlinedata{s}(:,4)
    == i_flow(i), 1)]);
97              zdata2 = [zdata2; s, to_sub, 4, -z2{s}(i_meas,4), z2{s}(i_meas,5)];
98          end
99      end
100     for i = 1:length(i_flow) %reactive power flow
101         i_meas = find(z2{s}(:,3) == 5 &  z2{s}(:,1) == i_flow(i)); %reactive power flow
    and belongs to indexed nodes in i_flow
102         if i_meas %is not empty
103             to_sub = max([extlinedata{s}(  extlinedata{s}(:,1) == s & extlinedata{s}(:,3)
    == i_flow(i), 2), ...
104                         extlinedata{s}(  extlinedata{s}(:,2) == s & extlinedata{s}(:,4)
    == i_flow(i), 1)]);
105             zdata2 = [zdata2; s, to_sub, 5, -z2{s}(i_meas,4), z2{s}(i_meas,5)];
106         end
107     end
108 end
109 zdata2 = sortrows(zdata2, [3,1,2]);
110
111
112 %% Executing conventional SE
113 if display
114     disp('Executing conventional SE')
115 end
116
117 %Executing first SE:
118 [stateestimate, h2, r, r_N2, J, x] = wlsfunction(zdata2, ybus, bsh, g, b);
119
120 [r_N_max2, i] = max(r_N2);
121
122 %Bad data processing and rerunning of the SE:
123 while r_N_max2 > 3 && J>chi2inv(0.95,size(zdata,1)-n_sub*2+1)
124     if display
```

```
125         disp('Removed bad data in conventional SE:')
126         disp(zdata2(i,:));
127     end
128     zdata2(i,:)=[];
129     [stateestimate, h2, r, r_N2, J, x] = wlsfunction(zdata2, ybus, bsh, g, b, x);
130     [r_N_max2, i] = max(r_N2);
131 end
132
133 %Final conventional estimate:
134 Del2=stateestimate(:,3);
135 V2=stateestimate(:,2);
136
137 %% Displaying the outputs
138
139 if display
140     %% Printing results:
141     disp('-------------------------------------------------')
142     disp('------------------ Load flow --------------------');
143     disp('-------------------------------------------------');
144     disp('| Bus |    V    |  Angle  | ');
145     disp('| No  |   pu    |  Degree | ');
146     disp('---------------------------');
147     for m = 1:n_sub
148         fprintf('%4g', m); fprintf('  %8.4f', Vreal(m)); fprintf('   %8.4f', delreal(m)-
    delreal(1)); fprintf('\n');
149     end
150     disp('-------------------------------------------------');
151     disp('------ Conventional State Estimation  ---------');
152     disp('-------------------------------------------------');
153     disp('| Bus |    V    |  Angle  | ');
154     disp('| No  |   pu    |  Degree | ');
155     disp('---------------------------');
156     for m = 1:n_sub
157         fprintf('%4g', m); fprintf('  %8.4f', V2(m)); fprintf('   %8.4f', Del2(m));
    fprintf('\n');
158     end
```

```
159     disp('-------------------------------------------------');
160     disp('-------- Two-level State Estimation  ----------');
161     disp('-------------------------------------------------');
162     disp('| Bus |     V    |  Angle  | ');
163     disp('| No  |    pu    |  Degree | ');
164     disp('---------------------------');
165     for m = 1:n_sub
166         fprintf('%4g', m); fprintf('  %8.4f', V(m)); fprintf('   %8.4f', Del(m)); fprintf
        ('\n');
167     end
168
169     %% Plotting results
170     figure(1)
171     K = 1:n_sub;
172     plot(K, V2, 'b:*',K,V,'r:*',K,Vreal,'b--o', 'linewidth',1.5)
173     title('Voltage magnitude estimate')
174     xlabel('Bus number')
175     xlim([1 n_sub])
176     ylabel('Voltage (p.u.)')
177     legend('Conventional estimate', 'Two-level estimate','True Value')
178     grid on
179
180     figure(2)
181     j = 1:n_sub;
182     plot(j, Del2, 'b:*', j,Del,'r:*',j,delreal,'b--o', 'linewidth',1.5)
183     title('Voltage phase angle estimate')
184     xlabel('Bus number')
185     xlim([1 n_sub])
186     ylabel('Voltage phase angle (degrees)')
187     legend('Conventional estimate', 'Two-level estimate','True Value')
188     grid on
189 end
```

B.3 local.m

The function *local* executes the Substation Level State Estimation, when called by the *central.m* script.

```
% This function runs the Substation Level State Estimation, and is called
% by central.m.
function [zcentral, H] = local(zlocal, localbranches, extlinedata, nodetype, s, display)

n_nodes = max(max(localbranches));

%Number of state variables:
n_V = n_nodes;
n_P = size(localbranches,1);
n_Q = n_P;

%Constructing measurement vectors:
z_V = zlocal(zlocal(:,3)==1 | zlocal(:,3)==11, :);
z_P= zlocal(zlocal(:,3)==2 | zlocal(:,3) == 4 | zlocal(:,3) == 6 | zlocal(:,3) == 12, :);
z_Q = zlocal(zlocal(:,3)==3 | zlocal(:,3) == 5 | zlocal(:,3) == 7 | zlocal(:,3) == 13, :)
    ;
nz_V = size(z_V,1);

%% Substation Level State Estimation, including BD processing
contP = true;
contQ = true;
contV = true;
while contP || contQ || contV

    if contP
        contP = false;
        nz_P = size(z_P,1);
        H.P =  zeros(nz_P, n_P);

        for k = 1:nz_P
            if z_P(k,3) == 2 || (z_P(k,3) == 4 && z_P(k,2) == 0) % if measurement type 2
   and 4 (with external line)
```

```
31              H.P(k, z_P(k,1) == localbranches(:,1))   = 1;
32              H.P(k, z_P(k,1) == localbranches(:,2))   = -1;
33          else % if measurement type 4 or 12
34              H.P(k, (z_P(k,1) == localbranches(:,1)) & (z_P(k,2) == localbranches(:,2)
    ))   = 1;
35              H.P(k, (z_P(k,1) == localbranches(:,2)) & (z_P(k,2) == localbranches(:,1)
    ))   = -1;
36          end
37      end
38
39      %Orthogonal factorisation solution method
40      zvec_P = z_P(:,4);
41      Ginv2 = ( H.P' * ((diag(z_P(:,5))) \ H.P)) \ eye(n_P);
42      Winvsq_P = diag(sqrt(z_P(:,5)));
43      zvec_wP= Winvsq_P\zvec_P;
44      [Q, R] = qr(Winvsq_P\H.P);
45      Un = R(1:size(R,2),1:size(R,2));
46      Qn = Q(:,1:size(R,2));
47      zq = Qn'*zvec_wP;
48      x2 = Un\zq;
49
50      %Calculation of normalised residuals
51      r_P = -zvec_P+H.P*x2;
52      Ohm = diag(z_P(:,5))-H.P*Ginv2*H.P';
53      r_P_N = abs(r_P)./sqrt(abs(diag(Ohm)));
54      [r_P_N_max, i_P] = max(r_P_N);
55
56  end
57
58  if contQ
59      contQ = false;
60      nz_Q = size(z_Q,1);
61      H.Q =  zeros(nz_Q, n_Q);
62
63      for k = 1:nz_Q
```

```
64          if z_Q(k,3) == 3 || (z_Q(k,3) == 5 && z_Q(k,2) == 0) % if measurement type 3
    and 5 (with external line)
65              H.Q(k, z_Q(k,1) == localbranches(:,1))   = 1;
66              H.Q(k, z_Q(k,1) == localbranches(:,2))   = -1;
67          else % if measurement type 5 or type 13
68              H.Q(k, (z_Q(k,1) == localbranches(:,1)) & (z_Q(k,2) == localbranches(:,2)
    ))   = 1;
69              H.Q(k, (z_Q(k,1) == localbranches(:,2)) & (z_Q(k,2) == localbranches(:,1)
    ))   = -1;
70          end
71      end
72
73      %Orthogonal factorisation solution method
74      Ginv3 = ( H.Q' * ((diag(z_Q(:,5))) \ H.Q )) \ eye(n_Q);
75      zvec_Q = z_Q(:,4);
76      Winvsq_Q = diag(sqrt(z_Q(:,5)));
77      zvec_wQ = Winvsq_Q\zvec_Q;
78      [Q, R] = qr(Winvsq_Q\H.Q);
79      Un = R(1:size(R,2),1:size(R,2));
80      Qn = Q(:,1:size(R,2));
81      zq = Qn'*zvec_wQ;
82      x3 = Un\zq;
83
84      %Calculation of normalised residuals
85      r_Q = zvec_Q-H.Q*x3;
86      Ohm = diag(z_Q(:,5))-H.Q*Ginv3*H.Q';
87      r_Q_N = abs(r_Q)./sqrt(abs(diag(Ohm)));
88      [r_Q_N_max, j] = max(r_Q_N);
89  end
90
91  if contV
92      contV = false;
93      nz_V = size(z_V,1);
94      H.V =  zeros(nz_V, n_V);
95
96      for k = 1:nz_V
```

```
97            if z_V(k,3) == 1 %if conventional voltage measurement
98                H.V(k,z_V(k,1)) = 1;
99            else %if CB pseudo-measurement
100               H.V(k,z_V(k,1)) = 1;
101               H.V(k,z_V(k,2)) = -1;
102           end
103       end
104
105       %Cholesky factorisation solution method
106       zvec_V = z_V(:,4);
107       Ginv1 = ( H.V' * ((diag(z_V(:,5))) \ H.V )) \ eye(n_V);
108       x1 = Ginv1*H.V'*(diag(z_V(:,5))\zvec_V);
109
110       %Calculation of normalised residuals
111       r_V = zvec_V-H.V*x1;
112       Ohm = diag(z_V(:,5))-H.V*Ginv1*H.V';
113       r_V_N = abs(r_V)./sqrt(abs(diag(Ohm)));
114       [r_V_N_max, i_V] = max(r_V_N);
115   end
116
117   %Bad data detection
118   P_BDdetected = r_P_N_max > 3; %alternative, the chi square criterion: r_P'*(diag(z_P
      (:,5))\r_P) > chi2inv(0.99,length(r_P)-n_P);
119   Q_BDdetected = r_Q_N_max > 3; %alternative, the chi square criterion: r_Q'*(diag(z_Q
      (:,5))\r_Q) > chi2inv(0.99,length(r_Q)-n_Q);
120
121   if (P_BDdetected && ~Q_BDdetected && r_P_N_max > 3) || (P_BDdetected && r_P_N_max >
      r_Q_N_max && r_P_N_max > 3) %error in active power
122       if  z_P(i_P,3) == 12 %incorrect open breaker
123           z_Q(z_Q(:,1) == z_P(i_P,1) & z_Q(:,2) == z_P(i_P,2) & z_Q(:,3) == 13, :) =
      []; %removes closed breaker constraint on Q also
124           if display
125               fprintf('Removed open breaker topology error in substation %4g:', s)
126               disp(z_P(i_P,1:2));
127           end
128
```

```
129             %Enforcing the opposite CB pseudo-measurement:
130             z_V = [z_V; z_P(i_P, 1:2), 11 ,z_P(i_P, 4:5)];
131             nz_V = nz_V +1;
132             z_P(i_P, :) = [];
133             z_Q(1,:) = [];
134             contP = true;
135             contQ = true;
136             contV = true;
137         else %bad analogue data
138             if display
139                 fprintf('Removed bad data in substation %4g:', s)
140                 disp(z_P(i_P,:));
141             end
142             z_P(i_P,:) =[];
143             contP = true;
144         end
145
146     elseif (Q_BDdetected && ~P_BDdetected && r_Q_N_max > 3) || (Q_BDdetected && r_Q_N_max
     > r_P_N_max && r_Q_N_max > 3) %error in reactive power
147         if z_Q(j,3) == 13 %incorrect open breaker
148             z_P(z_P(:,1) == z_Q(j,1) & z_P(:,2) == z_Q(j,2) & z_P(:,3) == 13, :) = [];  %
    removes closed breaker constraint on P also
149             if display
150                 fprintf('Removed open breaker topology error in substation %4g:', s)
151                 disp(z_Q(j,1:2));
152             end
153
154             %Enforcing the opposite CB pseudo-measurement:
155             z_V = [z_V; z_Q(j, 1:2), 11 ,z_Q(j, 4:5)];
156             nz_V = nz_V +1;
157             z_Q(j, :) = [];
158             contP = true;
159             contQ = true;
160             contV = true;
161         else  %bad analogue data
162             if display
```

```
163                 fprintf('Removed bad data in substation %4g:', s)
164                 disp(z_Q(j,:));
165             end
166             z_Q(j,:)=[];
167             contQ = true;
168         end
169
170     elseif r_V_N_max > 3
171          if z_V(i_V,3) == 11 %incorrect closed breaker
172             if display
173                 fprintf('Removed closed breaker topology error in substation %4g:', s)
174                 disp(z_V(i_ind,1:2));
175             end
176
177             %Enforcing the opposite CB pseudo-measurement:
178             z_P = [z_P; z_P(i_V, 1:2), 12 ,z_P(i_V, 4:5)];
179             z_Q = [z_Q; z_Q(i_V, 1:2), 13 ,z_Q(i_V, 4:5)];
180             nz_P = nz_P +1;
181             nz_Q = nz_Q +1;
182             z_V(i_V, :) = [];
183             contP = true;
184             contQ = true;
185             contV = true;
186          else %bad analogue data
187             if display
188                 fprintf('Removed bad data in substation %4g:', s)
189                 disp(z_V(i_V,:));
190             end
191             z_V(i_V,:)=[];
192             contV = true;
193         end
194     end
195 end
196
197
198 %% Bulding the pre-processed measurements of the central SE
```

```
199 Vbus = mean(x1);
200 Vbus_sigma = mean(diag(Ginv1));
201
202 %Building H.Pinj and H.Qinj matrices:
203     H.Pinj = zeros(n_nodes,1);  %injected power / incoming power flow on every node
204     for k = 1:n_nodes
205         H.Pinj(k,   (k) == localbranches(:,1)) = 1;
206         H.Pinj(k,   (k) == localbranches(:,2)) = -1;
207     end
208
209 %Calculating out to central SE
210 Pinj = H.Pinj*x2;
211 Pinj_sigma = H.Pinj*Ginv2*H.Pinj';
212
213 %% Building final measurement matrix from current substation:
214 index1 = find(extlinedata(:, 1) == s);
215 index2 = find(extlinedata(:, 2) == s);
216 zcentral = zeros(3+2*(length(index1)+length(index2)), 5);
217
218
219 %Active power flows:
220 for i = 1:length(index1)
221     zcentral(2+2*i,:) = [s,  extlinedata(index1(i), 2),  4, -Pinj(extlinedata(index1(i),
    3)), Pinj_sigma(extlinedata(index1(i), 3), extlinedata(index1(i), 3))];
222 end
223 for i = 1:length(index2)
224     zcentral(2+2*i+2*length(index1),:) = [s, extlinedata(index2(i), 1), 4, -Pinj(
    extlinedata(index2(i), 4)), Pinj_sigma(extlinedata(index2(i), 4), extlinedata(index2(
    i), 4))];
225 end
226
227
228
229
230
231 H.Qinj = zeros(n_nodes,1);
```

```
for k = 1:n_nodes
    H.Qinj(k,   (k) == localbranches(:,1)) = 1;
    H.Qinj(k,   (k) == localbranches(:,2)) = -1;
end

Qinj = H.Qinj*x3;
Qinj_sigma = H.Qinj*Ginv3*H.Qinj';
%include phase angles?

%Reactive power flows:
for i = 1:length(index1)
    zcentral(3+2*i,:) = [ s, extlinedata(index1(i), 2), 5, -Qinj(extlinedata(index1(i),
    3)), Qinj_sigma(extlinedata(index1(i), 3), extlinedata(index1(i), 3))];
end
for i = 1:length(index2)
    zcentral(3+2*i+2*length(index1),:) = [s, extlinedata(index2(i), 1),  5, -Qinj(
    extlinedata(index2(i), 4)), Qinj_sigma(extlinedata(index2(i), 4), extlinedata(index2(
    i), 4))];
end

%Voltage
zcentral(1,:) = [s, 0, 1, Vbus, Vbus_sigma];
%zcentral(1,:) = [s, 0, 1, mean(zlocal(zlocal(:,3)==1, 4)), mean(zlocal(zlocal(:,3)==1,
    5))/length(zlocal(zlocal(:,3)==1, 5))];

%Power injections:
ind = nodetype((nodetype(:,2)==1 | nodetype(:,2)==2 | nodetype(:,2)==3),1);
zcentral(2,:) = [s, 0, 2, sum(Pinj(ind)), max([10^-12, sum(sum(Pinj_sigma(ind,ind)))])];

ind = nodetype((nodetype(:,2)==1 | nodetype(:,2)==2 | nodetype(:,2)==3),1);
zcentral(3,:) = [s, 0, 3, sum(Qinj(ind)), max([10^-12, sum(sum(Qinj_sigma(ind,ind)))])];

end
```

B.4 wlsfunction.m

The function *wlsfunction* executes the conventional state estimation using weighted least square method.

```
% This function runs a conventional SE using Weighted Least Square Method

function [ result, h, r, r_N, J, x] = wlsfunction(zdata, ybus, bsh, g, b, x)

fbus = zdata(:,1); % from bus
tbus = zdata(:,2); % to bus
nbus = max([fbus;tbus]); % number of buses
type = zdata(:,3); % type of measurement
z = zdata(:,4); % measurement values

Ri = diag(zdata(:,5)); % measurement variance matrix
W = Ri\eye(size(Ri));
B = imag(ybus);
G = real(ybus);

%Initiating state variables
if exist('x', 'var') == 0
    V = ones(nbus,1); % Initialize the bus voltages..
    O = zeros(nbus,1);
    x = [O(2:end); V];   % State Vector, not including Theta_1
else
    O = [0; x(1:(nbus-1))];
    V = x(nbus:end);
end

%Indexes in zdata of the different type of measurements:
vi = find(type == 1); % Index of voltage magnitude measurements
ppi = find(type == 2); % Index of active power injection measurements
qi = find(type == 3); % Index of reactive power injection measurements
pf = find(type == 4); % Index of active power flow measurements
qf = find(type == 5); % Index of reactive power flow measurements
%vir = find(type == 6); % Index of real voltage measurements
```

```
%vij = find(type == 7); % Index of imaginary voltage measurements
ifr = find(type == 8); % Index of real current measurements
ifj = find(type == 9); % Index of imaginary current measurements
adi = find(type == 10); %Index of delta differences
vdi = find(type == 11); %Index of voltage differences

%Number of measurements of the different types:
nvi = length(vi); % Number of voltage measurements
npi = length(ppi); % Number of real power injection measurements
nqi = length(qi); % Number of reactive power injection measurements
npf = length(pf); % Number of real power flow measurements
nqf = length(qf); % Number of reactive power flow measurements
%nvir = length(vir); % Number of real voltage measurements
%nvij = length(vij); % Number of imaginary voltage measurements
nifr = length(ifr); % Number of real current measurements
nifj = length(ifj); % Number of imaginary current measurements
nadi = length(adi); %
nvdi = length(vdi); %

iter = 1;
tol = 5;
while(tol > 1e-5 && iter < 15)
    %% Building h vector
    h1 = V(fbus(vi),1);
    h2 = zeros(npi,1);
    h3 = zeros(nqi,1);
    h4 = zeros(npf,1);
    h5 = zeros(nqf,1);
    %h6 = V(fbus(vir)).*cos(O(fbus(vir)));
    %h7 = V(fbus(vij)).*sin(O(fbus(vij)));
    h8 = zeros(nifr, 1);
    h9 = zeros(nifj, 1);

    for i = 1:npi
        m = fbus(ppi(i));
        for k = 1:nbus
```

```
69            h2(i) = h2(i) + V(m)*V(k)*(G(m,k)*cos(O(m)-O(k)) + B(m,k)*sin(O(m)-O(k)));
70        end
71    end
72
73    for i = 1:nqi
74        m = fbus(qi(i));
75        for k = 1:nbus
76            h3(i) = h3(i) + V(m)*V(k)*(G(m,k)*sin(O(m)-O(k)) - B(m,k)*cos(O(m)-O(k)));
77        end
78    end
79
80    for i = 1:npf
81        m = fbus(pf(i));
82        n = tbus(pf(i));
83        h4(i) = V(m)^2*g(m,n) - V(m)*V(n)*(g(m,n)*cos(O(m)-O(n)) + b(m,n)*sin(O(m)-O(n)))
    ;
84    end
85
86    for i = 1:nqf
87        m = fbus(qf(i));
88        n = tbus(qf(i));
89        h5(i) = -V(m)^2*(b(m,n)+bsh(m,n)) - V(m)*V(n)*(g(m,n)*sin(O(m)-O(n)) - b(m,n)*cos
    (O(m)-O(n)));
90    end
91
92    for i = 1:nifr
93        m = fbus(ifr(i));
94        n = tbus(ifr(i));
95        h8(i) = ( V(m)*cos(O(m))-V(n)*cos(O(n)) )*g(m,n)  - ( V(m)*sin(O(m)) - V(n)*sin(O
    (n)) )*b(m,n) - bsh(1,2)*V(m)*sin(O(m));
96    end
97
98    for i = 1:nifj
99        m = fbus(ifj(i));
100       n = tbus(ifj(i));
```

```
101         h9(i) = ( ( V(m)*cos(O(m))-V(n)*cos(O(n)) )*b(m,n)  + ( V(m)*sin(O(m)) - V(n)*sin
    (O(n)) )*g(m,n) + bsh(m,n)*V(m)*cos(O(m)));
102     end
103
104     h = [h1; h2; h3; h4; h5; h8; h9];
105
106     r = z - h; % residual
107
108     %% Building H matrix
109
110     % H11 - Derivative of V with resepect to phase angles
111     H11 = zeros(nvi,nbus-1);
112
113     % H12 - Derivative of V with respect to V
114     H12 = zeros(nvi,nbus);
115     for k = 1:nvi
116         H12(k,fbus(vi(k))) = 1;
117     end
118
119     % H21 - Derivative of Real Power Injections with resepect to phase angles
120     H21 = zeros(npi,nbus-1);
121     for i = 1:npi
122         m = fbus(ppi(i));
123         for k = 1:(nbus-1)
124             if k+1 == m
125                 for n = 1:nbus
126                     H21(i,k) = H21(i,k) + V(m)* V(n)*(-G(m,n)*sin(O(m)-O(n)) + B(m,n)*cos
    (O(m)-O(n)));
127                 end
128                 H21(i,k) = H21(i,k) - V(m)^2*B(m,m);
129             else
130                 H21(i,k) = V(m)* V(k+1)*(G(m,k+1)*sin(O(m)-O(k+1)) - B(m,k+1)*cos(O(m)-O(
    k+1)));
131             end
132         end
133     end
```

```
134
135 % H22 - Derivative of Real Power Injections with V
136 H22 = zeros(npi,nbus);
137 for i = 1:npi
138     m = fbus(ppi(i));
139     for k = 1:(nbus)
140         if k == m
141             for n = 1:nbus
142                 H22(i,k) = H22(i,k) + V(n)*(G(m,n)*cos(O(m)-O(n)) + B(m,n)*sin(O(m)-O
(n)));
143             end
144             H22(i,k) = H22(i,k) + V(m)*G(m,m);
145         else
146             H22(i,k) = V(m)*(G(m,k)*cos(O(m)-O(k)) + B(m,k)*sin(O(m)-O(k)));
147         end
148     end
149 end
150
151 % H31 - Derivative of Reactive Power Injections with resepect to phase angles
152 H31 = zeros(nqi,nbus-1);
153 for i = 1:nqi
154     m = fbus(qi(i));
155     for k = 1:(nbus-1)
156         if k+1 == m
157             for n = 1:nbus
158                 H31(i,k) = H31(i,k) + V(m)* V(n)*(G(m,n)*cos(O(m)-O(n)) + B(m,n)*sin(
O(m)-O(n)));
159             end
160             H31(i,k) = H31(i,k) - V(m)^2*G(m,m);
161         else
162             H31(i,k) = V(m)* V(k+1)*(-G(m,k+1)*cos(O(m)-O(k+1)) - B(m,k+1)*sin(O(m)-O
(k+1)));
163         end
164     end
165 end
166
```

```
167 % H32 - Derivative of Reactive Power Injections with V
168 H32 = zeros(nqi,nbus);
169 for i = 1:nqi
170     m = fbus(qi(i));
171     for k = 1:(nbus)
172         if k == m
173             for n = 1:nbus
174                 H32(i,k) = H32(i,k) + V(n)*(G(m,n)*sin(O(m)-O(n)) - B(m,n)*cos(O(m)-O
    (n)));
175             end
176             H32(i,k) = H32(i,k) - V(m)*B(m,m);
177         else
178             H32(i,k) = V(m)*(G(m,k)*sin(O(m)-O(k)) - B(m,k)*cos(O(m)-O(k)));
179         end
180     end
181 end
182
183 % H41 - Derivative of Real Power Flows with resepect to phase angles
184 H41 = zeros(npf,nbus-1);
185 for i = 1:npf
186     m = fbus(pf(i));
187     n = tbus(pf(i));
188     for k = 1:(nbus-1)
189         if k+1 == m
190             H41(i,k) = V(m)* V(n)*(g(m,n)*sin(O(m)-O(n)) - b(m,n)*cos(O(m)-O(n)));
191         elseif k+1 == n
192             H41(i,k) = -V(m)* V(n)*(g(m,n)*sin(O(m)-O(n)) - b(m,n)*cos(O(m)-O(n)));
193         end
194     end
195 end
196
197 % H42 - Derivative of Real Power Flows with V
198 H42 = zeros(npf,nbus);
199 for i = 1:npf
200     m = fbus(pf(i));
201     n = tbus(pf(i));
```

```
202         for k = 1:nbus
203             if k == m
204                 H42(i,k) = -V(n)*(-G(m,n)*cos(O(m)-O(n)) - B(m,n)*sin(O(m)-O(n))) - 2*G(m
    ,n)*V(m);
205             elseif k == n
206                 H42(i,k) = -V(m)*(-G(m,n)*cos(O(m)-O(n)) - B(m,n)*sin(O(m)-O(n)));
207             end
208         end
209     end
210
211     % H51 - Derivative of Reactive Power Flows with resepect to phase angles
212     H51 = zeros(nqf,nbus-1);
213     for i = 1:nqf
214         m = fbus(qf(i));
215         n = tbus(qf(i));
216         for k = 1:(nbus-1)
217             if k+1 == m
218                 H51(i,k) = -V(m)* V(n)*(-G(m,n)*cos(O(m)-O(n)) - B(m,n)*sin(O(m)-O(n)));
219             elseif k+1 == n
220                 H51(i,k) = V(m)* V(n)*(-G(m,n)*cos(O(m)-O(n)) - B(m,n)*sin(O(m)-O(n)));
221             end
222         end
223     end
224
225     % H52 - Derivative of Reactive Power Flows with V
226     H52 = zeros(nqf,nbus);
227     for i = 1:nqf
228         m = fbus(qf(i));
229         n = tbus(qf(i));
230         for k = 1:nbus
231             if k == m
232                 H52(i,k) = -V(n)*(-G(m,n)*sin(O(m)-O(n)) + B(m,n)*cos(O(m)-O(n))) - 2*V(m
    )*(-B(m,n)+ bsh(m,n));
233             elseif k == n
234                 H52(i,k) = -V(m)*(-G(m,n)*sin(O(m)-O(n)) + B(m,n)*cos(O(m)-O(n)));
235             end
```

```
236         end
237     end
238 
239 %     % H61 - Derivative of V_R with resepect to phase angles
240 %     H61 = zeros(nvir,nbus-1);
241 %     for i = 1:nvir
242 %         m = fbus(vir(i));
243 %         for k = 1:(nbus-1)
244 %             if (k+1) == m
245 %                 H61(i,k) = -V(m)*sin(O(m));
246 %             end
247 %         end
248 %     end
249 %
250 %     % H62 - Derivative of V_R with respect to V
251 %     H62 = zeros(nvir,nbus);
252 %     for i = 1:nvir
253 %         m = fbus(vir(i));
254 %         for k = 1:nbus
255 %             if k == m
256 %                 H62(i,k) = cos(O(m));
257 %             end
258 %         end
259 %     end
260 %
261 %     % H71 - Derivative of V_imag with resepect to phase angles
262 %     H71 = zeros(nvij,nbus-1);
263 %     for i = 1:nvij
264 %         m = fbus(vij(i));
265 %         for k = 1:(nbus-1)
266 %             if (k+1) == m
267 %                 H71(i,k) = V(m)*cos(O(m));
268 %             end
269 %         end
270 %     end
271 %
```

```
%     % H72 - Derivative of V_imag with respect to V
%     H72 = zeros(nvij,nbus);
%     for i = 1:nvij
%         m = fbus(vij(i));
%         for k = 1:nbus
%             if k == m
%                 H72(i,k) = sin(O(m));
%             end
%         end
%     end

    % H81 - Derivative of I_real with resepect to phase angles
    H81 = zeros(nifr,nbus-1);
    for i = 1:nifr
        m = fbus(ifr(i));
        n = tbus(ifr(i));
        for k = 1:(nbus-1)
            if k+1 == m
                H81(i,k) = -V(m)*( g(m,n)*sin(O(m))+b(m,n)*cos(O(m))+bsh(m)*cos(O(m)));
            elseif k+1 == n
                H81(i,k) = V(n)*( g(m,n)*sin(O(n))+b(m,n)*cos(O(n)));
            end
        end
    end


    % H82 - Derivative of I_real with respect to V
    H82 = zeros(nifr,nbus);
    for i = 1:nifr
        m = fbus(ifr(i));
        n = tbus(ifr(i));
        for k = 1:nbus
            if k == m
                H82(i,n) = g(m,n)*cos(O(m)) - b(m,n)*sin(O(m)) - bsh(m)*sin(O(m));
            elseif k == n
                H82(i,n) = - g(m,n)*cos(O(n))+b(m,n)*sin(O(n));
```

```
308             end
309         end
310     end
311
312     % H91 - Derivative of I_imag with resepect to phase angles
313     H91 = zeros(nifj,nbus-1);
314     for i = 1:nifj
315         m = fbus(ifj(i));
316         n = tbus(ifj(i));
317         for k = 1:(nbus-1)
318             if (k+1) == m
319                 H91(i,k) = V(m) * (-b(m,n)*sin(O(m)) - g(m,n)*cos(O(m)) + bsh(m)*sin(O(m)
    ));
320             elseif (k+1) == n
321                 H91(i,k) = V(n)* (b(m,n)*sin(O(n))+g(m,n)*cos(O(n)));
322             end
323         end
324     end
325
326     % H92 - Derivative of I_imag with respect to V
327     H92 = zeros(nifj,nbus);
328     for i = 1:nifj
329         m = fbus(ifj(i));
330         n = tbus(ifj(i));
331         for k = 1:nbus
332             if k == m
333                 H92(i,k) = (b(m,n)*cos(O(m)) - g(m,n)*sin(O(m)) - bsh(m)*cos(O(m)));
334             elseif k == n
335                 H92(i,k) = (-b(m,n)*cos(O(n))+g(m,n)*sin(O(n)));
336             end
337         end
338     end
339
340     % The resulting H matrix:
341     H = [H11 H12; H21 H22; H31 H32; H41 H42; H51 H52; H81 H82; H91 H92;];
342
```

```
343    %%% Calculating correction dx
344
345     Gm = H'*W*H; % Gain Matrix
346
347     dx = Gm\(H'*W*r); % State update
348     x = x + dx; % State estimate
349
350     O(2:end) = x(1:nbus-1); %phase angle
351     V = x(nbus:end); %magnitude
352
353     iter = iter + 1;
354     tol = max(abs(dx));
355
356 end
357
358 % Final state estimate:
359 Del = 180/pi*O;
360 result = zeros(nbus,3);
361 for m = 1:nbus
362     result(m,1) = m;
363     result(m,2) = V(m);
364     result(m,3) = Del(m);
365 end
366
367 %% Building the final h vector
368 h1 = V(fbus(vi),1);
369 h2 = zeros(npi,1);
370 h3 = zeros(nqi,1);
371 h4 = zeros(npf,1);
372 h5 = zeros(nqf,1);
373 %h6 = V(fbus(vir)).*cos(O(fbus(vir)));
374 %h7 = V(fbus(vij)).*sin(O(fbus(vij)));
375 h8 = zeros(nifr, 1);
376 h9 = zeros(nifj, 1);
377
378 for i = 1:npi
```

```
    m = fbus(ppi(i));
    for k = 1:nbus
        h2(i) = h2(i) + V(m)*V(k)*(G(m,k)*cos(O(m)-O(k)) + B(m,k)*sin(O(m)-O(k)));
    end
end

for i = 1:nqi
    m = fbus(qi(i));
    for k = 1:nbus
        h3(i) = h3(i) + V(m)*V(k)*(G(m,k)*sin(O(m)-O(k)) - B(m,k)*cos(O(m)-O(k)));
    end
end

for i = 1:npf
    m = fbus(pf(i));
    n = tbus(pf(i));
    h4(i) = V(m)^2*g(m,n) - V(m)*V(n)*(g(m,n)*cos(O(m)-O(n)) + b(m,n)*sin(O(m)-O(n)));
end

for i = 1:nqf
    m = fbus(qf(i));
    n = tbus(qf(i));
    h5(i) = -V(m)^2*(b(m,n)+bsh(m,n)) - V(m)*V(n)*(g(m,n)*sin(O(m)-O(n)) - b(m,n)*cos(O(m
    )-O(n)));
end

for i = 1:nifr
    m = fbus(ifr(i));
    n = tbus(ifr(i));
    h8(i) = ( V(m)*cos(O(m))-V(n)*cos(O(n)) )*g(m,n) - ( V(m)*sin(O(m)) - V(n)*sin(O(n))
     )*b(m,n) - bsh(1,2)*V(m)*sin(O(m));
end

for i = 1:nifj
    m = fbus(ifj(i));
    n = tbus(ifj(i));
```

```
413     h9(i) = ( ( V(m)*cos(O(m))-V(n)*cos(O(n)) )*b(m,n)  + ( V(m)*sin(O(m)) - V(n)*sin(O(n
        )) )*g(m,n) + bsh(m,n)*V(m)*cos(O(m)));
414 end
415
416
417 %% Additional output variables:
418 h = [h1; h2; h3; h4; h5; h8; h9]; %final measurement estimate
419 r = z - h; % residual
420 Ohm = Ri-H*(Gm\H');
421 r_N = abs(r)./sqrt(abs(diag(Ohm))); %normalised residuals
422 J=r'*W*r; %objective function value
423
424 end
```

B.5 montecarlo.m

The *montecarlo.m* script runs Monte Carlo simulations, calling the *central.m* script, and produces box plots for illustration.

```
1 % A script for running Monte Carlo simulations on the central.m script and
2 % producing box plots.
3
4 %% Initiating the variables:
5 n_iterations=10000;
6
7 zest_two = zeros(n_iterations,length(kvec));
8 zest_conv = zeros(n_iterations,length(kvec));
9
10 zpre_two = zeros(n_iterations,length(kvec));
11 z_conv = zeros(n_iterations,length(kvec));
12 zest_two_without = zeros(n_iterations,length(kvec));
13
14 %% Running Monte Carlo simulations
15 for iteration = 1:n_iterations
16
17     central; %running central.m script (display should be set to false within central.m)
```

```
18
19      %Estimated power flows from substation 2 to substation 1:
20      zest_two_without(iteration) = V0(2)^2*4.9991 - V0(1)*V0(2)*(4.9991*cos((Del0(2)-Del0
        (1))*pi/180) - 15.2631*sin((Del0(2)-Del0(1))*pi/180));
21      zest_two(iteration) = V(2)^2*4.9991 - V(1)*V(2)*(4.9991*cos((Del(2)-Del(1))*pi/180) -
         15.2631*sin((Del(2)-Del(1))*pi/180));
22      zest_conv(iteration) = V2(2)^2*4.9991 - V2(1)*V2(2)*(4.9991*cos((Del2(2)-Del2(1))*pi
        /180) - 15.2631*sin((Del2(2)-Del2(1))*pi/180));
23
24      %Pre-processed and raw measurements:
25      zpre_two(iteration) = zcentral{2}(10,4);
26      z_conv(iteration) = zdata2(18,4);
27  end
28
29  %% Calculating results and displaying
30  %Calculate true measurement:
31  save('realestimate')
32  zreal = Vreal(2)^2*4.9991 - Vreal(1)*Vreal(2)*(4.9991*cos((Delreal(2)-Delreal(1))*pi/180)
        - 15.2631*sin((Delreal(2)-Delreal(1))*pi/180));
33
34  %Calculating estimate errors:
35  err1 =  zest_two_2_without - zreal;
36  err2 = zest_two_2-zreal;
37  err3= zest_conv_2-zreal;
38
39  %Calculating the estimate variances from samples:
40  v2ariance1 = sqrt(var(err1));
41  v2ariance2 = sqrt(var(err2));
42  v2ariance3 = sqrt(var(err3));
43
44  %Creating box plot:
45  figure(12)
46  boxplot([err1, err2, err3], 'Labels', {'Two-level SE without central BD detection', 'Two-
        level SE','Conventional SE'});
47  ylabel('Estimate error')
48  grid on
```

Appendix C

Simulation input data

C.1 Grid model

File: operation_linedata.xlsx

Table C.1: Grid model

From_sub	To_sub	From_node	To_node	r	x	b_half
1	1	1	2	0	0	0
1	1	1	5	0	0	0
1	1	2	3	0	0	0
1	1	3	4	0	0	0
1	1	4	6	0	0	0
1	1	5	6	0	0	0
1	2	6	1	0.01938	0.05917	0.0528
1	5	5	3	0.05403	0.22304	0.0492
2	2	1	2	0	0	0
2	2	1	3	0	0	0
2	2	2	5	0	0	0
2	2	3	4	0	0	0
2	2	4	5	0	0	0
2	3	5	3	0.04699	0.19797	0.0438
2	4	4	6	0.05811	0.17632	0.034
2	5	3	5	0.05695	0.17388	0.0346
3	3	1	3	0	0	0
3	3	1	4	0	0	0
3	3	2	3	0	0	0
3	3	2	4	0	0	0
3	4	4	8	0.06701	0.17103	0.0128
4	4	1	3	0	0	0
4	4	1	5	0	0	0
4	4	1	7	0	0	0
4	4	2	4	0	0	0
4	4	2	6	0	0	0
4	4	2	8	0	0	0
4	4	3	4	0	0	0
4	4	5	6	0	0	0
4	4	7	8	0	0	0
4	5	4	6	0.01335	0.04211	0
5	5	1	2	0	0	0
5	5	1	3	0	0	0
5	5	1	4	0	0	0
5	5	1	5	0	0	0
5	5	1	6	0	0	0
5	5	1	7	0	0	0
5	5	2	3	0	0	0
5	5	2	4	0	0	0
5	5	2	5	0	0	0
5	5	2	6	0	0	0
5	5	2	7	0	0	0

C.2 Operation inputs

File: operation_database.xlsx

Table C.2: Sheet 1, substation 1 operating conditions

Node	Type	Pinj	Qinj	Vg
1	4			
2	3			1.06
3	1	-50	-10	
4	4			
5	5			
6	5			

Table C.3: Sheet 2, substation 2 operating conditions

Node	Type	Pinj	Qinj	Vg
1	5			
2	2	18.3		1.045
3	5			
4	5			
5	5			

Table C.4: Sheet 3, substation 3 operating conditions

Node	Type	Pinj	Qinj	Vg
1	1		-19	
2	2	0		1.01
3	5			
4	5			

Table C.5: Sheet 4, substation 4 operating conditions

Node	Type	Pinj	Qinj	Vg
1	4			
2	4			
3	1	-10	-1	
4	5			
5	1	-10	-1	
6	5			
7	1	-27.8	1.9	
8	5			

Table C.6: Sheet 5, substation 5 operating conditions

Node	Type	Pinj	Qinj	Vg
1	4			
2	4			
3	5			
4	1	-5	-1	
5	5			
6	5			
7	1	-2.6	-0.6	

Brnch #	From Bus	To Bus	From Bus Injection P (MW)	From Bus Injection Q (MVAr)	To Bus Injection P (MW)	To Bus Injection Q (MVAr)	Loss (I^2 * Z) P (MW)	Loss (I^2 * Z) Q (MVAr)
1	1	2	-41.64	0.11	41.64	-0.11	0.000	0.00
2	1	5	41.64	-0.11	-41.64	0.11	-0.000	0.00
3	2	3	145.57	5.08	-145.57	-5.08	0.000	0.00
4	3	4	95.57	-4.92	-95.57	4.92	0.000	0.00
5	4	6	95.57	-4.92	-95.57	4.92	0.000	0.00
6	5	6	0.00	-0.00	-0.00	0.00	0.000	0.00
7	6	7	95.57	-4.92	-94.00	3.89	1.576	4.81
8	5	26	41.64	-0.11	-40.81	-1.84	0.837	3.46
9	7	8	94.00	-3.89	-94.00	3.89	0.000	0.00
10	7	9	0.00	-0.00	-0.00	0.00	0.000	0.00
11	8	11	112.30	-0.77	-112.30	0.77	0.000	0.00
12	9	10	-19.47	3.02	19.47	-3.02	0.000	0.00
13	10	11	-51.24	5.68	51.24	-5.68	-0.000	0.00
14	11	14	61.05	4.90	-59.43	-2.67	1.627	6.85
15	10	21	31.77	-2.66	-31.23	0.63	0.538	1.63
16	9	28	19.47	-3.02	-19.27	-0.13	0.198	0.61
17	12	14	-0.00	-0.00	0.00	0.00	0.000	0.00
18	12	15	-94.20	-19.00	94.20	19.00	0.000	0.00
19	13	14	-59.43	-2.67	59.43	2.67	0.000	0.00
20	13	15	59.43	21.16	-59.43	-21.16	0.000	0.00
21	15	23	-34.77	2.15	35.57	-1.45	0.799	2.04
22	16	18	-37.80	0.90	37.80	-0.90	0.000	0.00
23	16	20	10.00	1.00	-10.00	-1.00	0.000	0.00
24	16	22	27.80	-1.90	-27.80	1.90	0.000	0.00
25	17	19	-4.34	0.82	4.34	-0.82	0.000	0.00
26	17	21	-31.23	0.63	31.23	-0.63	0.000	0.00
27	17	23	35.57	-1.45	-35.57	1.45	0.000	0.00
28	18	19	-47.80	-0.10	47.80	0.10	0.000	0.00
29	20	21	-0.00	-0.00	0.00	0.00	0.000	0.00
30	22	23	-0.00	0.00	0.00	-0.00	0.000	0.00
31	19	29	-52.14	0.72	52.48	0.36	0.342	1.08
32	24	25	0.00	-0.00	-0.00	0.00	0.000	0.00
33	24	26	-40.81	-1.84	40.81	1.84	0.000	0.00
34	24	27	5.00	1.00	-5.00	-1.00	0.000	0.00
35	24	28	-19.27	-0.13	19.27	0.13	0.000	0.00
36	24	29	52.48	0.36	-52.48	-0.36	-0.000	0.00
37	24	30	2.60	0.60	-2.60	-0.60	0.000	0.00
38	25	26	-0.00	-0.00	0.00	0.00	0.000	0.00
39	25	27	0.00	0.00	-0.00	-0.00	-0.000	0.00
40	25	28	-0.00	-0.00	0.00	0.00	0.000	0.00
41	25	29	0.00	0.00	-0.00	-0.00	0.000	0.00

Figure C.1: Load flow results, showing branch data.

Bus	Voltage		Generation		Load	
#	Mag(pu)	Ang(deg)	P (MW)	Q (MVAr)	P (MW)	Q (MVAr)
1	1.060	-0.000	-	-	-	-
2	1.060	0.000*	187.22	4.97	-	-
3	1.060	-0.001	-	-	50.00	10.00
4	1.060	-0.001	-	-	-	-
5	1.060	-0.000	-	-	-	-
6	1.060	-0.002	-	-	-	-
7	1.045	-2.948	-	-	-	-
8	1.045	-2.948	18.30	3.11	-	-
9	1.045	-2.949	-	-	-	-
10	1.045	-2.949	-	-	-	-
11	1.045	-2.949	-	-	-	-
12	1.010	-9.339	-	-	94.20	19.00
13	1.010	-9.338	0.00	18.48	-	-
14	1.010	-9.337	-	-	-	-
15	1.010	-9.338	-	-	-	-
16	1.030	-5.958	-	-	-	-
17	1.030	-5.957	-	-	-	-
18	1.030	-5.957	-	-	10.00	1.00
19	1.030	-5.957	-	-	-	-
20	1.030	-5.958	-	-	10.00	1.00
21	1.030	-5.957	-	-	-	-
22	1.030	-5.958	-	-	27.80	-1.90
23	1.030	-5.957	-	-	-	-
24	1.037	-4.774	-	-	-	-
25	1.037	-4.774	-	-	-	-
26	1.037	-4.774	-	-	-	-
27	1.037	-4.774	-	-	5.00	1.00
28	1.037	-4.774	-	-	-	-
29	1.037	-4.774	-	-	-	-
30	1.037	-4.774	-	-	2.60	0.60
		Total:	205.52	26.56	199.60	30.70

Figure C.2: Load flow results, showing bus data.

C.3 Measurement sets

C.3.1 Measurement case 1

File: meas_database_case1.xlsx

Table C.7: Sheet 1, substation 1 measurements

from_node	to_node	type	value	R_ii
1		1		
2		1		
3		1		
4		1		
5		1		
2		2		
2		3		
3		2		
3		3		
5		4		
5		5		
6		4		
6		5		
1	2	4		0.05
1	2	5		0.05
2	3	4		0.05
2	3	5		0.05
3	4	4		0.05
3	4	5		0.05
4	6	4		0.05
4	6	5		0.05
5	6	4		0.05
5	6	5		0.05
1	5	4		0.05
1	5	5		0.05

Table C.8: Sheet 2, substation 2 measurements

from_node	to_node	type	value	R_ii
1		1		
2		1		
3		1		
4		1		
5		1		
2		2		
2		3		
1		4		
1		5		
3		4		
3		5		
4		4		
4		5		
5		4		
5		5		
1	2	4		0.05
1	2	5		0.05
1	3	4		0.05
1	3	5		0.05
2	5	4		0.05
2	5	5		0.05
4	5	4		0.05
4	5	5		0.05
3	4	4		0.05
3	4	5		0.05

Table C.9: Sheet 3, substation 3 measurements

from_node	to_node	type	value	R_ii
1		1		
2		1		
3		1		
4		1		
1		2		
1		3		
2		2		
2		3		
3		4		
3		5		
4		4		
4		5		
1	4	4		0.05
1	4	5		0.05
1	3	4		0.05
1	3	5		0.05
2	3	4		0.05
2	3	5		0.05
2	4	4		0.05
2	4	5		0.05

Table C.10: Sheet 4, substation 4 measurements

from_node	to_node	type	value	R_ii
1		1		
2		1		
3		2		
3		3		
5		2		
5		3		
7		2		
7		3		
4		4		
4		5		
6		4		
6		5		
8		4		
8		5		
1	3	4		0.05
1	3	5		0.05
1	5	4		0.05
1	5	5		0.05
1	7	4		0.05
1	7	5		0.05
3	4	4		0.05
3	4	5		0.05
5	6	4		0.05
5	6	5		0.05
7	8	4		0.05
7	8	5		0.05
2	4	4		0.05
2	4	5		0.05
2	6	4		0.05
2	6	5		0.05
2	8	4		0.05
2	8	5		0.05

Table C.11: Sheet 5, substation 5 measurements

from_node	to_node	type	value	R_ii
1		1		
2		1		
3		1		
4		1		
5		1		
6		1		
7		1		
4		2		
4		3		
7		2		
7		3		
3		4		
3		5		
5		4		
5		5		
6		4		
6		5		
1	3	4		0.05
1	3	5		0.05
1	4	4		0.05
1	4	5		0.05
1	5	4		0.05
1	5	5		0.05
1	6	4		0.05
1	6	5		0.05
1	7	4		0.05
1	7	5		0.05
2	3	4		0.05
2	3	5		0.05
2	4	4		0.05
2	4	5		0.05
2	5	4		0.05
2	5	5		0.05
2	6	4		0.05
2	6	5		0.05
2	7	4		0.05
2	7	5		0.05

Table C.12: Sheet 6, list of open CBs

Sub	From_node	To_node
1	5	6
2	1	3
3	1	3
4	5	6
4	7	8
5	2	3
5	2	4
5	2	5
5	2	6
5	2	7

C.3.2 Measurement case 2

File: meas_database_case2.xlsx

Table C.13: Sheet 1, substation 1 measurements

from_node	to_node	type	value	R_ii
1		1		
4		1		
2		2		
2		3		
5		4		
5		5		
1	2	4		0.05
1	2	5		0.05
2	3	4		0.05
2	3	5		0.05
3	4	4		0.05
3	4	5		0.05
4	6	4		0.05
4	6	5		0.05
5	6	4		0.05
5	6	5		0.05
1	5	4		0.05
1	5	5		0.05

Table C.14: Sheet 2, substation 2 measurements

from_node	to_node	type	value	R_ii
1		1		
4		1		
2		2		
2		3		
4		4		
4		5		
1	2	4		0.05
1	2	5		0.05
1	3	4		0.05
1	3	5		0.05
2	5	4		0.05
2	5	5		0.05
4	5	4		0.05
4	5	5		0.05
1	3	4		0.05
1	3	5		0.05
3	4	4		0.05
3	4	5		0.05
3	4	5		0.05

Table C.15: Sheet 3, substation 3 measurements

from_node	to_node	type	value	R_ii
1		1		
2		1		
1		2		
1		3		
2		2		
2		3		
4		4		
4		5		
1	4	4		0.05
1	4	5		0.05
1	3	4		0.05
1	3	5		0.05
2	3	4		0.05
2	3	5		0.05
2	4	4		0.05
2	4	5		0.05

Table C.16: Sheet 4, substation 4 measurements

from_node	to_node	type	value	R_ii
1		1		
2		1		
3		2		
3		3		
7		2		
7		3		
4		4		
4		5		
1	3	4		
1	3	5		
1	5	4		0.05
1	5	5		0.05
1	7	4		0.05
1	7	5		0.05
3	4	4		0.05
3	4	5		0.05
5	6	4		0.05
5	6	5		0.05
7	8	4		0.05
7	8	5		0.05
2	4	4		0.05
2	4	5		0.05
2	6	4		0.05
2	6	5		0.05
2	8	4		0.05
2	8	5		0.05

Table C.17: Sheet 5, substation 5 measurements

from_node	to_node	type	value	R_ii
1		1		
2		1		
4		2		
4		3		
7		2		
7		3		
5		4		
5		5		
1	3	4		0.05
1	3	5		0.05
1	4	4		0.05
1	4	5		0.05
1	5	4		0.05
1	5	5		0.05
1	6	4		0.05
1	6	5		0.05
1	7	4		0.05
1	7	5		0.05
2	3	4		0.05
2	3	5		0.05
2	4	4		0.05
2	4	5		0.05
2	5	4		0.05
2	5	5		0.05
2	6	4		0.05
2	6	5		0.05
2	7	4		0.05
2	7	5		0.05

Table C.18: Sheet 6, list of open CBs

Sub	From_node	To_node
1	5	6
2	1	3
3	1	3
4	5	6
4	7	8
5	2	3
5	2	4
5	2	5
5	2	6
5	2	7