USING GRAPH THEORY TO RESOLVE STATE ESTIMATOR ISSUES FACED BY DEREGULATED POWER SYSTEMS

A Dissertation

by

JIANSHENG LEI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2007

Major Subject: Electrical Engineering

USING GRAPH THEORY TO RESOLVE STATE ESTIMATOR ISSUES

FACED BY DEREGULATED POWER SYSTEMS

A Dissertation

by

JIANSHENG LEI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Garng M. Huang
Committee Members,	Chanan Singh
	Don R. Halverson
	Paul Nelson
Head of Department,	Costas N. Georghiades

May 2007

Major Subject: Electrical Engineering

ABSTRACT

Using Graph Theory to Resolve State Estimator Issues Faced by Deregulated Power Systems. (May 2007)

Jiansheng Lei, B.S., Tsinghua University, Beijing, China;

M.S., Tsinghua University, Beijing, China

Chair of Advisory Committee: Dr. Garng M. Huang

Power industry is undergoing a transition from the traditional regulated environment to the competitive power market. To have a reliable state estimator (SE) in the power market environment, two major challenges are emerging, i.e. to keep SE running reliably even under a contingency and to run SE over a grid with extremely large size.

The objective of this dissertation is to use graph theory to address the above two challenges.

To keep SE running reliably under a contingency, a novel topological approach is first proposed to identify critical measurements and examine network observability under a contingency. To advance the classical topological observability analysis, a new concept of contingency observability graph (COG) is introduced and it is proven that a power system network maintains its observability under a contingency if and only if its COG satisfies some conditions. As an application of COG, a two-stage heuristic topological approach is further developed based on the new concept of qualified COG (QCOG) to minimize the number of measurements and RTUs under the constraint that the system remains observable under any single contingency.

To overcome the disadvantages of existing SE over extremely large networks, a textured distributed state estimator (DSE), which consists of the off-line textured architecture design and the on-line textured computation, is proposed based on COG and a new concept of Bus Credibility Index (BCI). The textured DSE is non-recursive, asynchronous and avoids central controlling node. Numerical tests verify that the performance of the new textured DSE algorithm improves greatly compared with existing DSE algorithms in respect of bad data detection and identification. Furthermore, the software implementation for DSE is formulated as an information integration problem over regional power markets, and is very challenging because of its size and complexity. A new concept of semantic knowledge warehouse (SKW), together with the proposed concepts of semantic reasoning software component (SRSC) and deduction credibility, is developed to implement such an information integration system.

To my family

ACKNOWLEDGEMENTS

First of all, I would like to express my deep gratitude and intellectual indebtedness to the chair of my advisory committee, Dr. Garng M. Huang. I feel so fortunate to have him leading me through all my graduate school years. His energizing research spirit, unprecedented dedication to students, patient guidance, insightful comments, reflections, and feedback on my research were indispensable ingredients in my successful completion of the dissertation.

I would like to extend my heartfelt thanks to the members of my advisory committee - Dr. Chanan Singh, Dr. Don R. Halverson and Dr. Paul Nelson, for their time, constructive feedback and encouragement. I also would like to thank Dr. Ali Abur for serving as my committee member before 2005.

I appreciate the encouragement from my colleagues and friends in the past few years, in particular, Mr. Kun Men, Dr. Xingbin Yu, Dr. Yishan Li, Ms. Ping Yan, Dr. Nirmal-Kumar C Nair, Dr. Shan Zhong and Dr. Qifeng Ding, et al.

Last but not least, a special appreciation is directed to my mother who is my greatest source of motivation. She has always been there listening to my dreams, encouraging me to pursue them and sharing my joys unfolding them. My special gratitude also goes to my dear wife, Duan Zhang, for her love for me, her endurance with me, her true belief in me, and for her hard work all these years to support me and our family.

TABLE OF CONTENTS

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

1.1 The emerging challenges faced by state estimator in deregulated power grid

State estimation (SE) is an essential tool to monitor an electric power system [1]. The main functionality of a state estimator is to calculate a reliable estimate of the power system state vector consisting of bus voltage angles and magnitudes based on telemetric measurements distributed throughout the power grid [2, 3, 4]. The mathematical formulation of SE problem is given in Appendix A.

Power industry is undergoing a transition from the traditional regulated environment to the competitive power market. In order to have a reliable state estimator in the power market environment, there are two major emerging challenges as follows.

A. Challenge 1: Keep the state estimator running reliably even under a contingency

Here the contingency refers to a disturbance such as a branch outage. A loss of measurement is also considered as a contingency in this dissertation as it has a major impact on the state estimator.

As long as the power market meets a reasonable short-term reliability requirement, the power system will be running in a normal operating state when there is no contingency. When a contingency happens, the system may maintain its normal operating state or transfer to an alert operating state. And it is critical to keep the state estimator running at this time regardless of the contingency because:

1. The operators heavily rely on the successful execution of state estimator to get the accurate state of the current grid so that they can closely monitor the system and take appropriate preventive controls to make the system secure again.

This dissertation follows the style of *IEEE Transactions on Power Systems*.

2. Market related functions such as congestion management, determination of local marginal price (LMP) and available transfer capability computation need to continue working in spite of the occurred contingency; all these functions are heavily dependent of a reliable and accurate state estimation results. [5]

Therefore, it is necessary to design the state estimator in such a way that it can run reliably against a branch outage or a loss of measurement. The detailed technical issues related to this Challenge 1 are further discussed in section 1.2 and 1.3 as topic 1 and 2.

B. Challenge 2: Run state estimator over a grid with extremely large size

In the traditional regulated environment, the power grid is regionally divided and owned by a few locally monopolistic utilities. Though the power grid is physically connected as a whole, there are few if any financial connections between these utilities. Therefore, the utilities only run their own state estimators independently over their own part of the grid that is relatively small compared to the whole grid [6].

On the other hand, in power market environment, individual local utilities are releasing the operating authority to an Independent System Operator (ISO) or Regional Transmission Organization (RTO) that works with all stakeholders to ensure a reliable and nondiscriminatory operation of the whole power transmission system. These ISOs/RTOs will be responsible for a much larger power grid compared to their member utilities. The implementation of real-time state estimator functions is crucial for the proper independent system operation because operators will need to utilize the state estimation result to make and justify technical and economical decisions, such as congestion management and the procurement for adequate ancillary services, and to uncover potential operational problems related to voltage and transient stability. [7]

Moreover, a recent trend for these ISOs/RTOs is to further cooperate and run the power market on a bigger grid as a Mega-RTO for a better market efficiency. The grid size of a Mega-RTO becomes extremely large, as concluded recently by Federal Energy Regulatory Commission (FREC), that only four Mega-RTOs should cover the whole USA territory excluding Texas. Accordingly, the grid size will become extremely large. [8-11]

For example, there were totally 10 independent control centers in Texas for different utilities before the power deregulation and each control center had their individual state estimators to monitor their own part of the grid. On July 31, 2001, the existing 10 control areas were consolidated into a single control area managed by the Electric Reliability Council of Texas (ERCOT). That is to say, as the ISO for Texas power market, ERCOT needs to get the state of the whole Texas grid which is the sum of the grids of the previous ten individual state estimators. [8]

As a conclusion, it has been an overwhelming challenge to ensure the state estimator running smoothly over an extremely large grid. The detailed technical issues related to Challenge 2 are further discussed respectively in section 1.4 and 1.5 as topic 3 and 4.

The objective of this dissertation is to use graph theory to address the above two major challenges faced by state estimation in deregulated power systems. This chapter is organized as follows: Challenge 1 is addressed in Section 1.2 and 1.3, where Topic 1 and Topic 2 are respectively discussed. Next the Topic 3 and Topic 4 are separately described in Section 1.4 and 1.5 in order to meet Challenge 2. The organization of this dissertation and the notations are given in Sections 1.6 and 1.7.

1.2 Topic 1: Network observability examination under contingency

Due to the increasing concerns on the power grid security in a deregulated power market as discussed in Challenge 1, a reliable state estimator to withstand disturbances, such as losses of a measurement or branch, becomes much more important [12, 13].

A network is said to be observable if it is possible for the state estimator on it to determine the bus voltage magnitudes and angles throughout the entire network from the installed measurements [4, 14]. It is clear that network needs to retain its observability under a contingency in order to meet Challenge 1.

Based on the Jacobian matrix manipulation, some numerical approaches [12, 13] are developed to examine the network observability under a contingency.

On the other hand, so far the topological approaches [4, 15-18] mainly focus on the network observability when there is either no contingency or only measurement loss; and the topological observability analysis under a branch outage is not addressed yet.

Therefore, we propose a graphical and topological approach to examine whether the network retains its observability under a contingency, which has the following new features:

1. Advancing the current topological approaches, we examine the network observability under a branch outage besides the critical measurement identification.

2. We introduce a new concept of contingency observability graph (*COG*) and theoretically prove that a power system network maintains its observability under a contingency if and only if its *COG* satisfies some conditions.

3. The proposed approach can serve as a natural extension of the classical topological observability analysis with little extra computation load to determine the *COG*, which is an important advantage for online applications.

Many insights in this topic can be used to design and upgrade a measurement system to maintain the network observability against a contingency, which is discussed in the coming topic.

1.3 Topic 2: Measurements design considering observability under contingency

Regardless of the algorithm applied in a state estimator over a network, the measurement system need to be carefully designed [19, 20, 21] so that the network can maintain its observability against a contingency to meet Challenge 1.

Measurement placement design has been addressed in many literatures [12, 13, 22-27], and the problem is formulated as an optimization problem, whose objective is to minimize the installation cost of measurements while satisfying some performance requirements as constraints. State estimation accuracy is a main constraint in some studies [24-27]; while the system observability under some single branch outages or measurement losses is the constraints in others [12, 13]. In addition, the approaches in [12, 13] are numerical algorithms while the topological approach has not been developed to address the system observability maintenance under a branch outage.

The remote terminal units (RTUs) placement is not considered in the above measurement designs. However, RTUs actually play a critical role in state estimator. Measurement devices measure power injection, power flow and bus voltage magnitude, and RTUs' communication channels transfer the measured raw data from a substation to the control centers for further processing [28]. Accordingly, both RTUs and measurement devices are necessary for a measurement system. Therefore, the conventional approach needs to be refined to include the RTUs costs as part of the installation cost, especially since the RTU cost is significantly large in practice.

The placement design considering both measurements and RTUs has been addressed in [29, 30], and the problem is formulated as an optimization problem, whose objective is to minimize the number of equipments or the installation cost while satisfying some performance constraints. Note that the performance constraints in [29, 30] are limited to measurement or RTU losses while branch outage is not considered.

A novel two-stage measurement and RTU placement design approach is proposed in this thesis with the following three features:

1. The objective is to minimize the number of both measurements and RTUs with the constraints that the system maintains its observability against a contingency. In addition to measurement installation cost, we also include the RTUs costs as part of the installation cost.

2. Different from the numerical approach based on measurement Jacobian matrix and integer programming technique [12, 13], we propose a topological heuristic two-step approach that maintains system observability under loss of certain lines or measurements. The new concepts of Qualified COG (QCOG) and preferred QCOG are defined and play a critical role in our methodology.

3. A single RTU loss is also included as a contingency. Since all measurements at a substation can have their signals fed to a common RTU for transmission to the control center, the failure of the RTU would make all the measurements that it serves unavailable, which will have a major impact during the reliability evaluation for a measurement system [28-30].

1.4 Topic 3: State estimator for extremely large grid

Due to Challenge 2, there are two choices to run a state estimator over an extremely large grid in power market: we can either set up a brand new state estimator over the whole system or try to implement a distributed state estimation algorithm that takes advantages of the existing local state estimators from the member utilities. Details of the two choices are given as follows.

A. One state estimator scheme

One state estimator (OSE) can be set up to run over the whole system. However, OSE approach has the following disadvantages:

1. The investment on the new estimator can be enormous. The maintenance cost over such a huge area is also high.

2. The system size is extremely large, which raises the scalability issue. The system matrix becomes more ill-conditioned [31], and the computation speed and convergence performance becomes slower and poorer. The scalability issue becomes even more severe for a Mega-RTO.

3. The existing local state estimators distributed in different entities are wasted.

Because of the above disadvantages of OSE, many distributed state estimator schemes are proposed.

B. Distributed state estimation algorithms

We assume that multiple entities such as member companies, ISOs and RTOs are physically connected and cooperate to run the whole system as Fig.1.1. Accordingly, there are multiple existing estimators distributed in the subsystems like Company A, ISO B and RTO C. And every entity will maintain and execute their local state estimator on their own areas. These entities are connected through tie lines near the boundary buses.

With the development of Information Technology (IT), DSE algorithms, especially those without central controlling node [32, 33], become more and more applicable.

The main drawback of the existing DSE algorithms [32-42] is that bad data detection and identification ability decreases dramatically compared to OSE who is over the whole system, especially when bad data is close to the boundary of individual estimators. Moreover, the estimation accuracy on boundary buses in DSE is much lower than that in OSE, and there can be significant discrepancy around boundary buses that makes the whole result inconsistent.

A new concurrent non-recursive textured distributed state estimator (DSE) is developed in this thesis as an alternative to determine the state of whole grid. The textured DSE consists of the off-line textured architecture design and the on-line textured computation. After the off-line textured architecture design determines the texture, the on-line computation relies on the raw data or estimated value exchanges among the neighboring entities through the communications in the textured architecture, fully takes advantage of the existing local state estimators and ultimately determines the system state over the whole system.

Concurrent textured algorithm has been well developed to deal with the optimization problem of power systems by our team led by Dr. Huang [43-48]. The basic idea of a textured algorithm is as follows. First, the problem on a large system is decomposed into several smaller and more tractable sub-problems for concurrent computation by fixing some boundary variables. Then by rotating the fixed variables, a recursive sequence of concurrent sub-problems are solved and the original high dimensional problem is solved by divide-and-conquer. The origin of term 'texture' is because there are overlapping areas between the neighboring sub-systems, which are just like texture. And the boundary variables are located on these overlapping areas.

Since the textured control and optimization methodology have been developed in [43-48] by Dr. Huang, it is only natural to further develop a textured state estimator as the monitoring component. Such a distributed textured state estimation algorithm will

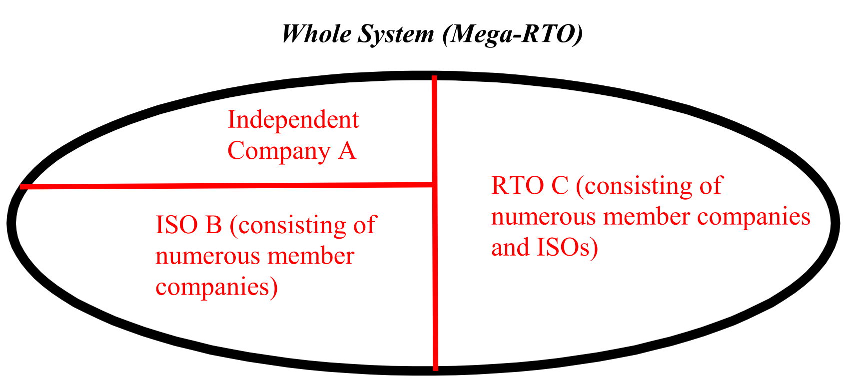


Fig.1.1 Multiple Companies/ISOs/RTOs physically connected

overcome the disadvantages of OSE. In addition, compared with existing DSE algorithm, the textured DSE improves greatly on bad data detection, identification ability and decreasing the discrepancy around boundary buses.

1.5 Topic 4: Information integration

To implement the textured DSE as discussed in Topic 3, the on-line computation relies on the raw data or estimated value exchanges among neighboring entities in the textured architecture, which is a special form of information exchanges and integration in terms of Information Technology.

Note that the information integration is a general topic in software engineering and its application in power market is not limited to SE problem. Therefore, we extend our study on this topic for a more general power market.

In the traditionally regulated environment, the power system is owned by a few locally monopolistic organizations and there is almost no need to exchange data with other organizations since the utilities provide bundled services to customers and perform both power network and marketing functions.

By contrast, in a deregulated environment, monopolistic utilities are broken up into competitive entities. These market participants cooperate to run the system. On the other hand, each individual entity tries to maximize its own profits in the power market. In addition, consumers in the market have the choice to choose their power suppliers based on cost-effectiveness and reliability. Thus the market participants consist of generation plants, utilities, large commercial consumers, transmission companies, for-profit or nonprofit agents and organizations, and possibly the individual resident customers.

To run such a market smoothly and efficiently, participants need to share the information from diversified sources such as accounting departments, operation control centers and logistics departments.

Information integration for such a system has become very challenging [49-55]. Common Information Model (CIM) is utilized for data exchange [53]. How to transform the legacy Energy Management System (EMS) system to the new environment has also been studied [49, 55]. However, even though some work has been done to extend the

CIM for a better utilization in information integration, the efforts on the information integration of power market are still very limited.

Federal Energy Regulatory Commission (FERC) orders in 2001 that only four Regional Transmission Organizations (mega-RTOs) should cover the whole USA territory excluding Texas [9-11]. Accordingly, the number of market participants in a single market will increase further dramatically, which leads to the following emerging major concerns in terms of information integration:

Concern 1: How to efficiently manage and integrate the numerous distributed information sources? For example, the electric power flow data and the financial transaction data are logically related. However, the power flow data is from Energy Management System (EMS) while the financial transaction data are from the accounting departments. Consequently, the information from different sources is independently stored in various software applications and formats and therefore is difficult to be integrated.

Note that the information integration is not to simply share the information among all market participants; instead, it will follow some rules from the market protocol and determine whether the information is accessible based on many factors such as the information content (financial or EMS data), the entity to request the information (the market participant or the independent system operator), the request time (day ahead, real-time or day after), etc.

Concern 2: How to adapt to the changing market environment? Note that the market rules and the information remain changing from time to time as the markets evolve. The number of market participants and information sources also changes dynamically.

Concern 3: How to improve the software reusability and expandability? Many existing software applications are designed for one specific power application and are difficult to be reused in other applications, even though these applications are logically related. Furthermore, when more and more information sources are integrated, these software applications are no longer suitable due to the expanded size and complexity.

In order to meet the concerns, a semantic based software architecture is proposed in this thesis with the following features:

Feature 1: Information sources are integrated into a regionally regulated semantic knowledge warehouse (SKW). In a power market over a particular region, either a for-profit or nonprofit organization may initiate such a global SKW. It can also be part of the market protocol design and therefore regulated by ISO/RTO.

Feature 2: Based on the above SKW, a semantic reasoning software component (SRSC) is developed to automatically deduct new information based on a deductive reasoning algorithm. Note that the two independent terms 'semantic' and 'software component' are for the information reusability and software reusability respectively in computer science. In this thesis these two concepts are further combined as SRSC to act as a virtual reasoning machine. Without any code change, SRSC is able to be applicable for different knowledge domains; and it intends for reusability in a higher level: the intelligence reusability.

Feature 3: A concept of deduction credibility index is developed to further refine the SRSC.

Note that the system integration is very challenging since in practice a management system has to be implemented over a large-scale complicated energy market. In order to reduce the software integration risk, service-oriented architecture (SOA), a well developed concept in computer science, has been introduced into power industry recently [56-61]. Service is a software component that is a business-complete logical unit of work with a generic interface protocol and self-contained extensible messages, while SOA aims to design the whole system with loosely coupled services and service consumers (clients) [56, 57]. SOA describes a distinct style of software integration and is not tied to a specific technology. For example, SOA can be implemented using Common Object Request Broker Architecture (CORBA), Distributed Component Object Model (DCOM), Web Services, etc. As a real-world experience, SOA has been applied successfully in the Market Redesign and Technology Upgrade project at CAISO and played a critical role throughout the integration design [58, 59]. SOA is also utilized as the architecture style in a proposed Web Services based SCADA system [60] and an ongoing open-source development project for power system simulation [61].

The usage of the semantic architecture for service definitions in SOA leads to the creation of semantic SOA that provides significantly enhanced interoperability and expandability between services [62]: numerous services are based on the data model derived from a unified semantic knowledge warehouse and new services can be generated via reasoning. Details of semantic SOA will be discussed in a future research.

1.6 Objectives and organization of the dissertation

The objective of this dissertation is to address the Challenge 1 and 2 faced by state estimators in deregulated power systems. The detailed technical issues related to the two challenges are further discussed as Topic 1, 2, 3 and 4 that are studied in Chapter II, III, IV and V respectively as follows:

In Chapter II the topology analysis for state estimator is discussed and a new concept of contingency observability graph (COG) is defined and applied as a novel topological approach to examine system observability under a measurement loss or branch outage.

As an application of COG, a heuristic topological approach for measurements and RTUs placement design is developed in Chapter III based on the new concepts of qualified COG (QCOG) and preferred QCOG.

Taking advantage of the COG and new concepts of Bus Credibility Index (BCI), a textured distributed state estimator (DSE) that consists of the off-line textured architecture design and the on-line textured computation is proposed in Chapter IV for large-scale power systems such as Mega Regional Transmission Organizations (Mega-RTOs).

In Chapter V the software implementation for data exchange is proposed as an information integration problem over regional power markets, and is very challenging due to its size and information complexity. A new concept of semantic knowledge warehouse (SKW) is proposed as a semantic software architecture for the power market information integration problem.

Finally Chapter VI provides a summary of the dissertation.

1.7 Notations

The notations are summarized as follows:

(G, M): network G with a measurement set M;

b_i : bus i;

b_ib_j : branch br from bus i to bus j;

B-G: the set $\{b_i\}$ consisting of buses in G;

BR-G: the set $\{br_i\}$ consisting of branches in G;

m-b_ib_j : the measurement located near bus i and measuring the real and reactive power flow along branch b_ib_j;

m-b_i : the measurement located on bus i and measuring the real and reactive power net injections into bus i;

M_f: the set $\{m\text{-}b_ib_j\}$ consisting of power flow measurements in M ;

M_i : the set $\{m\text{-}b_i\}$ consisting of power injection measurements in M.

Based on the above notations we have $M=M_f \cup M_i$ and $M_f \cap M_i=\Phi$, when voltage magnitude measurements are excluded.

CHAPTER II

A NOVEL TOPOLOGICAL APPROACH TO EXAMINE NETWORK OBSERVABILITY UNDER A CONTINGENCY

2.1 Introduction

As discussed in Topic 1 of Chapter I, a reliable state estimator to withstand a contingency, such as a measurement loss or a branch outage, becomes more and more important due to the increasing concerns on the network security in a deregulated power market.

A topological approach to examine the network observability under a contingency is presented in this chapter as follows: first the problem is formulated in Section 2. Then the new concept of *COG* and its properties are introduced in Section 3. Details on applying COG to network observability analysis against a contingency are discussed in Section 4. The demonstrating example on the IEEE 30-bus system is given in Section 5. Then we conclude in the last section. Note that some proofs of theorems are given in the Appendix B.

2.2 Problem formulation

This section is organized as follows: the network observability without contingency consideration is discussed first in subsection A. Then some preliminary concepts are introduced in subsection B. Finally the network observability under a single measurement loss, a single branch outage and a contingency are studied respectively in subsection C, D and E.

A. Network observability without contingency consideration

To determine if a network G with a measurement set M is observable without worrying about contingency, the classical topological observability analysis has been well developed [4] based on the following concept of *measurement assignment*:

Definition1: For (G, M), in which T is a tree of G and $BR\text{-}T$ is the set of branches in T, we have the following definitions:

1) Measurement assignment A is a function defined as follows:

i. A maps a branch $b_ib_j \in BR\text{-}T$ to a measurement $m \in M$, denoted as $A(b_ib_j)=m$.

ii. $A(b_ib_j)$ is either a power flow measurement on branch b_ib_j or a power injection measurement at an end bus of the branch b_ib_j, i.e. $A(b_ib_j) \in \{ m\text{-}b_ib_j, m\text{-}b_jb_i, m\text{-}b_i, m\text{-}b_j\}$.

iii. A is a one-to-one mapping, i.e., if $br_i \neq br_j$, then $A(br_i) \neq A(br_j)$.

2) Branches belonging to T are termed as *tree branches*, and all the remaining branches are termed as *co-tree branches* [4], denoted as $BR\text{-}T$={tree branches} and $BR\text{-}\overline{T}$={co-tree branches} respectively.

3) Measurements belonging to the range of $A(BR\text{-}T)$ are termed as *essential measurements*, and all the remaining measurements in M are termed as *nonessential measurements* [12]. Essential and nonessential measurements are denoted as M_e and M_n respectively, and we have $M = M_e \bigcup M_n$ and $M_e \bigcap M_n = \varnothing$.

With the above definitions, a measurement assignment is further denoted as $A{:}BR\text{-}T \rightarrow M_e$.

Based on Definition 1, the following theorem is given in [4]:

Theorem 1: (G, M) is observable if and only if there exists a valid measurement assignment $A{:}BR\text{-}T \rightarrow M_e$ where T is a spanning tree of G.

Measurement m is termed as a *critical* measurement if G becomes unobservable after the removal of m (but there is no branch outage); otherwise m is termed as a *redundant* measurement [15]. Critical and redundant measurements are denoted as M_c and M_r respectively, and we have $M = M_c \cup M_r$ and $M_c \cap M_r = \varnothing$. It is clear that classification of M_e and M_n depends on a specific $A{:}BR\text{-}T \rightarrow M_e$ while classification of M_c and M_r does not.

Here we assume there are enough voltage magnitude measurements and we only focus on the active and reactive power measurements.

Based on Theorem 1, a practical topological approach has been developed to determine a valid $A{:}BR\text{-}T \rightarrow M_e$ for a given (G, M) [4] and widely applied in commercial software packages.

To advance the network observability analysis toward contingency consideration, we introduce the following concepts.

B. Some preliminary concepts

First, to tell the differences between a measurement and its associated physical branches, we define the concepts of *topological related branch* concept that is solely determined by the configuration of (G, M) and *mapped branch* concept that is determined by the measurement-branch mapping. Details are as follows.

Definition 2: For a given observable (G, M), measurement m's *topologically related branch* $TB(m)$ is defined as a set of branches as follows:

For a power flow measurement m, say $m\text{-}b_ib_j \in M_f$, $TB(m\text{-}b_ib_j)=\{b_ib_j\}$; for a power injection measurement m , say $m\text{-}b_i \in M_i$, $TB(m\text{-}b_i)=\{$all branches incident to bus $b_i\}=\{b_ib_l \mid l=l_1, l_2, \ldots\}$.

Definition 3: For an $A{:}BR\text{-}T\rightarrow M_e$ of a given observable (G, M) where T is a spanning tree of G,

1) Suppose $A(br_i)=m_i$ where $br_i \in BR\text{-}T$ and $m_i \in M_e$, then m_i is termed as br_i's *A-mapped measurement*.

2) Since A is a one-to-one and onto function from $BR\text{-}T$ to M_e, there exists an inverse mapping $A^{-1}(m_i)=br_i$ from $m_i \in M_e$ to $br_i \in BR\text{-}T$. Accordingly, br_i is termed as m_i's *A^{-1}-mapped branch* and $A^{-1}(M_e)= BR\text{-}T$.

We have the following observations based on the above definitions:

Observation 1 [15]: A critical measurement is an essential measurement for any $A: BR-T \rightarrow M_e$, i.e. $M_c \subseteq M_e$. In other words, a nonessential measurement is redundant, i.e. $M_n \subseteq M_r$.

Observation 2 [15]: An essential measurement in an $A{:}BR\text{-}T\rightarrow M_e$ may be redundant, and a redundant measurement may be essential in an $A{:}BR\text{-}T\rightarrow M_e$.

Observation 3: Given the condition that M_r is non-empty, some redundant measurements are nonessential in an $A{:}BR\text{-}T\rightarrow M_e$.

Proof: Suppose not, all redundant measurements are essential. Then together with Observation 1, this implies all measurements are essential in $A{:}BR\text{-}T\rightarrow M_e$, which only happens when every measurement is critical and contradicts to the assumption that M_r is non-empty.

Observation 4: All measurements have their topologically related branches. On the other hand, only essential measurements have their defined A^{-1}-mapped branches in an $A{:}BR\text{-}T \rightarrow M_e$.

Observation 5: $A^{-1}(m) \in TB(m)$, where $m \in M_e$.

Proof: Essential measurement m is either a power flow measurement $m\text{-}b_ib_j \in M_f$ or a power injection measurement $m\text{-}b_i \in M_i$, i.e. $m \in M_e \subset M = M_f \cup M_i$.

If $m=m\text{-}b_ib_j$, then $A^{-1}(m\text{-}b_ib_j)=b_ib_j \in \{b_ib_j\}=TB(m\text{-}b_ib_j)$; otherwise $m=m\text{-}b_i$, then $A^{-1}(m-b_i)=b_ib_l \in \{b_ib_l| \, l=l_1, l_2, \ldots\}=TB(m\text{-}b_i)$. In either case, $A^{-1}(m) \in TB(m)$ holds.

Observation 6: The valid measurement assignment $A{:}BR\text{-}T \rightarrow M_e$ for a given observable (G, M) is not necessarily unique [15]. Therefore, a branch can be a tree branch in $A_1{:}BR\text{-}T_1 \rightarrow M_{e1}$ but is a co-tree branch in another valid measurement assignment $A_2{:}BR\text{-}T_2 \rightarrow M_{e2}$.

Observation 7: The classification of critical and redundant measurements is unique and independent of $A{:}BR\text{-}T \rightarrow M_e$, but the classification of essential and nonessential measurements depends on the choice of $A{:}BR\text{-}T \rightarrow M_e$ [15].

A spanning tree contains no loop. Therefore, a loop of graph contains at least one co-tree branch. On the other hand, a sub-graph consisting of a spanning tree plus a co-tree branch contains one and only one loop, which leads to the concept of a basic loop:

Definition 4: Given a spanning tree T of a graph G, a *basic loop* L is defined as the loop containing one and only one co-tree branch.

Note that a basic loop L is T dependent; and there is also a one-to-one and onto mapping between co-tree branches and the basic loops for a given T. In this dissertation, notation $L(T, br_{co\text{-}tree,T})$ stands for the basic loop L which contains co-tree branch $br_{co\text{-}tree}$ with the given spanning tree T.

C. Network observability under a single measurement loss

The network maintains its observability against any single measurement loss if and only if there is no critical measurement. To examine the network observability under a single measurement loss, an algorithm to identify the critical measurements is necessary.

D. Network observability under a branch outage

To extend the network observability analysis toward a branch outage, two preliminary concepts are proposed as follows:

Definition5: For a connected graph *G*, a branch *br* is termed as a *radial* branch iff the removal of *br* splits *G* into two isolated networks or buses.

Definition6: For a given observable (*G*, *M*), a branch *br* is termed as a *removable* branch iff *G* remains observable after the removal of *br*.

As a consequence, *br* is removable only if either of the following two cases occurs:

1) *br* is radial and removing *br* splits *G* into two isolated observable networks/buses;

2) *br* is non-radial (thus *G* is not split) and *G* remains observable after the removal of *br*.

Thus, the network maintains its observability against any single branch outage if and only if every branch is removable.

Observation 8: A radial branch *br* in *G* is removable for an observable (*G*, *M*).

Proof: Since *br* is radial, *G* will be split into two isolated networks, say G_1 and G_2, after *br* is removed. In addition, the power flow measurement *m-br* may be lost if there is one.

Since (*G*, *M*) is observable, there exists a valid measurement assignment $A{:}BR\text{-}T \rightarrow M_e$, where *T* is a spanning tree of *G* and therefore is a connected subgraph covering all buses in both G_1 and G_2. Then *T* must contain *br*, the only branch connecting G_1 and G_2. After the removal of *br*, *T* is split into two isolated trees T_1 and T_2, which are spanning trees of G_1 and G_2 respectively. We use the same measurement mapping function *A* except that A_1 and A_2 are restricted within trees T_1 and T_2 respectively. Clearly $A_1{:}BR\text{-}T_1 \rightarrow M_{e1}$ and $A_2{:}BR\text{-}T_2 \rightarrow M_{e2}$ are valid measurement assignment for G_1 and G_2, which proves both G_1 and G_2 are observable. Then by Definition 6, *br* is removable.

Observation 9: The concept of critical measurement is developed without considering branch outages, while our cases may have a simultaneous loss of a branch and the associated measurement. For example, *G* is unobservable after the loss of a critical measurement *m-br*. However, suppose *br* is radial and both *m-br* and *br* are lost, *G* remains observable since the two isolated islands are both observable after removing br. Therefore, *m-br* is not a critical measurement any more when there is a branch outage on *br*. In other words, the underlying assumption for the definition of critical measurement is based on a particular network topology where there is no branch outage. When a branch outage does occur, the topology is changed and a critical measurement may become non-critical regarding the new network configuration.

E. Network observability under a contingency

To examine the problem of network observability against a contingency (either a single measurement loss or single branch outage), we can decompose the problem into two sub-problems, i.e., the network observability under a measurement loss and the observability under a branch outage.

On the other hand, since the two sub-problems are tightly coupled, one can accelerate the process by solving the problem all together, which motivates our *COG* based approach presented in the coming section.

2.3 Contingency Observability Graph (COG)

This section is organized as follows: First the new concept of contingency observability graph (COG) is defined in subsection A. Then a construction algorithm of COG is discussed in subsection B. Note that the critical measurements regarding the original topology without any branch outage are also identified during the construction of COG. Next some properties of COG are discussed in subsection C. Finally the uniqueness of COG is briefly studied in subsection D.

A. Definition of COG

A measurement m_i in a given observable (*G*, *M*) is either critical or redundant. According to Observation 1 and 4, a critical m_i always has its *A*-mapped branch for any $A{:}BR\text{-}T \rightarrow M_e$ and that branch can be used for our observability studies. On the other hand, according to Observation 3 and 4, the A^{-1}-mapped branch of a redundant m_i does not necessarily exist. However, from Observation 4 the topologically related branches of m_i always exist and can be used for our studies.

Based on the above analysis, we define *contingency observability graph* (*COG*) for an $A{:}BR\text{-}T \rightarrow M_e$ of an observable (G, M) as a graph consisting of the A^{-1}-mapped branches of the critical measurements and the topologically related branches of the

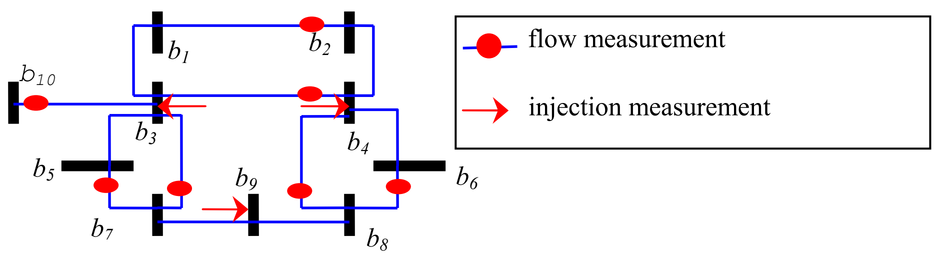


Fig. 2.1 The given (*G*. *M*)

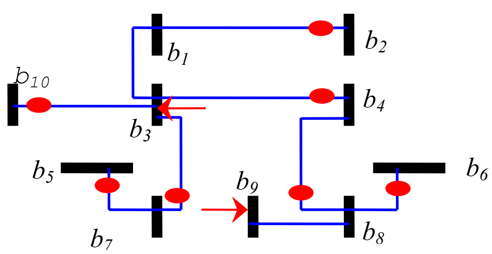


Fig. 2.2 Spanning tree and essential measurements in Fig.2.1 for $A{:}BR\text{-}T \rightarrow M_e$

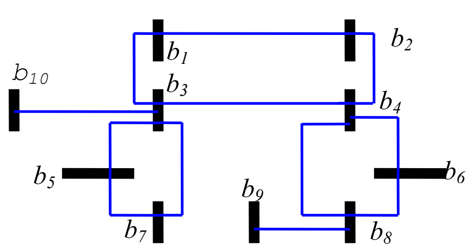


Fig.2.3 *COG* for $A{:}BR\text{-}T \rightarrow M_e$ in Fig.2.2 of the given (*G*. *M*) in Fig.2.1

redundant measurements together with the associated buses, i.e. $BR-COG = A^{-1}(M_c) \bigcup TB(M_r)$, where *BR-COG* is the set of branches in *COG*.

An illustrative example of *COG* is given as follows:

In Fig.2.1, (*G*, *M*) represents the network *G* with a given measurement set *M*. Suppose we know that m-b_9 is the only critical measurement, whose identification will be discussed in detail later on. Measurement m-b_9 is A^{-1}-mapped to branch b_8b_9 by $A: BR-T \to M_e$, where $A{:}BR\text{-}T \rightarrow M_e$ is found as shown in Fig.2.2 via the classical topological observability analysis. Then b_8b_9 is part of *COG*. As for the other redundant measurements, their topologically related branches are part of *COG*. Accordingly, *COG* is obtained as shown in Fig.2.3.

Note that branch b_7b_9 in *G* does not belong to the *COG*. Though b_7b_9 is one of m-b_9's topologically related branches, only m-b_9's A^{-1}-mapped branch, i.e. b_8b_9, is added into *COG* because m-b_9 is critical measurement.

B. Construction of COG and determination of critical measurement

After the classical topological observability analysis on an observable (*G*, *M*), an $A{:}BR\text{-}T \rightarrow M_e$ is found. Accordingly, the tree branches set *BR-T*, the co-tree branches set $BR\text{-}\bar{T}$, the essential measurements set M_e and the nonessential measurements set M_n are all determined respectively. Note that the critical measurements are not identified yet.

Then the *COG* for the $A{:}BR\text{-}T \rightarrow M_e$ of the (*G*, *M*) is constructed via the following algorithm:

Stage1: Label all nonessential measurements $m_n \in M_n$ as '*redundant*', and label all essential measurements $m_e \in M_e$ as '*unknown*' temporarily. Note that the measurements being labeled as '*unknown*' will be re-labeled as either '*redundant*' or '*critical*' later on.

Stage2: For all measurements being labeled as '*unknown*', add their A^{-1}-mapped branches together with the associated buses into *COG*. Note that the set of measurements being labeled as '*unknown*' equals to M_e at this time; therefore, Stage2 actually initializes *COG* as *T* because $A^{-1}(M_e)=BR\text{-}T$.

Note that the spanning tree *T* of *G* is always a subgraph and a spanning tree of *COG*, which is the Property 1 of COG and proved in the appendix B.3.

Stage3: For all measurements being labeled as '*redundant*', add their topologically related branches to *COG* if they are not in *COG* yet.

Stage4: Initialize iteration number i=1 and $M_{u,i}$ is defined as the set of measurements which are being labeled as '*unknown*' in iteration i. Obviously $M_{u,1}=M_e$.

Stage5: For every measurement m_u in $M_{u,i}$, check whether one of the following two conditions is satisfied:

Condition 1: $A^{-1}(m_u)$ is in a basic loop $L(T, br_{co\text{-}tree})$ of the existing *COG*, where $br_{co-tree} \in COG$ and $br_{co\text{-}tree} \in BR\text{-}\bar{T}$.

Condition 2: There exists another measurement m_2 which satisfies $m_2 \notin M_{u,i}$ and $A^{-1}(m_u) \subset TB(m_2)$.

If yes, re-label m_u as '*redundant*' and add $TB(m_u)$ into *COG* if they have not been added into *COG* yet.

Stage6: Set iteration number i=i+1 and determine the current $M_{u,i}$. If $M_{u,i} \neq M_{u,i-1}$, go back to Stage5; otherwise, re-label all measurements still in $M_{u,i}$ as '*critical*'. Then all the measurements are labeled as either '*critical*' or '*redundant*', and the construction process of *COG* ends.

The above algorithm constructs a graph that consists of topologically related branches of measurements labeled as '*redundant*' and the A^{-1}-mapped branches of the measurements labeled as '*critical*' together with the associated buses.

The constructed graph is confirmed to be the *COG* for the $A{:}BR\text{-}T \rightarrow M_e$ of the (G, M) due to the following theorem.

Theorem 2: Measurement's redundant or critical property matches with '*redundant*' or '*critical*' labeling in the above construction algorithm.

Theorem 2 is proven in the appendix B.2.

Based on Theorem 2 and the COG construction algorithm, it is clear that an essential measurement is redundant as long as its A^{-1}-mapped branch is in a loop of COG.

In the illustrative example also added the following:

As an illustrative example, the *COG* associated with the $A{:}BR\text{-}T \rightarrow M_e$ in Fig.2.2 of the (G, M) in Fig.2.1 is constructed as follows:

Stage1: Since M_n={m-b_4}, label m-b_4 as '*redundant*' and label all the other measurements as '*unknown*' temporarily.

Stage2: Initialize *COG* as *T*.

Stage3: Since m-b_4 is labeled as '*redundant*', add *TB*(m-b_4)={b_2b_4, b_3b_4, b_6b_4, b_8b_4} to *COG*. Note that b_3b_4 and b_8b_4 have been already included in *COG* in Stage1; only b_2b_4 and b_6b_4 are newly added to *COG*.

Stage4: Initialize iteration number i=1. Then the iteration process of Stage 5 is summarized in Table 2.1. For example, m-b2b1 is labeled as 'redundant' in Iteration 1 because A^{-1}(m-b_2b_1) is in a basic loop $L(T, b_2b_4)$ of *COG*., where b_2b_4 is added into *COG* in Stage 3.

TABLE 2.1 THE ITERATION PROCESS OF STAGE 5 FOR FIG.2.1

i	$M_{u,i}$	branch added to *COG*
1	{m-b_2b_1, m-b_4b_3, m-b_3, m-b_8b_4, m-b_6b_8, m-b_7b_3, m-b_5b_7, m-$b_{10}b_3$, m-b_9}	b_3b_5
2	{m-b_7b_3, m-b_5b_7, m-$b_{10}b_3$, m-b_9}	None
3	{m-b_9}	None
4	Iteration ends since $M_{u,4}$ =$M_{u,3}$	

Since $M_{u,4}$=$M_{u,3}$, we re-label all measurements in $M_{u,4}$, i.e. m-b_9, as '*critical*' and complete the construction of *COG*. Then the *COG* is finalized as shown in Fig.2.3.

Note that the critical measurements are identified after the construction of *COG* based on Theorem 2. For example, m-b_9 is the only measurement being labeled and identified as '*critical*' in the above example.

The above construction algorithm can be easily implemented in large-scale systems with fast computation speed because:

1. As a topological approach, there is no floating-point computation during the construction process.

2. The iteration process of Stage 5 ends when $M_{u,i}$=$M_{u,i-1}$. Since there are (n-1) essential measurements for an n-bus system, there are (n-1) measurements in $M_{u,1}$.

Furthermore, the number of measurements in $M_{u,i}$ is strictly decreasing until the end of the iteration process, which indicates that the number of iterations is less than (*n-1*).

Note that when G consists of a single branch loop without any radial branch, the number of iterations is only 1, which is the lower bound of the iteration numbers. On the other hand, when G consists of only radial branches in series with neither forking nor branching loops, the iteration number can be (n-1), which is the upper bound of iteration numbers. Since the real power networks are always with many loops and relatively small number of radial branches, the actual iteration number will be much less than the upper bound (n-1), where n is the number of buses. For example, the iteration number in the above example is 4, much less than the upper bound 9.

3. Basic loops used in Stage5 are easy to be found, stored and processed for computation because there exists a one-to-one mapping between a basic loop and a co-tree branch.

C. Properties of COG

For an $A{:}BR\text{-}T \rightarrow M_e$ of an observable (*G*, *M*), the associated *COG* has the following properties.

Property 1: *COG* contains *T*, and *T* is a spanning tree of *COG*.

For example, for the *COG* in Fig.2.3 and *T* in Fig.2.2, obviously *T* is a spanning tree of *COG* and *COG* contains *T*.

Property 1 is useful in Stage2 of the *COG* construction algorithm.

Proof of Property 1 is given in Appendix B.3.

Property 2: A branch *br* is removable if one of the following cases holds:

Case1: *br* is a radial branch in *G*;

Case2: *br* is non-radial in *G* and is not part of *COG*;

Case3: *br* is non-radial in both *G* and *COG*.

Proof of Property2 is given in Appendix B.3.

For example, for the (*G*, *M*) in Fig.2.1 and the associated *COG* in Fig.2.3, every branch except b_8b_9 is removable based on property 2.

Property 2 gives a sufficient condition for a branch to be removable; however, it is not a necessary condition. Accordingly, b_8b_9 in the above example may still be removable.

For any branch *br* in *G*, there are only four possibilities: 1) *br* is radial in *G*; 2) *br* is non-radial in G and not part of *COG*; 3) *br* is non-radial in both *G* and *COG*; 4) *br* is non-radial in *G* while radial in *COG*. Therefore, the following property immediately follows from Property2.

Property 3: A branch *br* in *G* is not removable only if *br* is non-radial in *G* while radial in *COG*.

Note that Property 3 tells the necessary condition for a branch to be non-removable.

Property 4: When all injection measurements are redundant, a branch *br* in *G* is not removable if and only if *br* is non-radial in *G* while radial in *COG*.

Proof of Property 4 is given in Appendix B.3.

Note that Property 4 tells the sufficient and necessary condition for a branch to be non-removable when all injection measurements are redundant.

D. Uniqueness of COG

Note that the *COG* associated with an $A{:}BR\text{-}T \rightarrow M_e$ of a given (*G*, *M*) is unique due to the definition of *COG*. However, it is possible to have different measurement assignments $A_1{:}BR\text{-}T_1 \rightarrow M_{e1}$ and $A_2{:}BR\text{-}T_2 \rightarrow M_{e2}$ for the same (*G*, *M*). Accordingly, the question whether the COG_1 for $A_1{:}BR\text{-}T_1 \rightarrow M_{e1}$ and COG_2 for $A_2{:}BR\text{-}T_2 \rightarrow M_{e2}$ are the same arises and is discussed as follows:

First, for a redundant measurement m_r, its topologically related branches are independent of measurement assignment and therefore are the same for $A_1{:}BR\text{-}T_1 \rightarrow M_{e1}$ and $A_2{:}BR\text{-}T_2 \rightarrow M_{e2}$.

Second, for a critical measurement m_c, m_c must be an essential measurement in both $A_1{:}BR\text{-}T_1 \rightarrow M_{e1}$ and $A_2{:}BR\text{-}T_2 \rightarrow M_{e2}$ according to Observation 1. If m_c is a power flow measurement $m\text{-}b_lb_k$, its A_1^{-1}-mapped branch and A_2^{-1}-mapped branch are the same branch b_lb_k; else if m_c is a power injection measurement $m\text{-}b_k$, its A_1^{-1}-mapped branch in $A_1{:}BR\text{-}T_1 \rightarrow M_{e1}$ and A_2^{-1}-mapped branch in $A_2{:}BR\text{-}T_2 \rightarrow M_{e2}$ may be two different branches incident to bus b_k. Accordingly, it is concluded that COG_1 and COG_2 may be different only when there exist critical injection measurements, which is summarized in the follow theorem:

Theorem 3: *COG* of a given (*G*, *M*) is always unique unless there are critical injection measurements.

Note that the observability property of a (G, M) is unique and determined solely by the configuration of G and M. The COG based topological approach discussed in the next section is to investigate the observability property of (G, M) and therefore different COGs should and will lead to the same result when they are applied to analyze the observability properties of a given (G, M).

2.4 Examine the network observability under a contingency

First, the COG based necessary and sufficient conditions for a power system network to maintain its observability against any single contingency are developed in subsection A. Then a conceptual example is discussed in subsection B to examine the network observability under a contingency accordingly.

A. A Theorem to examine network observability using COG

After constructing the associated *COG* for an $A{:}BR\text{-}T \rightarrow M_e$ of an observable (*G*, *M*), we can then apply the following theorem to examine the network observability under a contingency. The theorem is proven in the appendix B.4:

Theorem 4: (*G*, *M*) maintains its observability against any single contingency if and only if the associated *COG* satisfies both of the following conditions:

Condition 1: Radial branches in *COG* are radial in *G*;

Condition 2: Every measurement whose A^{-1}-mapped branch is radial in *COG* is redundant.

B. A conceptual example

For the (*G*, *M*) in Fig.2.1, it cannot maintain its observability against every single contingency because radial branch b8b9 in its COG in Fig.2.3 is non-radial in *G*, which violates the first condition in Theorem 4. Actually measurement m-b9 is labeled and identified as 'critical' during the construction of COG and therefore the (*G*, *M*) is not observable against a single contingency, a loss of *m-b9*.

If injection measurement m-b9 is replaced by two flow measurements *m-b9b8* and *m-b8b9*, then there will be no critical measurement any more. However, COG remains unchanged and therefore (*G*, *M*) still cannot maintain its observability under the contingency of a branch outage on *b8b9*.

If injection measurement m-b9 is retained while an extra flow measurement m-b8b9 is added, then there will be no critical measurement any more. Furthermore, branch b7b9

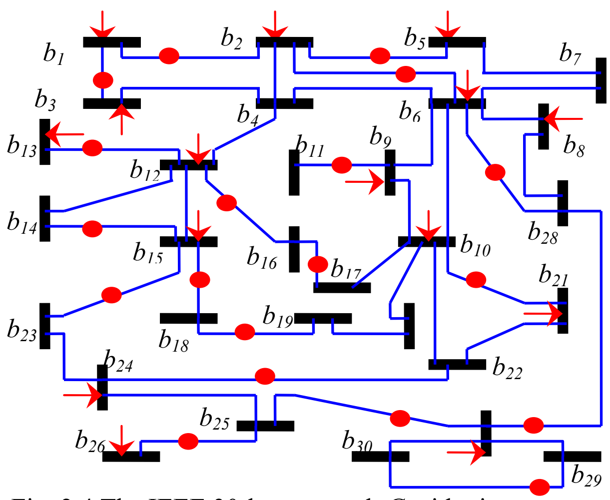


Fig. 2.4 The IEEE 30-bus network *G* with given measurement set *M*

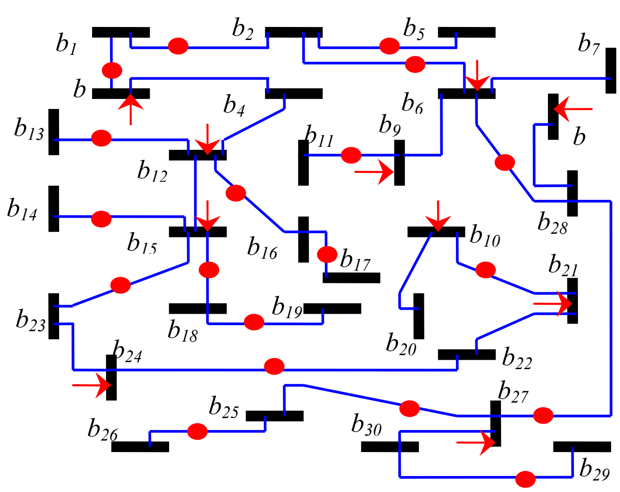


Fig. 2.5 Spanning tree and essential measurements in Fig.2.4 for A:BR-T→Me

will be inserted into the COG as the topologically related branch of redundant measurement m-b9 and therefore (G, M) can maintain its observability under any single contingency because both conditions in Theorem 4 are satisfied now.

2.5 Demonstrating examples

Three examples are given to demonstrate that the COG based observability examination methodology is valid in different situations where:

1. Some critical measurements exist and (G, M) can not maintain observability under a contingency.

2. No critical measurement exists but (G, M) still can not maintain observability under a contingency.

3. (G, M) can maintain observability under a contingency.

Note that the network G is the standard IEEE 30-bus sample system while the measurement configurations M are different in the three examples.

A. Example1: Critical measurement exist and (G, M) is not observable under a contingency

The IEEE 30-bus network *G* with a given measurement set *M* is shown in Fig.2.4.

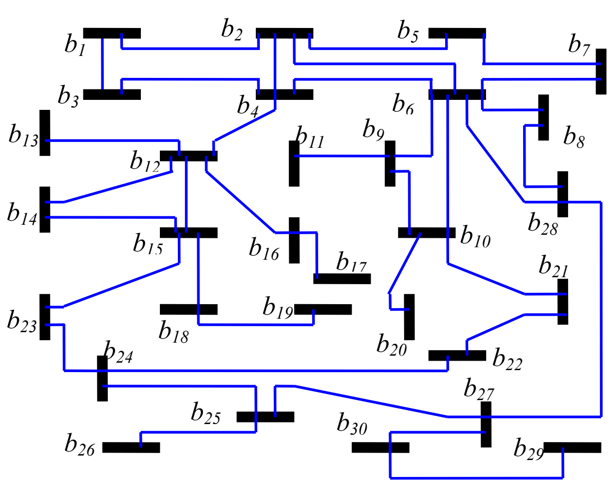


Fig. 2.6 Final *COG* for $A{:}BR\text{-}T \rightarrow M_e$ in Fig.2.5 of the given (*G*. *M*) in Fig.2.4

Step1: Construction of COG

An $A{:}BR\text{-}T \rightarrow M_e$ is found for the (G, M) via the classical topological observability analysis. The spanning tree T and the essential measurements $m_e \in M_e$ are shown in Fig.2.5. The set of nonessential measurements M_n={$m\text{-}b_1$, $m\text{-}b_2$, $m\text{-}b_5$, $m\text{-}b_{13}$, $m\text{-}b_{26}$}.

With the *COG* construction algorithm, branches b_2b_4 and b_5b_7 are added into *COG* in Stage3. Furthermore, after the iteration process of Stage 5 that is summarized in Table 2.2, more branches are added into the *COG* and the *COG* is finalized as shown in Fig.2.6. In addition, measurements in $M_{u,5}$ are labeled as '*critical*' while other measurements are labeled as '*redundant*', which indicates measurements in $M_{u,5}$ are critical while other measurements are redundant according to Theorem 2.

TABLE 2.2 THE ITERATION PROCESS OF STAGE 5 FOR FIG.2.4

i	$M_{u,i}$	branch added to *COG*
1	{measurements shown in Fig.2.5}	b_4b_6, b_6b_8, b_6b_{10}
2	{$m\text{-}b_8$, $m\text{-}b_6b_{28}$, $m\text{-}b_{10}b_{21}$, $m\text{-}b_{21}$, $m\text{-}b_{22}b_{24}$, $m\text{-}b_{24}$, $m\text{-}b_{15}b_{23}$, $m\text{-}b_{15}$, $m\text{-}b_{12}$, $m\text{-}b_9$, $m\text{-}b_{25}b_{27}$, $m\text{-}b_{27}b_{28}$, $m\text{-}b_{14}b_{15}$, $m\text{-}b_{15}b_{18}$, $m\text{-}b_{12}b_{16}$, $m\text{-}b_9b_{11}$, $m\text{-}b_{27}$, $m\text{-}b_{29}b_{30}$, $m\text{-}b_{10}$, $m\text{-}b_{16}b_{17}$, $m\text{-}b_{18}b_{19}$}	$b_{12}b_{14}$, $b_{24}b_{25}$, b_9b_{10}
3	{$m\text{-}b_{25}b_{27}$, $m\text{-}b_{27}b_{28}$, $m\text{-}b_{14}b_{15}$, $m\text{-}b_{15}b_{18}$, $m\text{-}b_{12}b_{16}$, $m\text{-}b_9b_{11}$ $m\text{-}b_{27}$, $m\text{-}b_{29}b_{30}$, $m\text{-}b_{10}$, $m\text{-}b_{16}b_{17}$, $m\text{-}b_{18}b_{19}$}	None
4	{$m\text{-}b_{27}$, $m\text{-}b_{29}b_{30}$, $m\text{-}b_{10}$, $m\text{-}b_{16}b_{17}$, $m\text{-}b_{18}b_{19}$}	None
5	Iteration ends since $M_{u,5} = M_{u,4}$	

Note that the number of iterations is 5 in this example, which is much less than 29, the upper bound of the number of iterations.

Step2: Examine the network observability in a contingency

The radial branches $b_{12}b_{16}$, $b_{16}b_{17}$, $b_{10}b_{20}$, $b_{18}b_{19}$, $b_{15}b_{18}$, $b_{27}b_{30}$ and $b_{29}b_{30}$ in *COG* are non-radial in G, which violates the necessary and sufficient conditions given in Theorem 4 to maintain the network observability against any single contingency. Therefore, it is concluded that the given (G, M) is unable to maintain its observability against every single contingency.

B. Example2: No critical measurement but (G, M) still is not observable under a contingency

The sample system in Example2 is as same as that in Example1 except that measurements m-b_{18}, m-b_{16}, m-$b_{27}b_{29}$ and m-$b_{10}b_{20}$ are added.

Step1: Construction of COG

With the same $A{:}BR\text{-}T \rightarrow M_e$ in Example 1, the corresponding *COG* is finalized as same as the *COG* in Example 1 except that additional branches $b_{10}b_{17}$, $b_{10}b_{22}$ and $b_{27}b_{29}$ are further included in *COG*. In addition, all measurements are labeled as '*redundant*' after the construction of the *COG* and therefore there is no critical measurement.

Step2: Examine the network observability in a contingency

The radial branches $b_{18}b_{19}$ and $b_{15}b_{18}$ in *COG* are non-radial in G, which violates the necessary and sufficient conditions given in Theorem 4 to maintain the network observability against any single contingency. Therefore, it is concluded that the given (G, M) is unable to maintain its observability against every single contingency even though there is no critical measurement.

C. Example3: (G, M) is observable under a contingency

The sample system in Example3 is as same as that in Example1 except that measurements m-b_{20} and m-$b_{27}b_{29}$ are added.

Step1: Construction of COG

With the same $A{:}BR\text{-}T \rightarrow M_e$ in Example 1, the corresponding *COG* is finalized as same as the *COG* in Example 1 except that additional branches $b_{10}b_{17}$, $b_{10}b_{22}$, $b_{19}b_{20}$ and $b_{27}b_{29}$ are further included in *COG*. In addition, all measurements are labeled as '*redundant*' after the construction of the *COG* and therefore are identified to be redundant.

Step2: Examine the network observability in a contingency

The current *COG* satisfies the necessary and sufficient conditions given in Theorem 4 to maintain the network observability against any single contingency. Therefore, we can conclude that the given (*G*, *M*) maintains its observability against any single contingency.

The novel *COG* based approach can also be applied to design a measurement system to maintain the network observability against any single contingency, which will be discussed in the coming chapter.

2.6 Conclusions

A novel topological approach is proposed to identify critical measurements and to examine network observability under a contingency, where the contingency can be any single branch outage or measurement loss. To advance the classical topological observability analysis, a new concept of contingency observability graph (*COG*) is introduced and it is proven that a power system network maintains its observability under a contingency if and only if its *COG* satisfies some conditions. With little extra computation load, the proposed *COG* based approach is a natural extension of the classical topological observability analysis for online applications. The IEEE 30-Bus system is used to demonstrate the validity and efficiency of our approach.

CHAPTER III

MEASUREMENTS AND RTUS PLACEMENT DESIGN AGAINST A CONTINGENCY

3.1 Introduction

As discussed in Topic 2 of Chapter I, it becomes more and more important to design a measurement system that can maintain its observability under a contingency.

In this chapter a two-stage topological measurement and RTU placement design is proposed as follows: We formulate the measurements and RTUs design problem in Section 2. In Section 3 the concepts of QCOG and preferred QCOG are discussed and an algorithm to determine a preferred QCOG is proposed as stage 1 of our two-stage approach. A heuristic measurement and RTU placement algorithm is developed in Section 4 as stage 2. The demonstrating example on the IEEE 30-bus system is given in Section 5. Then we conclude in the last section.

3.2 Problem statement

A. Constraints: Maintaining observability

The network is required to maintain its observability under a contingency which can be a measurement loss, a pre-selected branch outage or an RTU loss. A sample system is given in Fig.3.1.

For an N-Bus power system with (2N-1) state variables, the minimum number of measurements to make the system observable is (2N-1). Such (2N-1) measurements consist of (N-1) measurement pairs and a bus voltage magnitude measurement and they are all critical measurements. Note that the location of the (2N-1) measurements needs to be configured carefully so that there exists a valid measurement assignment as discussed in Theorem 1 of Section 2.2.A.

In order to keep system observable under any single measurement loss, every measurement needs to be redundant, which indicates that at least N power flow or injection measurement pairs plus a bus voltage magnitude measurement are necessary.

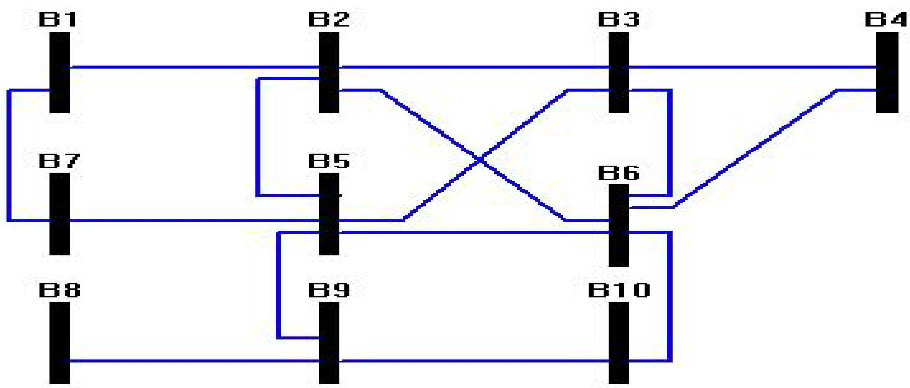


Fig.3.1 The sample system

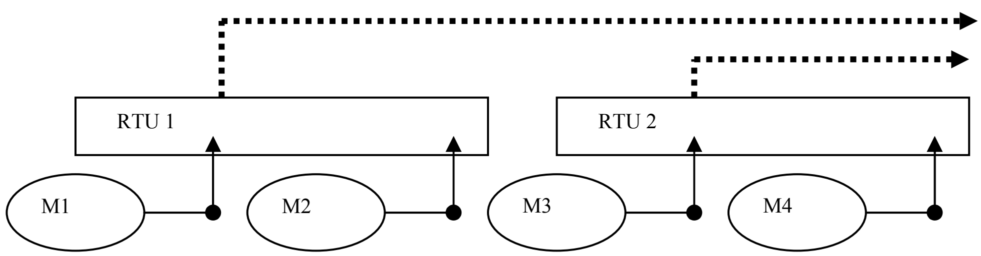


Fig.3.2 An RTU configuration scheme in a substation

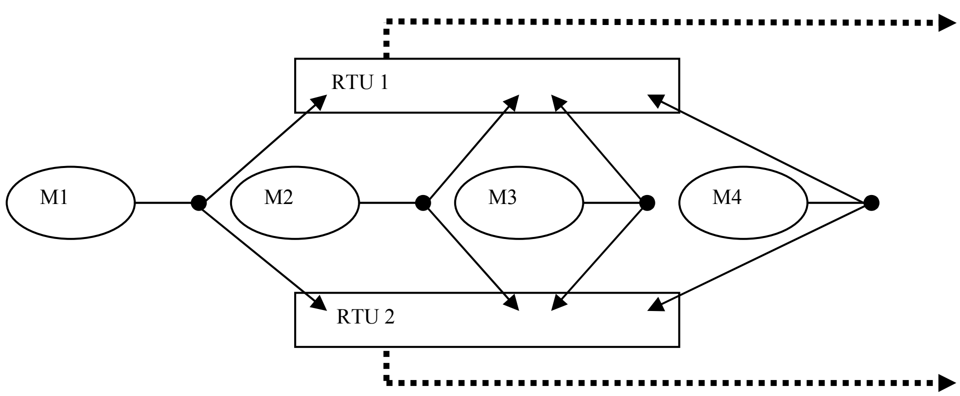


Fig.3.3 A new RTU configuration scheme in a substation

In other words, the minimum number of measurements is (2N+1) for observability maintenance under any single measurement pair loss.

Meters measure power injections, power flows and bus voltage magnitudes, while RTUs' communication channels transfer the measured raw data from a substation to the control center for further processing. Recent technical advance makes abundant channels in one RTU, and thus all the measurement raw data from a substation can share one RTU installed in the substation. However, in order to maintain the observability under loss of an RTU, two RTUs are installed in one substation to transfer measurement data as shown in Fig.3.2. Then when one RTU is lost, not all measurement data of the substation are lost. For example, if RTU1 in Fig.3.2 is lost, then measurement data 1 and 2 are lost while measurement data 3 and 4 are still available to the control center. Most traditional RTU design algorithms [29, 30] are based on the architecture as Fig.3.2.

On the other hand, a new RTU architecture shown in Fig.3.3 is proposed in this chapter with the following novelties:

1. As shown in Fig.3.3 all data is transferred into two RTUs simultaneously. Since all the devices are in the same substation, we can easily send the measurement data into more than one RTU at one time.

2. Modern analog-to-digital converter (A/D) in RTUs can deal with more than 100 analog measurement inputs per second [28, 30]. The technical trend is that numerous channels are available in one RTU; accordingly, there are always much more channels in one RTU than the needed channels for one P/Q measurement pair. Therefore, all measurement data from one substation can be transferred by a single RTU. To fully utilize the channels and reduce the total number of RTUs, it is better to merge as many measurements as possible into one substation. For the time being we assume every bus stands for a distinct substation, which is true when the buses are connected via transmission lines. The impact of transformers in the grid is discussed later in Section 3.4.

The advantages of the novel scheme in Fig.3.3 are as follows:

1. When one RTU is lost, all measurement data are still available to the control center. Therefore, the observability is maintained. By contrast, when one RTU is lost in Fig.3.2, some measurement data will be unavailable to the control center, which will jeopardize the system observability.

2. The measurement data are transferred to the control center by channels of two independent RTUs. Therefore, we can detect RTU transmission errors from data

inconsistency of these two RTUs. Schemes such as consistency polling can be devised to further identify the transmission errors.

Since the architectures in Fig.3.2 and Fig.3.3 both use two RTUs, the novel RTU architecture in Fig.3.3 is preferred and applied in this chapter and the corresponding results are compared in Section 5 with those based on the old architecture in Fig.3.2 [29, 30].

B. Objective: Minimizing the number of equipments

The optimization objective is to minimize the installation cost.

Since both RTUs and measurement devices are necessary for a measurement system, we refine the conventional approach and include the RTU costs as part of the installation cost besides the measurement installation cost, especially since the RTU cost is significantly large in practice.

In order to simplify our optimization problem, we assume that the measurement installation costs are the same and RTUs costs are the same, which is actually true in most cases. Consequently, our objective can be further simplified to minimize the number of measurement devices and RTUs.

C. Overall idea of the problem solution

We are searching for a measurement configuration *M* to make the system network *G* observable against any single contingency. The problem is addressed by the following two-stage topological heuristic approach based on the concept of *COG*:

Based on the given system network G, even though the measurement configuration *M* is still unknown, we can determine a desired *COG* of (*G*,*M*) so that (*G*, *M*) retains its observability under one contingency. Such a desired *COG* is termed as a qualified *COG* (*QCOG*). Among many QCOGs, we further construct a preferred QCOG based on which the measurement and RTU design can be further simplified.

Based on the preferred *QCOG* constructed from stage1 and the given *G*, determine what the measurement configuration *M* will be so that COG of (G, M) is the preferred QCOG. Furthermore, in order to reduce the number of RTUs, we can refine the measurement placements in Stage 2 while keeping the *COG* of (*G*, *M*) unchanged.

We begin with Stage 1 to study the preferred *QCOG*.

3.3 Stage 1: Determination of a preferred Qualified COG (QCOG)

This section is organized as follows: First the new concepts of QCOG and preferred QCOG are proposed in subsection A and B respectively. Then an algorithm to find preferred QCOG is developed in subsection C. Finally, an illustrative example is discussed in subsection D.

A. Definition of QCOG

A graph *Q* is a qualified COG (QCOG) if the following two conditions are satisfied:

Q is a connected sub-graph of *G* and contains all buses of *G*, i.e. Q contains at least a tree of *G*. Branches and buses in *Q* are termed as Q-radial if they are radial in *Q*, i.e. they are not in any branch loop of *Q*; otherwise they are termed as *Q*-in-loop.

Q-radial branches do not contain any pre-selected single branch outage unless these *Q*-radial branches are also radial in *G*.

Note: A *Q*-radial branch only implies this branch is radial in *Q* while it does not have to be radial in G.

The following theorem tells the motive of the above definition of QCOG.

Theorem 3.1: (*G*, *M*) maintains observability under one contingency if and only if all measurements are redundant and the *COG* of (*G*, *M*) is a *QCOG*.

The proof of the above theorem is straightforward according to the properties of *COG* in Chapter II and appendix B. Note that stage 2 will ensure all measurement are redundant while in stage 1 we focus on *QCOG*.

Even if all branches in *G* are in the pre-selected single branch outage, *QCOG* still exists because: for radial branch in *G*, it is fine to include them in *QCOG* as *Q*-radial branches; for non-radial branch in *G*, we need to either exclude them from *QCOG* or included them in *QCOG* as a Q-in-loop branch. As an extreme case, Q can be equal to *G*, i.e. G itself is a QCOG of G.

B. Definition of preferred QCOG

As discussed above, there are many possible QCOGs and each of them can be COG of an $A{:}BR\text{-}T \rightarrow M_e$. As an extreme case, G itself can be QCOG of G. Actually G is the COG of (G, M) when M includes power injection measurements on every bus of G.

In order to facilitate the measurement and RTU placement design, we are tying to search for some preferred QCOGs satisfying certain extra conditions based on the design principles as discussed later.

A QCOG is a preferred QCOG if the following two additional conditions are satisfied:

Q-radial branches that are non-radial in G must be incident to a *Q*-in-loop bus. Consequently, preferred *QCOG* has at most one *Q*-radial tier and contains no branch whose two end buses are both *Q*-radial when there is no radial branch in G.

The number of branches in preferred *QCOG* is minimal, i.e., preferred *OCOG* will violate C1, C2 or C3 after the removal of any branch.

The condition C3 and C4 is to reduce the number of RTUs and detailed explanation will be given in section 3.4.

The following terms are defined to further facilitate the determination of a preferred *QCOG*:

• The Kernel Graph (KG) of a *QCOG* consists of all the branches and buses that are Q-in-loop, i.e. Q-radial branches and buses are excluded from KG.

Note that the KG of QCOG may be disconnected or even empty. For example, if *G* consists of two branch loops and three branches in series connecting the two loops, then the KG of *QCOG* will be disconnected since the three branches in series are excluded from the KG. If G itself has no branch loop at all, then the only tree of *G* is *G* itself. Accordingly, *QCOG* will be equal to *G* and also have no branch loop, which implies that the KG of *QCOG* will be empty.

• The degree of a bus in a graph is the number of branches incident to the bus. Accordingly, there are two degrees for a bus in QCOG: namely, α-degree is the degree in QCOG itself, and β-degree is the degree in the corresponding KG of QCOG. The β-degrees of Q-radial buses are zero because they are excluded from the KG, while the β-degree of any Q-in-loop bus is at least two.

C. An algorithm to find preferred QCOG

The algorithm, described as follows, modifies the network graph by deleting branches until the above four conditions are met so that a preferred QCOG is determined.

Step1: Begins with a candidate graph Q that includes all branches and buses of the network G.

Step2: Find the degrees of buses in Q.

Step3: Find the buses with α-degree=1 (i.e., only one branch is incident to the bus), keep these buses and their incident branches untouched during the whole process since they will remain radial no matter how Q evolves when some other branches are deleted. Permanently mark these buses and their incident branches as 'Q-radial' to indicate their topological property in the final preferred QCOG.

Step4: Tentatively mark the rest of buses and branches as '*in-loop*' since these buses and branches are indeed topologically in some branch loop at this time but may become radial in the final preferred QCOG after some branches are deleted later in the process.

Step5: Start with a set of buses, say {Bus-Id}, with a rule that the smaller α-degree the higher priority; call function mark-as-radial (Bus-Id) to tentatively remove some branches so that some buses and branch become topologically radial in Q and therefore can be marked as Q-radial. Details of function mark-as-radial (Bus-Id) and the motive are given at the end of this sub-section.

Step6: Update the degree information and re-mark all the relevant buses temporarily. Check if the conditions (1-3) of a preferred QCOG are satisfied in the current candidate graph. If yes, permanently retain the updated degree information and the tentative operations, i.e. the branches tentatively removed in Step 5 are permanently removed now. Otherwise, null the tentative operations and the degree information.

Step7: Repeat Steps 5-6 until no more branches can be removed.

Step8: Check those branches, whose two end buses have β-degrees greater than 2, named branch-in-test; tentatively remove branch-in-test from *Q* and check if *Q* still satisfies conditions (1-3). If yes, make the removal permanent; otherwise null the tentative removal of branch-in-test.

Step9: Repeat Step8 until no more branches can be removed from *Q*. Then condition 4 is satisfied and a preferred *QCOG* is found accordingly.

Steps 1-4 simply initialize a preferred QCOG, while Steps 5-7 and Steps 8-9 are two procedures to remove the redundant branches at different situations. In Steps 5-7, we are trying to create as many radial branches and buses as possible by removing redundant branches. Therefore, the bus's topological properties are changed from the temporary *in-loop* to *radial* in the final preferred QCOG. On the other hand, Steps 8-9 are to remove the redundant branches in loop based on the given bus topological properties from Steps 5-7, and the bus topological properties will remain unchanged during Steps 8-9.

The function mark-as-radial (Bus-Id) is described in the following C-style pseudo codes:

Function mark-as-radial (Bus-Id)

{

Get all the buses linked to Bus-Id, say {Bus-1, …, Bus-m};

If all the branches linked to Bus-Id are labeled as pre-selected outage, then exit;

Else, if a branch linked to Bus-Id, say branch-r, is radial, i.e. if a radial bus is connected to Bus-Id via a radial branch branch-r, then exit;

Else, tentatively remove all branches linked with Bus-Id from the graph except Branch-1, where Branch-1 connects Bus-Id and Bus-1 satisfying the following two conditions:

1, Branch1 is not pre-selected outage;

2, Current β-degree of Bus-1 is greater than 2.

}

The motive of this function is to remove as many branches as possible from Q so that some buses and branch become topologically radial in Q and therefore can be marked as Q-radial while the condition 1, 2 and 3 of preferred QCOG definition are not violated. Note that fewer branches in *QCOG* leads to less RTUs during the measurement and RTU placement design as discussed later in Section 3.4.

D. Illustrative example

A 10-bus system, as shown in Fig.3.1 with pre-selected single branch outage from {B1B7, B2B3, B2B5, B3B6, B4B6, B5B9, B6B10, B9B10}, is used to illustrate the above algorithm to determine a preferred QCOG.

First, a candidate graph is initialized as the network G including all branches and buses. Obviously branch B8B9 and bus B8 are topologically radial because current α-degree of B8 is one. Then the other buses and branches are tentatively marked as *in-loop* since these buses and branches are indeed topologically in some branch loop at this time but may become radial in the final preferred QCOG.

In Step5, since the α-degree of B1 is as small as 2, we call function mark-as-radial to first process B1. B1 is connected to B2 and B7, however only branch B1B2 (Here Bus-Id is B1, Bus-1 is B2, Branch-1 is B1B2) satisfies the two conditions, i.e.

1, B1B2 is not pre-selected outage;

2, Current β-degree of B2 is greater than 2.

Therefore, we tentatively remove all branches linked with B1 except branch B1B2, i.e. we remove B1B7.

In Step6, we update the degree information temporarily and re-mark *Q-radial* on buses B1 and B7 and branches B1B2 and B7B5 because at this time the α-degrees of both B1 and B7 are one and they are indeed topologically radial after the tentative

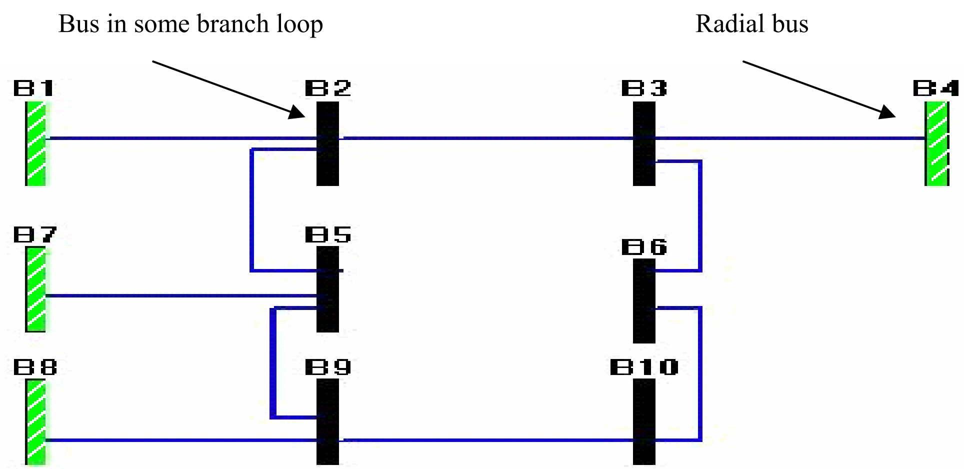


Fig.3.4 Preferred QCOG for the sample system

removal of branch B1B7. We find the candidate graph satisfies conditions (1-3) after the above tentative operations; therefore we permanently keep all the tentative modifications.

The same process is repeated on other buses until no more modifications can be made. Branch B4B6 is removed during the process, and bus B4 and branch B3B4 become *Q-radial* accordingly.

In Step8 and Step9, redundant branches such as B2B6, B5B3 and B5B6 are removed one by one. Note that β-degrees of the corresponding buses are all greater than two before the removal and all the involved buses B2, B3, B5 and B6 continue to be *Q-in-loop*.

Finally, a preferred QCOG is determined as shown in Fig.3.4, which consists of a branch loop (B2B5, B5B9, B9B10, B10B6, B6B3, B3B2) and four Q-radial branches {B2B1, B5B7, B9B8, B3B4}.

3.4 Stage 2: Measurement/RTU placement algorithm

Based on the preferred QCOG given in Stage 1, the measurement/RTU placement is determined in Stage 2 via the following sub-stages:

A. Stage 2.1 Place injection measurement pair on every bus of preferred QCOG.

Since QCOG covers all buses in G, it equals to place injection measurement on every buses of network G. It is straightforward to prove that the given (G, M) has the following three features:

Feature 1: All measurements are redundant.

Feature 2: The COG of such a (G, M) is a QCOG.

Feature 3: Based on Theorem 3.1, (G, M) is observable under any single measurement pair loss or a pre-selected single branch outage.

Note that a voltage magnitude measurement needs to be placed on a bus.

The total number of measurements after Stage 2.1 is (2N+1), the minimum number of measurements under the observability constraints, which indicate the design is already optimal if measurement placement is the only concern. On the other hand, if the RTU placement is considered, the problem become so complicated that only some heuristic approaches are applied in the existing publications where the optimal or even suboptimal solution under some conditions are not guaranteed. Similarly, we further refine the measurements placement in the following Stage 2.2 so that the number of RTUs can be reduced later in Stage 2.3.

B. Stage 2.2 Replace injection measurement with flow measurement based on QCOG.

Replace the injection measurement on the Q-radial buses in the preferred QCOG with flow measurement on the corresponding Q-radial branches in the preferred QCOG.

Based on the construction of COG over (G, M), it is clear that the three features in Stage 2.1 are retained after Stage 2.2.

The motive of Stage 2.2 and Condition 3 of preferred QCOG definition is discussed as follows:

To merge multiple measurements into one bus to fully utilize RTU communication channels, power flow measurement pairs are preferred because a flow measurement pair can install on two alternative buses with the same observability performance; but power injection measurement pairs have no such alternatives. Therefore, the flow measurements are more flexible than the injection measurement pairs for RTU channel allocations and hence are preferred. The voltage magnitude measurement is even more flexible than flow measurement pairs since its bus location has no impacts on the system observability. As a consequence, since we locate injection pairs on buses in branch loops of the preferred QCOG and place flow pairs on Q-radial branch of the preferred QCOG, we should increase the number of Q-radial branches of the QCOG as long as they do not contain the pre-selected branch outage.

Furthermore, the power flow measurements should be installed on the buses where injection measurements are already installed. For example, for a radial branch (Bi, Bj) whose one end bus Bi is radial and the other end Bj is in some branch loop. Then we have power flow measurement either M-BiBj or M-BjBi on this radial branch and a power injection measurement M-Bj on the bus Bj. We should choose M-BjBi instead of M-BiBj because M-BjBi and M-Bj are in the same bus (substation) and can share one

RTU.

Based on the above analysis, the motivation of condition 3 of preferred QCOG definition is to merge as many power flow and injection measurements as possible into one bus to reduce the number of RTUs in the next stage 2.3.

C. Stage 2.3 Place RTUs based on the measurement configuration.

Based on the novel architecture in Fig.3.3, one RTU is installed in substations with only one measurement pair and two RTUs are installed in substations with more than one measurement pair.

Accordingly, loss of any single RTU has no impact on the system observability because we place two RTUs when there is more than one measurement pair in the substation. If there is only one measurement pair and one RTU in the substation, losing one RTU leads to the loss of a single measurement pair, which will not jeopardize the system observability due to the fact that all measurements are redundant as the first feature in Stage 2.1.

D. Stage 2.4 Adjust RTUs placement with the considerations of transformers.

The assumption that every bus stands for an independent substation is not always true. In fact, multiple terminal buses of a transformer are located in the same substation. For example, if branch B2B5 stands for a 2-winding transformer connecting B2 and B5 in Fig.3.1, then buses B2 and B5 are in the same substation. Accordingly, this substation is labeled as substation (B2, B5). Note that the measurements located on the same substations can always share one RTU, even though the measurements are located on different buses of the same substation. Accordingly, we need to assign RTU based on the substations instead of buses. For example, if branch B2B5 is a transformer, then B2 and B5 are in the same substation (B2, B5) and the measurements located on B2 and B5 can share the same RTUs in substation (B2, B5) accordingly.

E. Stage 2.5 Final heuristic adjustments on measurements/RTUs placement.

Tentatively replace the injection measurement pair on those buses whose β-degree in QCOG is as small as 2 (i.e. only two branches are incident to the bus, thus these branches are in the same loop of QCOG) by flow measurement pair on one of those two branches and reapply Stage 2.3 on the updated (G, M). Permanently retain such a tentative modification if the following two conditions are double checked to be satisfied:

C1. The corresponding (G, M) still satisfies the three features in Stage 2.1, thus such a change has no impact on the observability performance.

C2. The number of necessary RTUs decreases after such an adjustment, which is very possible because flow measurement pair is more flexible than injection measurement pair in terms of RTU allocations.

Condition 4 of a preferred QCOG definition is to simplify the configuration of QCOG and therefore helps to apply this heuristic principle to reduce the number of RTUs and utilize the RTUs more efficiently.

F. An illustrative example

For the sample system in Fig.3.1 and its corresponding preferred QCOG in Fig.3.4, the injection measurements are installed on every bus after Stage 2.1.

In Stage 2.2, M-B1 can be replaced by either M-B1B2 or M-B2B1. Since there is already an injection measurement M-B2 in bus B2, M-B2B1 should be chosen to replace M-B1 so that both M-B2B1 and M-B2 are installed in the same bus. Similarly M-B4, M-B7 and M-B8 are replaced by M-B3B4, M-B5B7, M-B9B8 respectively.

In Stage 2.3, we need to install two RTUs in B2, B5, B9 and B3 and one RTU in B6 and B10 because there are only one measurement in B6 and B10 while more than one measurement in B2, B3, B5, and B9.

In Stage 2.4, suppose branch BR-B2B5 is actually a transformer and therefore B2 and B5 are in the same substation (B2, B5), then we only need to install two RTUs in the substation (B2, B5).

In Stage 2.5, we tentatively replace M-B10 with M-B9B10 and confirm:

1. The three features in Stage 2.1 are not violated;

2. RTU in bus B10 can be removed now.

Therefore such a modification is permanently retained.

If we continue to tentatively replace M-B6 with M-B3B6, then we find the feature 2 is violated as follows:

Branch BR-B9B10 is one of the pre-selected branch outage and is not radial in G at the same time; however, the corresponding COG does not include branch BR-B6B10 any more and thus branch BR-B9B10 is radial in COG, which violates the second condition of QCOG.

3.5 Demonstrating tests

The problem is to minimize the number of measurements and RTUs for the IEEE 30-Bus standard system shown in Fig.3.5, so that system remains observable under any single measurement loss, any single RTU loss or any pre-selected single branch outage from {B15B23, B2B5, B22B24, B10B20, B27B28, B16B17, B12B14, B12B15}. The pre-selected branches are labeled in Fig.3.5.

Here every substation consists of one bus except substations (B12, B4), (B6, B9, B10) and (B27, B28). Note: B6, B9 and B10 are the terminal buses of a three-winding transformer.

Stage 1: Based on the algorithm in Section3, a preferred QCOG in Table 3.1 is found.

Stage 2.1: Injection measurements are installed in every bus, and a voltage magnitude measurement is also installed, where the number of measurements is already minimized under the observability requirements.

Stage 2.2: The measurements placement is given in Fig.3.5, where many injection measurements in Stage 2.1 are replaced with power flow measurements.

Stage 2.3: Place two RTUs on buses with more than 1 measurement pairs and one RTU on bus with one measurement pair, as given in Table 3.2.

Stage 2.4: We only need two RTUs on substation (B12, B4), (B6, B9, B10) and (B27, B28) respectively.

Stage 2.5: The injection measurements M-B22, M-B5 and M-B18 are replaced by flow measurements M-B24B22, M-B2B5 and M-B15B18 respectively. Accordingly, the RTUs located on buses {B22, B5, B18} are removed and the existing RTUs located on buses {B24, B2, B15} are more efficiently utilized.

In addition, the voltage magnitude measurement on B7 in Fig.3.5 is relocated to another substation with more than one measurement pair, say B24. Then only one RTU

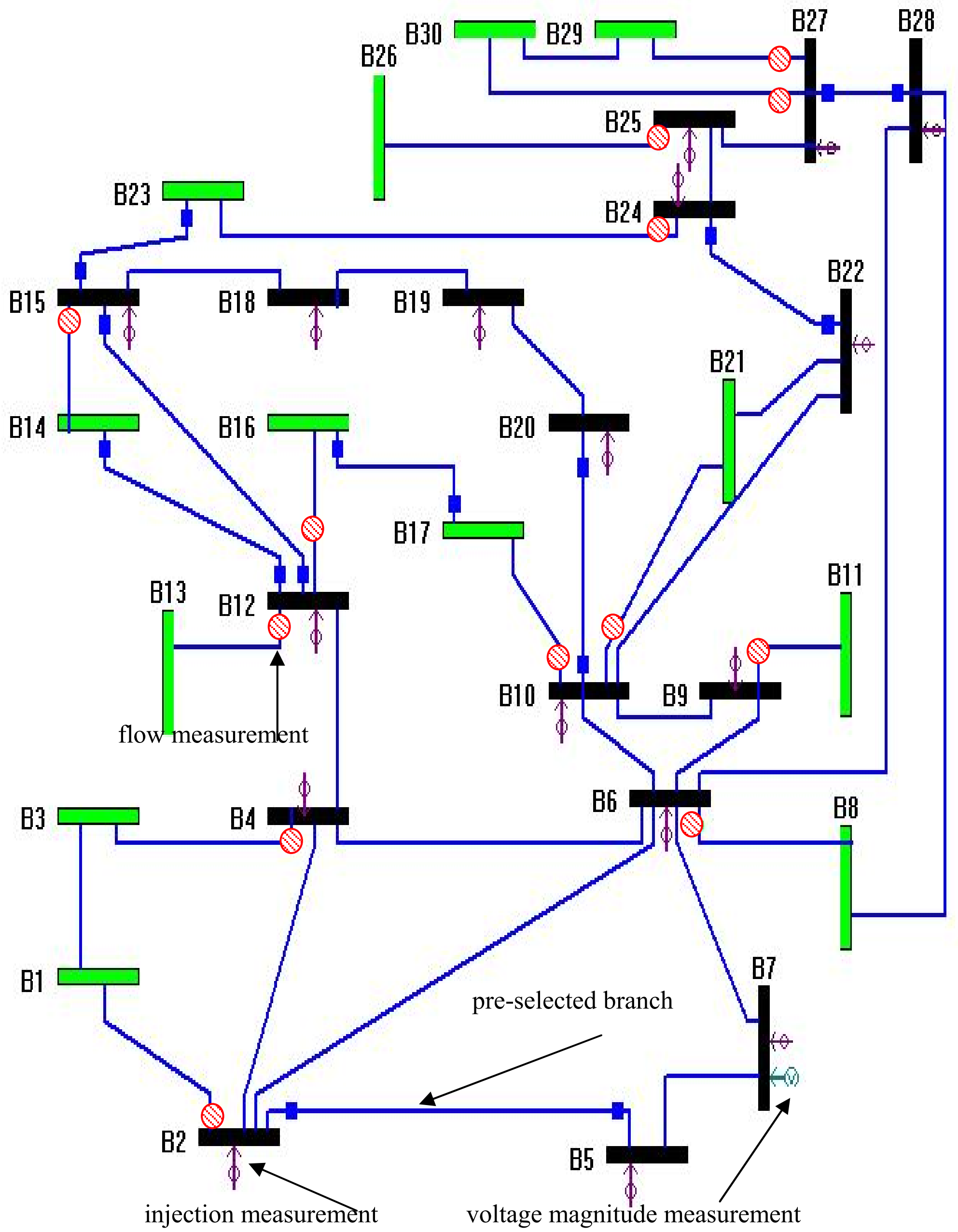


Fig.3.5 Measurement configuration after Stage 2.2

is needed in B7 because there is only one measurement pair on B7 now. At the same time, the number of RTU on B24 is still two as before.

Finally the number of RTUs is reduced to 17 as shown in Table 3.3. Note that the measurement number is unchanged and the observability requirements are still satisfied.

Compared with the result in [29, 30], the number of RTUs is 17, same as 17 in [29] and smaller than 18 in [30] while all the constraints considered in [29, 30] are satisfied. In addition, the number of measurements is minimized in this chapter, while in [29, 30] the number of measurements is much higher.

Furthermore, the topological algorithm avoids the complicated numerical iteration process in [29, 30].

TABLE 3.1 CONFIGURATION OF THE PREFERRED QCOG

Branch loops	Topologically Radial branches
B2B4, B4B6, B6B7, B7B5, B5B2	B2B1, B4B3, B6B8
B6B4, B4B12, B12B15, B15B18, B18B19,B19B20,B20B10,B10B6	B12B13, B15B14, B12B16, B10B17
B6B10, B10B9, B9B6	B9B11
B6B10, B10B22, B22B24, B24B25,B25B27,B27B28,B28B6	B10B21, B24B23, B25B26, B27B30,B27B29

TABLE 3.2 CONFIGURATION OF THE RTUS AFTER STAGE 2.3

Substations w/ one RTU	Substations with two RTUs
B19, B20, B22, B5, B18	B2, B4, B12, B6, B9, B10, B7, B15, B24, B25, B27, B28

TABLE 3.3 CONFIGURATION OF THE RTUS AFTER STAGE 2.5

Substations with one RTU	Substations with two RTUs
B7, B19, B20	B2, (B4, B12), (B6, B9, B10), B15, B24, B25, (B27, B28)

3.6 Conclusions

A two-stage topological approach is proposed to minimize the number of measurements and RTUs under the constraint that the system remains observable under any single contingency. The contingency can be a single branch outage from a pre-selected branch set or a single measurement loss. In addition, a single RTU loss is taken into account as a type of contingency. The approach is based on the concepts of qualified COG (QCOG) and preferred QCOG, and an algorithm to search for a preferred QCOG is developed and a heuristic measurement and RTU placement methodology is proposed based on the preferred QCOG. The IEEE 30-Bus system is used to demonstrate the validity and efficiency of our approach.

CHAPTER IV

A TEXTURED DISTRIBUTED STATE ESTIMATION ALGORITHM BASED ON DATA EXCHANGE

4.1 Introduction

As discussed in Topic 3 of Chapter I, a concurrent textured algorithm in state estimation problem is proposed in this chapter as a distributed state estimation algorithm and will overcome the disadvantages of OSE. Furthermore, compared with existing DSE algorithm, the performance of texture DSE improves greatly in respect of bad data detection and identification ability and decreasing the discrepancy around boundary buses.

This chapter is organized as the following: the overall idea of the textured DSE algorithm which consists of the off-line textured architecture design and the on-line textured computation is discussed in Section 2. The basic idea of off-line textured architecture deisgn is introduced in Seciton 3. As a preliminary study for off-line architecture design, the concept of Bus Credibility Index (BCI) is described in Section 4. Furthermore, a heuristic approach is discussed in Section 5 to complete the off-line textured architecture design. On-line textured computation and the advantages of the textured DSE algorithm are discussed in Section 6 and 7 respectively. Numerical tests are studied in Section 8. A conclusion is drawn in the last section.

4.2 Overall idea of the concurrent textured DSE Algorithm

The textured DSE consists of the off-line textured architecture design and the on-line textured computation. The off-line design is to determine the architecture of texture, i.e. to design the raw data (real time raw measurement information before execution of local estimator) and estimated values (estimation results after execution of local estimator) exchanges among neighboring state estimators; the on-line computation is to fully take advantage of the existing local state estimators and complete the state estimation over the whole system.

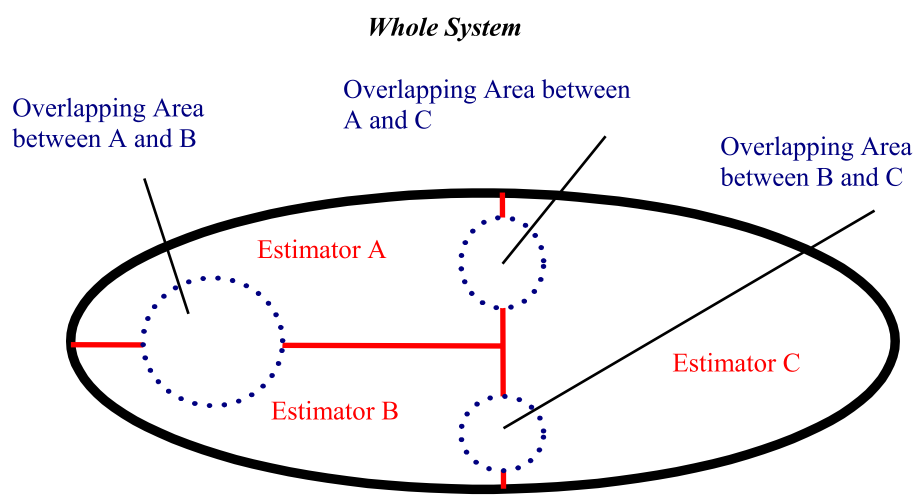


Fig.4.1 Overlapping areas come into being after data exchange

Note that the on-line textured computation relies on the raw data or estimated value exchanges among neighboring estimators through the communication in the textured architecture. And the texture which is critical to the online computation is determined in the off-line textured architecture design.

When off-line design is completed, there are some overlapping areas in the neighboring estimators as shown in Fig.4.1, where some information is exchanged and shared. The key here is to determine the scope of the shared information: extended from the existing DSE algorithm where only the boundary buses are shared in the estimation sub-problems, the additional raw data and estimated values near the boundary buses are also exchanged among neighboring entities. These data exchanges are introduced simultaneously between multiple entities, such as Company A and ISO B, ISO B and RTO C, etc. Accordingly, textured network is formed in Fig.4.1.

4.3 Basic idea of off-line textured architecture design

As the first stage of the textured DSE, off-line textured architecture design is to determine how to exchange raw data or estimated values within neighboring state estimators, i.e. to select a set of real time raw data and a set of estimated values to be exchanged between neighboring estimators. The quality of individual estimators is improved via the carefully selected data exchanges in terms of both estimation reliability and estimation accuracy, where estimation reliability refers to probability to maintain the observability under measurement loss and bad data detection and identification capability.

It is not true that 'bigger area always leads to a better estimation result'. For example, there exist some 'harmful' data exchanges as demonstrated in Example 1 in Section 4.8, where area is expanded and gets bigger while the estimation result gets worse. Even for beneficial data exchange, a trade off in the selection of data exchange is necessary as discussed in the following.

As an extreme case, if all the raw data is exchanged and shared, the estimator becomes an OSE over the whole system instead of a DSE. On the other hand, if no measurement is exchanged, it becomes the existing DSE algorithm, which has the drawbacks described before in Section 1.4.B. Therefore, a trade off in the selection of data exchange is necessary to make the overlapping areas moderate, neither too large nor too small. In other words, we need to carefully determine an appropriate texture which is decisive for the online computation.

In addition, in the original textured decomposition method [45-47], the decomposition is based on the algorithm. However, in the textured DSE problem discussed here, the range of individual local estimators has been determined in advance from the actual industry ownership. And the hardware and software cost on data exchange implementation should be minimized, which implies smaller overlapping areas are preferred as long as the estimation performance is satisfied.

Experience alone cannot resolve the texture design issues because our studies indicate some data exchange actually can be harmful for estimators as discussed in numerical Example 1 given in Section 4.8 and few people if any have enough experience on data exchanges between estimators yet; particularly there is no experience for data exchange among big mega-RTOs.

Therefore, it is critical to develop a systematic approach to efficiently search for appropriate data exchanges. Since the computation complexity increases dramatically for large grids, data exchange design problem become very challenging. A heuristic approach with the corresponding principles to complete the off-line textured architecture design is developed in Section 5, which is based on some preliminary study such as the concept of Bus Credibility Index (BCI) as described in Section 4.

4.4 Preliminary concepts for off-line textured architecture design

First, the concept of critical p-tuples in [23, 63] are reviewed. Then we introduce the new concepts of Weak Bus Set, Bus Redundancy Descriptor (BRD) and Bus Credibility Index (BCI) to facilitate the off-line textured architecture design.

A. Critical p-tuples

As a well developed concept, critical p-tuples is defined as a set of *p* measurements with respect to a system (G, M), where the originally observable system (G, M) becomes an unobservable system (G, M') after the removals of all the *p* measurements in the set M, i.e. M=M'$\cup$ {*p* measurements}. In addition, removals of any (*p-1*) measurements in the set {*p* measurements} will still keep the system observable.

The size of the critical *p*-tuples is defined as *p*. Critical p-tuples can be determined based on the analysis of symbolic Jacobian matrix *H* [64, 65]. The methodology in [18] is adopted by us to determine the critical tuples. A critical measurement alone is a critical measurement set with size equal to 1.

For the sample system *S*=(*G*,*M*) as shown in Fig.4.2, measurement M-B*9B10*, M-B*10*, and M-B*12B13* are a critical 3-tuples, which is denoted as (M-B*9B10, M-B10, M-B12B13*|*S*).

Remark 1: It is well known that a bad data in a critical p-tuples where p=1 can not be detected and therefore the estimation reliability is dramatically low; if p=2, the bad data can be detected but cannot be identified, which will still make the estimate result not reliable enough; if p=3, a single bad data can be not only detected but also identified, but two conforming bad data cannot be detected. Even if p=4, two conforming bad data in the set are still hard to be identified. Here the data refers to the raw measurement value from instrumentations. Therefore, the size of the critical p-tuples has a decisive impact on the estimation reliability when there are potential bad data in the p-tuples.

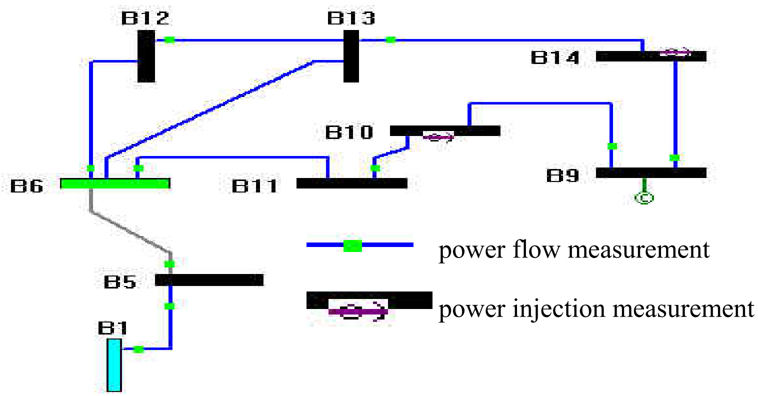


Fig.4.2 A sample system *S*=(*G, M*)

Remark 2: For critical p-tuples where p is greater than 4, even two conforming bad data can be identified and therefore will not have any major impact on the estimator reliability. Therefore, we will only consider critical measurement sets with size p=1,2,3 or 4 since now on in order to focus on the potential weak parts of the system.

B. Weak buses sets of a critical p-tuples

The concept of critical p-tuples is purely measurement oriented and cannot tell the estimation reliability of individual buses. Therefore, we need to further define the new concept of Weak Buses Set to evaluate the bus estimation reliability.

After the removals of a critical p-tuples, S=(G, M) becomes S_1=(G, M_1). Since there is no branch outage, G is still a connected graph; however, S_1 is not observable as a whole any more. Instead, there are two isolated observable islands (G_A, M_A) and (G_B, M_B) satisfying

$G_A \subset G, G_B \subset G, G_A \cap G_B = \Phi$;

$M_A \subset M_1, M_B \subset M_1, M_A \cap M_B = \Phi$;

(G_A, M_A) and (G_B, M_B) are observable respectively.

Since G is a connected graph, there exist some branches, say branch b_ib_j connecting GA and GB. If a power flow measurement m-b_ib_j (or m-b_jb_i) on branch b_ib_j is added into S1, then $S_2 = (G, M \cup \{m - b_ib_j\})$ obviously becomes observable as a whole again. The end buses of branch b_ib_j, i.e. buses b_i and b_j are defined as the *weak bus* of the critical p-tuples. Note that the branch b_ib_j is in neither observable island (GA, MA) nor (GB, MB) after the removals of the critical p-tuples, i.e. the power flow along b_ib_j remains unknown during the state estimation process and consequently the end buses b_i and b_j are rather weak in terms of estimation reliability, which explains the origin of the term '*weak bus*'. Accordingly, *the weak buses set* for this critical p-tuples is defined as the end buses of all those branches like b_ib_j connecting G_A and G_B. Obviously these weak buses are always located on the boundary of G_A and G_B because they are the end buses of branches connecting G_A and G_B.

The weak bus set of a critical p-tuples can be determined from its definition directly. For example, after the removal of measurements (M-B9B10, M-B10, M-B12B13|*S*) in the sample system in Fig.4.2, *S* becomes S1 while S1 is unobservable as a whole. Instead, there are two isolated observable islands as the followings

G_A={branches B1B5, B5B6, B6B12, B6B11, B11B10 and the corresponding buses B1, B5, B6, B12, B11, B10};

G_B={branches B9B14, B14B13 and the corresponding buses B9, B13, B14 },

M_A={M-B1B5, M-B5B1, M-B5B6, M-B6B12, M-B6B11, M-B11B10},

M_B={M-B9B14, M-B14, M-B13B14},

(G_A, M_A) and (G_B, M_B) are observable respectively.

If a pair of active and reactive flow measurements M-B6B13 (or M-B13B6) is added on line B6B13, the system S1=(G, M1) becomes S2=(G, M2) where M2=M1$\cup${M-B6B13} and S2 is observable as a whole again. Therefore, buses B6 and B13 belong to the Weak Bus Set of critical 3-tuples (M-B*9B10, M-B10, M-B12B13*|*S*). Repeat the trials on all branches without existing power flow measurement, the Weak Bus Set of (M-B*9B10, M-B10, M-B12B13*|*S*) is determined to be {bus B6, B9, B10, B12, B13}, which is denoted as {B6, B9, B10, B12, B13|(M-B*9B10, M-B10, M-B12B13*)|S}.

Similarly, the corresponding weak bus sets of all critical p-tuples can be determined. The result for the sample system in Fig.4.2 is given as Table 4.1:

TABLE 4.1. CRITICAL SET AND WEAK BUS SET OF THE SAMPLE SYSTEM IN FIG.4.2

p	Constitution of Critical p-set and the weak bus set
1	{B5, B6|(M-B5B6)|S}
2	{B6, B11, B12, B13|(M-B6B11,M-B6B12) |S }; {B6, B11, B12, B13|(M-B6B11, M-B12B13) |S }; {B6, B12, B13|(M-B6B12, M-B12B13) |S }; {B1, B5|(M-B1B5, M-B5B1) |S }
3	{B6, B9, B10, B12, B13|(M-B9B10, M-B10, M-B12B13) |S }; … …
4	… …

C. Bus Redundancy Descriptor (BRD)

Bus Redundancy Descriptors (BRD) of bus *b* is defined as the set of critical measurement *p*-tuples whose weak bus set contains bus *b*. The motive of *BRD* is to describe the estimation reliability of a particular bus *b* in terms of given (G, M). BRD is a natural extension of Weak Bus Set in the sense that Weak Bus Set is still defined on critical measurement set p-tuples while BRD is defined on individual network bus.

According to the above definition, bus *b* in G may have different BRD between (G, M) and (G, M') where M and M' are different measurement configurations over the same grid G.

Once the weak bus set of critical p-tuples are determined, the BRD can be obtained straightforwardly according to their definitions.

For example, as discussed before, the Weak Bus Set of (M-B9B10, M-B10, M-B12B13|S) is buses B6, B9, B10, B12 and B13; consequently, BRD of Bus6 consists of at least (M-B9B10, M-B10, M-B12B13|S) and some other possible critical sets. According to the results in Table 4.1, BRD of the buses in the sample system in Fig.4.2 is:

BRD(B1,*S*)={ (M-B1B5, M-B5B1), …};

BRD(B5,*S*)={(M-B5B6), (M-B1B5, M-B5B1), …};

BRD(B6,*S*)={(M-B5B6), (M-B6B11, M-B6B12), (M-B6B11, M-B12B13), (M-

B6B12, M-B12B13), (M-B9B10, M-B10, M-B12B13), …};

BRD(B9,*S*)={(M-B9B10, M-B10, M-B12B13), …};

BRD(B10,*S*)={(M-B9B10, M-B10, M-B12B13), …};

BRD(B11,*S*)={(M-B6B11, M-B6B12), (M-B6B11, M-B12B13), …};

BRD(B12,*S*)={(M-B6B11, M-B6B12) (M-B6B11, M-B12B13) (M-B6B12, M-B12B13), (M-B9B10, M-B10, M-B12B13), …}.

BRD(B13,*S*)={(M-B6B11, M-B6B12) (M-B6B11, M-B12B13) (M-B6B12, M-B12B13), (M-B9B10, M-B10, M-B12B13), …}.

Remark 1: Note that the measurements in BRD(*b*,*S*) either connects directly with *b* or locates on a loop that includes *b*. For example, BRD(B5,*S*) consists of (M-B5B6) and (M-B1B5, M-B5B1) which connect directly with B5, and BRD(B13,*S*) consists of critical tuples such as (M-B6B11, M-B6B12), (M-B6B11, M-B12B13), (M-B6B12, M-B12B13) and (M-B9B10, M-B10, M-B12B13), which are all located in the loop B6→B12→B13→B14→B9→B10→B11→B6.

Remark 2: BRD reveals the detailed relationship among measurements. For example, based on BRD(11,*S*)={(6-11,6-12), (6-11,12-13), …}, the result on Bus11 will be unreliable from conforming bad data on 6-11 and 6-12 because they are in the same critical p-tuples where p=2. On the other hand, bad data on 1-5 has no major impact on Bus11.

D. Bus Credibility Index (BCI)

The motive of Bus Credibility Index of Bus *b* is to quantify the estimation reliability on bus b according to bus b's corresponding detailed BRD information. BCI of bus b with respect to a specified system S=(G, M) is determined as:

$$\begin{aligned} BCI(b,S) &= 1 - P(C_1 \cup C_2 \cup \cdots \cup C_k) \\ &= 1 - \sum_{i=1}^{k}\left((-1)^{i-1} \sum_{1\le t1<t2<\cdots<ti\le k} P(C_{t1} \cap C_{t2} \cap \cdots \cap C_{ti}) \right) \\ &= 1 - \Big(\sum_{1\le t1\le k} P(C_{t1}) - \sum_{1\le t1<t2\le k} P(C_{t1} \cap C_{t2}) \\ &+ \sum_{1\le t1<t2<t3\le k} P(C_{t1} \cap C_{t2} \cap C_{t3}) - \cdots + (-1)^{k-1} P(C_1 \cap C_2 \cap \cdots \cap C_k) \Big) \end{aligned} \quad (1)$$

where

BCI*(b,S)* is the BCI of Bus *b* with respect to system S;

BRD*(b,S)* consists of *k* critical *p*-tuples C_i, $p=1,2,3,\ldots$;

$P(C_i \bigcap C_j)$ stands for the failure probability when all measurements in C_i and C_j fail.

If the failure probabilities of measurements are independent from each other, then $P(C_i \bigcap C_j)$ can be determined by:

$$P(C_i \bigcap C_j) = P(\{M_1, M_2, \cdots, M_l\}) = P(M_1) \cdot P(M_2) \cdots P(M_l) \qquad (2)$$

where

$\{M_1, M_2, \ldots, M_l\}$ are the measurement set which makes up $(C_i \cap C_j)$,

$P(M_l)$ stands for the failure probability of M_l.

Given the failure probability of every measurement, BCI*(b,S)* can be determined according to (1) and (2).

For example, suppose the failure probability of all measurements is a constant 0.01, then BCI*(b,S)* is determined as Table 4.2:

TABLE 4.2. BCI OF BUSES WITH RESPECT TO SAMPLE SYSTEM IN FIG.4.2

BCI(B5,S)	BCI(B11,S)	BCI(B13,S)
0.9900	0.9998	0.9997

Remark 1: BCI is a probability measure that quantifies the estimation reliability on bus b with respect to a specific system S. If the failure probability of measurements stands for the probability of measurement availability, then BCI(b,S) stands for the probability to maintain observability on bus b with respect to system *S* since the removal of all measurements of a critical *k*-tuples will make *S* unobservable. If the failure probability of a measurement stands for the probability of bad data in this measurement, then BCI(b,S) reflects the probability to successfully identify bad data since bad data cannot be identified if all the measurements of a critical k-tuples are bad data.

Remark 2: If BCI(*b1*,*S1*)>BCI(*b2*,*S2*), then Bus *b1* with respect to system *S1* is said to be stronger than Bus *b2* with respect to system *S2*. Note that data exchanges modify the original system *S* to *S'*, and the incremental difference of BCI from (*b*,*S*) to (*b*,*S'*) stands for the benefit of such a data exchange on bus *b* in terms of estimation reliability.

Remark 3: Given the condition that the failure probability of every measurement is very small, say less than 0.01, then the failure probability that all measurements in the critical 4-tuples will be less than 1e-8. and small enough to be ignored in the computation of BCI. Therefore, critical p-tuples where p is greater than 4 are discarded in the BRD/BCI, which is consistent with the Remark 2 in subsection 4.4.A.

Remark 4: Generally speaking, the failure probability of measurements is very small. Therefore, BCI is always a value less than but near to 1.0. Even a minor difference of BCI stands for a major difference in terms of estimation reliability.

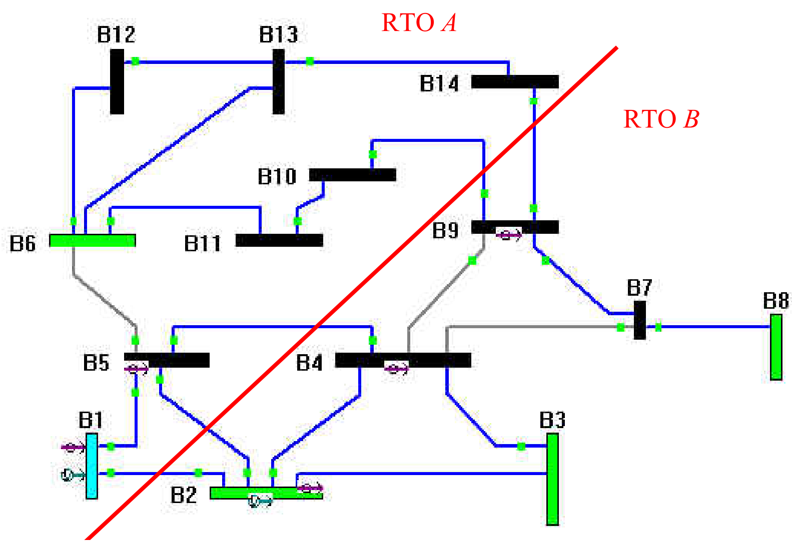


Fig.4.3 Two RTOs merge into one Mega-RTO

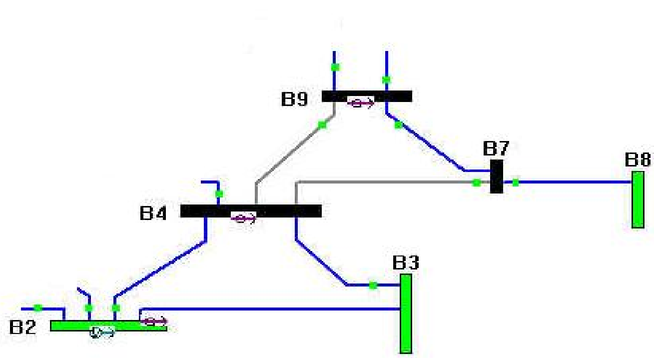


Fig.4.4 Original system of B before data exchange

4.5 A heuristic approach for off-line textured architecture design

In this section a heuristic approach is described for the off-line textured architecture design consisting of step A, B, C and D as follows. The objective is to find out the most beneficial data exchanges in terms of BCI increment where BCI quantifies the estimation reliability.

Note that the algorithm and principles are applicable to the data exchanges among any two connected neighboring estimators simultaneously, say *A* and *B*, *B* and *C*, and *A* and *C*, as shown in Fig.4.3 and 4.4. To simply the algorithm demonstration, we suppose (*G*, *M*) stands for a system consisting of only two local neighboring estimators *A* and *B*, i.e. $G=GA \cup GB$ and $M=MA \cup MB$.

A. Prepare input data

The necessary input data from the user include the followings:

1) The configuration, parameters and ownership of current power system network and measurement system;
2) COG of the system (G, M). Note that the COG instead of G is applied in the heuristic design approach, though COG may be equal or almost equal to the G for many realistic power systems due to many redundant injection measurements in the system.
3) The failure probability of measurements;
4) BCI(*b*, *S*), which reflects the estimation reliability on a bus *b* with respect to a specific system *S*, where *S* refers to A, B and A⋃B. A lot of computation will be involved later to figure out BCI(b, S) for different buses with respect to different systems. When *A* and *B* are merged to be part of a bigger Mega-RTO, many advanced functionalities such as congestion management need the network model over the whole Mega-RTO grid. Accordingly, we assume the network model including the measurement configuration over the whole grid is available for Mega-RTO and therefore the BCI information for every bus with respect to $A \cup B$ is available via off-line computations.

B. Search for beneficial raw data to be exchanged from B to A

We focus on *A* and extend *A* to include some buses and the corresponding measurements from *B*. We exhaust all boundary buses $\{b_{i,A}\}$ which belonging to *A* and adjacent to *B* with the following process:

StepB.1 Apply the following principles to search for possible beneficial raw data exchange around bus $b_{i,A}$.

Principles1: Estimator *A* should be extended from boundary bus $b_{i,A}$ to include adjacent buses in *B* given the condition $BCI(b_{i,A}, A \cup B) - BCI(b_{i,A}, A) > \varepsilon$.

Where

A and *B* are two connected neighboring estimators;

$b_{i,A}$ is a boundary bus of *A* and connected to *B*;

ε is a small positive number and serves as a threshold.

$BCI(b_{i,A}, A)$ stands for the estimation reliability in an extreme case when no raw data is exchanged while $BCI(b_{i,A}, A \cup B)$ stands for the estimation reliability in the other extreme case when all possible raw data are exchanged from B to A. With a moderate raw data exchange between the above two extremities, the estimation reliability will be between $BCI(b_{i,A}, A)$ and $BCI(b_{i,A}, A \cup B)$. Therefore, $BCI(b_{i,A}, A \cup B)$ - $BCI(b_{i,A}, A)$ is a reasonable estimation for the maximum possible BCI increase on $b_{i,A}$ after some raw data exchange. However, this is just a heuristic approach and Step B.2 will be applied to verify the overall benefit towards estimator *A*.

Principles2: The final configuration after data exchange should avoid forming a radial structure; instead, a loop is preferred. Note that COG of (G, M) instead of G itself is used here to check if there is a loop after the raw data exchange.

Principles3: There may be numerous candidate loops and the loop is preferred if the number of buses needed to be added into *A* to form such a loop is small and the BCI of these buses are high with respect to *A* after they are added into *A*.

StepB.2 Verify the overall benefit of the raw data exchanges towards *A*

Suppose some raw data is found to be exchanged from estimator *B* to estimator *A* in StepB.1, then estimator *A* is extended accordingly. Update the BCI information of estimator *A* to check if the data exchange is increasing the average BCI of all buses of Estimator *A*. If yes, retain the data exchange and continue; otherwise, abandon the found raw data exchange just found in StepB.1.

The following index may be used to check the overall impact of the data exchange on estimator A:

$$\sum_{b_{i,A} \in A} W_{b_A} \left(BCI(b_{i,A}, A') - BCI(b_{i,A}, A) \right)$$

Where $W_{bi,A}$ stands for the importance of bus $b_{i,A}$; for example, a bus in a higher voltage level or with a major load or generator may be assigned with a higher weight;

A' is the estimator *A* after raw data exchange from B to *A*.

If the above index is positive, then the raw data exchange is overall beneficial to the estimator *A* in terms of estimation reliability.

Note that StepB.1 and B.2 are executed repeatedly to exhaust all boundary buses $\{b_{i,A}\}$.

C. Search for beneficial raw data to be exchanged from A to B

Now we focus on *B* and extend *B* to include some buses and the corresponding measurements from *A*. We exhaust all boundary buses $\{b_{i,B}\}$ which belonging to *B* and adjacent to *A*. Similar processes as in Step B are executed except that the roles of *A* and *B* in Step B are reversed. Details are omitted.

D. Search for beneficial estimated value to be exchanged between A and B

Suppose in Step B and C we have finalized a set of selected raw data among neighboring estimators to be exchanged in on-line computation, there will be some overlapping areas accordingly after the implementation of the raw data exchanges between estimators *A* and *B*.

For a bus b in the overlapping area between *A* and *B*, if $BCI(b, A) > BCI(b, B)$, then estimated value exchange from *A* to *B* on this bus will be beneficial to B; else if

$BCI(b,B) > BCI(b,A)$, then the estimated value exchange from B to A on this bus will be beneficial to *A*.

Note that BCI(b, S) quantifies the estimation reliability on bus b with respect to system S and higher BCI stands for a higher estimation reliability.

4.6 On-line textured computation

Suppose in off-line texture design we have finalized a set of selected raw data and estimated values among neighboring estimators to be exchanged in on-line computation; then the online textured computation process is as follows:

Step 1: Taking the exchanged raw data into account, the distributed local estimators are extended and therefore there are some overlapping areas between neighboring estimators. Execute the numerous estimators simultaneously and asynchronously until they converge individually to the desired tolerance.

Step 2: If the bad data identification results in local estimators are same on the measurements at the overlapping areas, to to Step 3; otherwise, re-run local estimators with the (more reliable) exchanged estimated data.

For example, suppose a measurement *m-AB* is in the overlapping area of estimator *A* and *B*, i.e. *m-AB* is included in both *A* and *B*. If the estimation reliability of *m-AB* is higher in *A* than that in *B*, then the estimated value of *m-AB* should be exchanged from *A* to *B* according to the principles in off-line texture design.

After estimator *A* and *B* are executed in online Step 1, if *m-AB* is identifies as bad data in *A* but not in *B* or vice versa, then the estimated value of *m-AB* in *A* should be exchanged to *B* to replace the original raw data of *m-AB* in *B*; in addition, this exchanged estimation value should be assigned with a higher reliability in *B* to ensure the bad data identification result on *m-AB* during the rerun of *B* will be the same as that in A.

Note that estimated value exchange is very costly in the sense that one estimator has to wait for the result of the other one. As a DSE algorithm, such a waiting in on-line computation should be avoided whenever possible. Therefore, estimated value exchange given in off-line design is not always applied in on-line computation; instead, output of

Step 1 in online computation decides whether estimated value exchange will be executed.

Step 3: Based on the modified results of local estimators, finally determine the state of the whole system as follows:

Step 3.1: Determine the angle difference of the reference buses between any two local estimators. A reasonable scheme is based on the estimation accuracy of different local estimators, and the scheme is formulated as:

$$\Delta\theta_{AB} = \sum_{i\in I}(\theta_{i,A} - \theta_{i,B})(c_{i,A}^{-1} + c_{i,B}^{-1}) \Big/ \sum_{i\in I}(c_{i,A}^{-1} + c_{i,B}^{-1})$$

where $\Delta\theta_{AB}$ is the angle difference of reference buses between local estimator A and B,

I is the set of buses in the overlapping area between estimator A and B,

$\theta_{i,A}$ is the estimated angle on bus i in estimator A,

$c_{i,A}$ is the i-th diagonal element of covariance matrix $C = G^{-1}$. Note that matrix C stands for the variances of estimation errors on bus i in estimator A. Therefore, the magnitude of $c_{i,A}$ is proportional to the estimation error on bus i in estimator A.

Step3.2: Select a reference bus of one estimator (e.g. A) as the global reference bus for the whole grid.

Step3.3: Determine the angle difference between this global reference bus and reference bus in every local estimator. For local estimators (e.g. B) who connect directly with A, equation (1) alone can give the angle difference directly. However, for local estimators (e.g. C) who only connects the neighboring estimators of A (e.g. B) while estimator C itself does not connects A directly, then the following equation is utilized:

$\Delta\theta_{AC} = \Delta\theta_{AB} + \Delta\theta_{BC}$.

Step3.4: The estimated bus voltage angle of each local estimator will be subtracted with the angle difference between the global reference bus and the local reference bus.

Step3.5: For non-overlapping buses, the state variables have been finalized by now, which is just the current estimation result in local estimators after the previous step.

Step3.6: For overlapping bus i belonging to multiple local estimators $K_j, j = 1,2,...,m$, the state variables x_i are finally determined as:

$$x_i = \sum_{j=1}^{m}(x_{i,K_j} c_{i,K_j}^{-1}) \Big/ \sum_{i=1}^{m} c_{i,K_j}^{-1}$$

where x_{i,K_j} is the state variable of bus i in estimator K_j.

4.7 Advantages of the textured DSE algorithm

Advantage 1: Bad data detection and identification ability in the textured DSE is higher than the existing DSE algorithm, especially when bad data is close to the boundary of individual estimators. Such an improvement is due to the estimated value exchange and the textured network formed by the raw data exchange, which is explained as follows:

In the existing DSE algorithm, the bad data in the boundary of one local estimator A are very difficult to be detected for A. In contrast, the boundary buses in A are internal buses of another estimator (for example B) at the same time in the textured DSE, and therefore these bad data can be detected in B instead. And the corrected estimation value will be exchanged from B to A in Step2 of on-line computation, which enables estimator A to detect the bad data, too. This capability suits well for an industrial environment where it is much more desirable to obtain estimation results with good enough accuracy and without undetected bad data than the results with higher accuracy but with possible undetected bad data.

Advantage 2: Many earlier DSE algorithms assume a star-like function network [37], where the communications between the multiple remote processors and the central computer are critical during iteration processes. Such a hierarchical approach suffers from the bottleneck and reliability issues because of the central controlling node.

On the contrary, the concurrent textured DSE algorithm is asynchronous and there is no central controlling node.

Advantage 3: Utilizing the decoupling nature in SE, we removed the original recursion process in the textured optimization approach [45-46] while the estimation performance remains satisfied as demonstrated in the numerical tests. Consequently, the speed of the textured DSE algorithm gets even faster and further advances our original textured algorithm.

Advantage 4: The multiple local estimators can use different SE algorithms; even the convergence tolerance can be different based on different quality of local measurement system. Accordingly, the textured DSE algorithm becomes very flexible and there is no major change necessary for the current existing estimators.

Advantage 5: The performance of bad data detection and estimation accuracy in the individual existing estimators improves as well, which benefits individual companies/ISOs/RTOs. Accordingly, these entities are more willing to share the information for their own benefits.

4.8 Numerical tests

An IEEE-14 Bus system as shown in Fig.4.3 is used to illustrate how the heuristic approach works, where RTO-A and RTO-B are physically connected. There are two existing local estimators for system *A* and *B* respectively, where neither overlapping areas nor data exchange is involved before the off-line design.

The following examples are given to respectively demonstrate:

1. Not all the data exchange is beneficial. In fact, some data exchange may harm the local estimators in both estimation reliability and estimation accuracy.

2. Estimation reliability of textured DSE can be improved to the level as high as OSE, where the estimation reliability is quantified by BCI.

3. Bad data detection ability in textured DSE improves greatly than existing DSE algorithm.

4. The estimation result of textured DSE is satisfied and comparable to OSE.

A. Example1: Harmful data exchange

RTO B with a data exchange is given in Fig.4.5 and the original system before data exchange is given in Fig.4.4. Clearly such a data exchange does not follow the Principle 2 in Step B of off-line design because there is no loop. Actually this data exchange is

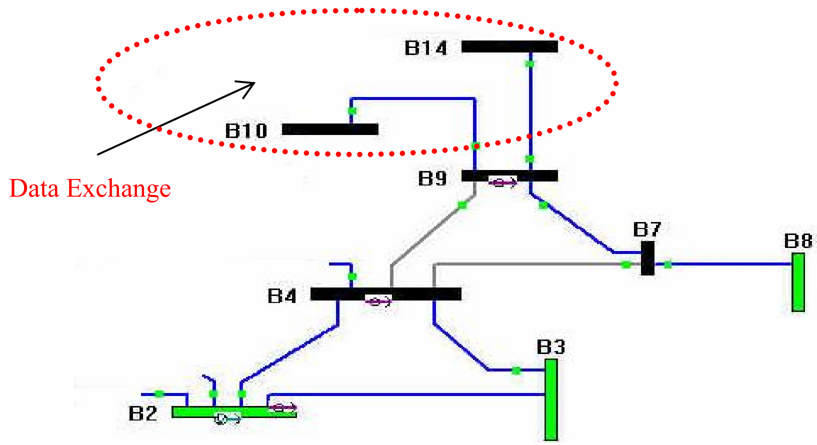


Fig.4.5 Modified system of B after data exchange

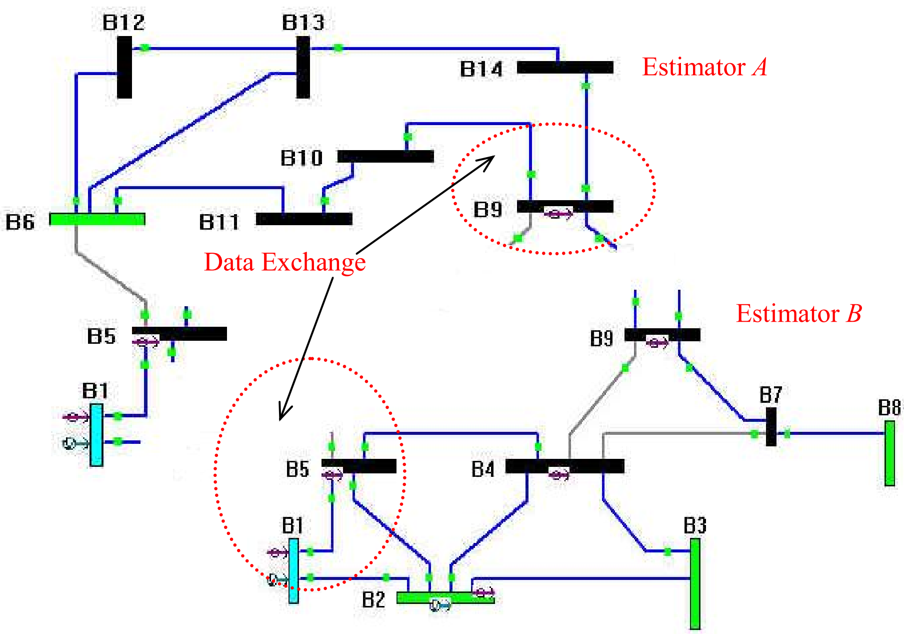


Fig.4.6 Local estimators after raw data exchange

harmful for estimator B, which is demonstrated in the following:

TABLE 4.3 AVERAGE BCI ON THE BUSES OF COMPANY B

B in Fig.4.4	B in Fig.4.5	B in Fig.4.6	B in OSE
0.9647	0.9643	0.9662	0.9662

TABLE 4.4 NORMALIZED RESIDUES FOR LOCAL ESTIMATOR *B* IN EXAMPLE 1

Iteration No.	Estimator B in Fig.4.4		Estimator B in Fig.4.5	
	Meas.	Max. Residue	Meas.	Max. Residue
1^{st}	*M-B9*	164.72	*M-B9B4*	89.41
2^{nd}	*M-B9B7*	108.05	*M-B7B4*	56.78
3^{rd}	*No bad data detected*		*M-B4*	34.68
4^{th}	*N/A*		*No bad data detected*	

The comparison between original B and modified B is given in Table 4.3 and 4.4 (given the bad data probability of any measurement is 0.1).

Table 4.3 implies that the data exchange shown in Fig.4.5 decreases *B*'s BCI, which means such a data exchange decreases estimation reliability. Accordingly, the bad data detection ability gets worse after such a raw data exchange. The following example further demonstrates our conclusion:

Suppose that both measurements, M-B*9* and M-B*9B7,* are bad data in which the sign of these measurements are reversed. Note that when the measurement with largest normalized residue is removed as bad data in WLS algorithm for SE, the SE is executed again to find other possible bad data. Table 4.4 shows the result of such an iteration process. It is clear that before data exchange (Fig.4.4) these two bad data are detected, identified, and removed correctly while after data exchange (Fig.4.5) these bad data cannot be even detected at all.

Note that even though the exchanged raw data are with no bad data in this example, the estimation reliability on local estimator B is still decreased after raw data exchange.

B. Example2: BCI improves after beneficial raw data exchange from A to B

It is reasonable for B4 in B to extend to include B5 in A to form the loop B4-B5-B2-B4 because:

Reasoning 1: BCI(B4, B)=0.97 < BCI(B4, A⋃B)=0.999

Reasoning 2: The new loop B4-B5-B2-B4 only needs to add one new bus B5 from Estimator A.

A can further extends to include B1 because:

Reasoning 1: BCI(B2, B)=0.97 < BCI(B2, A⋃B)=0.999

Reasoning 2: The new loop B2-B1-B5-B4-B2 only needs to add one new bus B1 from Estimator A.

According to Table 4.3, it is clear that B in Fig.4.6 improves the BCI over the buses belonging to original system in Fig.4.4. Furthermore, BCI for B shown in Fig.4.6 is almost as good as the OSE, which indicates little benefit on BCI can be further gained by exchanging more raw data from A to B at this time.

Note that COG of (G, M) instead of G itself is used to search for data exchanges. In this particular example of IEEE-14 bus system, COG is almost equal to G except that branch B6B13 is excluded from COG. Though B6B13 is removed during the off-line textured architecture design, it is still physically part of grid G and therefore is retained in the on-line textured computation as in Fig.4.6.

C. Example3: Bad data identification ability improves in textured DSE

Besides the raw data exchange from B to A as discussed above, raw data exchange from A to B can be executed as follows: It is reasonable for B10 and B14 in A to extend to include B9 in B because:

Reasoning 1: BCI(B14, A)=0.9 < BCI(B14, A⋃B)=0.98 and BCI(B13, A)=0.9 < BCI(B10, A⋃B)=0.99

Reasoning 2: The new loop B10-B9-B14-B13-B12-B6-B11-B10 only needs to add one new bus B9 from Estimator B.

Finally the raw data exchange for A and B is shown in Fig.4.6.

Estimated value exchange between A and B:

Estimation result on bus 1 and 5 are exchanged from *B* to *A* because the corresponding BCI on these buses are higher in B than that in A as follows:

BCI(B1, B)=1.000 > BCI(B1, A)=0.998

BCI(B5, B)=1.000>BCI(B5, A)=0.988

Example 3.1: Bad Data in A

Suppose that M-B11B10 is bad raw data (sign is reversed) in estimator A. Note that measurements with largest normalized residues will be selected as bad data according to WLS algorithm for SE.

For existing DSE without raw data exchange, M-B11B10 can only be detected as bad data in estimator A in Fig.4.4 but can not be identified according to Table 4.5.

For textured DSE with raw data exchange, M-B11B10 is identified successfully in estimator A in Fig.4.6 according to Table 4.5.

TABLE 4.5 NORMALIZED RESIDUES FOR LOCAL ESTIMATOR A IN EXAMPLE 3

Iteration No.	Estimator A in Fig.4.4		Estimator A in Fig.4.6	
	Meas with max residue	Value of Max. Residue	Meas with max Residue	Value of Max. Residue
1st	M-B10	53.38660	M-B11B10	52.40
	M-B11B10	53.38660	M-B10	46.42

TABLE 4.6 NORMALIZED RESIDUES FOR LOCAL ESTIMATOR B IN EXAMPLE 3

Iteration No.	Estimator B in Fig.4.4		Estimator B in Fig.4.6	
	Meas with max Residue	Value of Max. Residue	Meas with max Residue	Value of Max. Residue
1st	M-B2	106.8	M-B2B4	94.9
2nd	M-B4	44.8	M-B2B3	83.4

Example 3.2: Bad Data in B

Suppose that both M-B2B3 and M-B2B4 are bad data (all increase by 0.1 p.u.) in estimator B.

For existing DSE without raw data exchange, M-B2 and M-B4 are selected incorrectly in estimator B in Fig.4.4 as bad data according to Table 4.6.

For textured DSE with raw data exchange, M-B2B4 and M-B2B3 are identified successfully in estimator B in Fig.4.6 one by one according to Table 4.6.

Example 3.3: Bad data in overlapping area

Suppose that both M-B1B5 and M-B5B1 are bad data (all increased by 0.1 p.u.) in the overlapping area between estimator A and B.

In Step1, local estimators are run separately. Even for estimator A with raw data exchange as shown in Fig.4.6, M-B1 and M-B5 are still selected incorrectly as bad data according to Table 4.7. Simultaneously, estimator B in Fig.4.6 is executed where M-B1B5 and M-B5B1 are both identified as bad data successfully one by one based on table 4.7.

In Step2, since the bad data identification result are different in the overlapping area {B1, B5, B9}, estimated value exchange becomes necessary. Since estimation results on M-B1B5 and M-B5B1 are more reliable in estimator B, these corrected values on M-B1B5 and M-B5B1 are exchanged from B to A, which follows the estimated data exchange scheme mentioned before. And these values are treated in estimator A with particular high reliability so that A will not select them as bad data any more. Taking the new pseudo measurements into account, estimator A is re-run and this time M-B1B5 and M-B5B1 are both identified successfully as bad data according to Table 4.8.

TABLE 4.7 NORMALIZED RESIDUES FOR LOCAL ESTIMATOR A AND B

Iteration No.	A in Fig.4.6		B in Fig.4.6	
	Meas with max Residue	Value of Max. Residue	Meas with max Residue	Value of Max. Residue
1st	M-B1	68.95	M-B5B1	87
2nd	M-B5	59.4	M-B1B5	84

TABLE 4.8 NORMALIZED RESIDUES FOR LOCAL ESTIMATOR B DURING ITS RE-RUN

Iteration No.	A in Fig.4.6 with estimated value exchange	
	Meas with max Residue	Value of Max. Residue
1st	M-B5B1	84
2nd	M-B1B5	94

D. Example4: Estimation result comparison between OSE and the textured DSE

The IEEE-30 bus system with the given measurement system is given in Fig.4.7, where Estimator A includes buses from B1 to B11 and Estimator B includes the other buses.

Raw data exchange from B to A:

A should be extended to include B12, B16 and B17 to form a new loop B4-B12-B16-B17-B10-B6-B4 according to the three principles in Step B of the off-line texture design:

Principle 1: BCI(B4, A)=0.614 < BCI(B4, A⋃B)=0.959

Principle 2: Loop B4-B12-B16-B17-B10 only needs to add three new buses into A

Principle 3: The other loop candidate will need to add more buses. For example, the loop B4-B12-B15-B18-B19-B20-B10-B6-B4 is not preferred because five buses need to be added.

Similarly A should also be extended to include B28 to form a new loop B8-B28-B6-B8.

Raw data exchange from A to B:

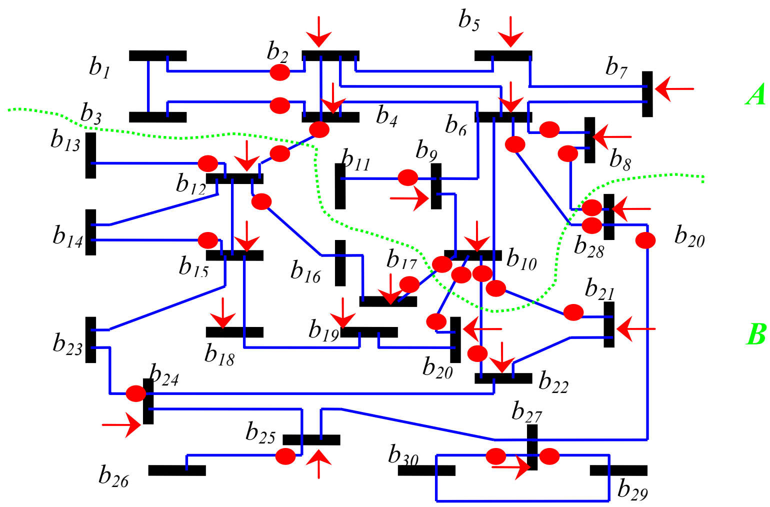


Fig. 4.7 The IEEE 30-bus network *G* with given measurement

B is extended to include {B4, B6, B10}.

Since no bad data is artificially introduced during this simulation, Step 2 utilizing estimated value exchange will not be executed and therefore the details of the suggested estimated value exchange are omitted for this example.

Table 4.9 gives the comparison between the textured DSE and OSE, which indicates the estimation result of textured DSE over the whole system is as good as that of OSE. Furthermore, the difference in the boundary buses {B4, B6, B8, B10, B12, B17, B20, B21, B28} between DSE and OSE is as small as that in other buses.

TABLE 4.9 ESTIMATION RESULT COMPARISONS BETWEEN OSE AND TEXTURED DSE

Bus	OSE		Textured DSE		ABS of Difference	
	θ (rad.)	|V|	θ (rad.)	|V|	|Δθ| (rad.)	|Δ|V||
1	0	1.06	0	1.0599	<1E-04	1E-04
2	-0.0939	1.045	-0.0939	1.0449	<1E-04	1E-04
3	-0.1316	1.0212	-0.1316	1.0211	<1E-04	1E-04
4	-0.1621	1.0123	-0.1621	1.0123	<1E-04	<1E-04
5	-0.247	1.01	-0.247	1.0099	<1E-04	1E-04
6	-0.1931	1.0106	-0.1931	1.0105	<1E-04	1E-04
7	-0.2245	1.0025	-0.2245	1.0025	<1E-04	<1E-04
8	-0.2061	1.01	-0.2061	1.0099	<1E-04	1E-04
9	-0.2464	1.0511	-0.2465	1.0511	1E-04	<1E-04
10	-0.2742	1.0454	-0.2741	1.0454	1E-04	<1E-04
11	-0.2466	1.0818	-0.2466	1.0817	<1E-04	1E-04
12	-0.2609	1.0575	-0.2609	1.0575	<1E-04	<1E-04
13	-0.261	1.0711	-0.2609	1.0713	1E-04	0.0002
14	-0.2763	1.0425	-0.2762	1.0427	1E-04	0.0002
15	-0.278	1.038	-0.2779	1.0381	1E-04	1E-04
16	-0.2712	1.0448	-0.2712	1.0448	<1E-04	<1E-04
17	-0.2771	1.0403	-0.277	1.0402	1E-04	1E-04
18	-0.289	1.0283	-0.2888	1.0284	0.0002	1E-04
19	-0.292	1.0258	-0.2918	1.0259	0.0002	1E-04
20	-0.2885	1.03	-0.2883	1.03	0.0002	<1E-04
21	-0.2819	1.033	-0.2817	1.033	0.0002	<1E-04
22	-0.2817	1.0336	-0.2815	1.0335	0.0002	1E-04
23	-0.2848	1.0273	-0.2847	1.0274	1E-04	1E-04
24	-0.2879	1.0218	-0.2877	1.0218	0.0002	<1E-04
25	-0.2806	1.0173	-0.2801	1.0173	0.0005	<1E-04
26	-0.2879	0.9985	-0.2875	0.9986	0.0004	1E-04
27	-0.2716	1.0235	-0.271	1.0235	0.0006	<1E-04
28	-0.204	1.0071	-0.204	1.007	<1E-04	1E-04
29	-0.2932	1.0041	-0.2926	1.004	0.0006	1E-04
30	-0.3094	0.992	-0.3088	0.9919	0.0006	1E-04

4.9 Conclusions

A recent trend for ISOs/RTOs is to further merge to become a bigger Mega-RTO grid. The determination of state over the whole system becomes very challenging due to its size. Instead of starting a totally new estimator over the whole grid, a concurrent textured distributed state estimation algorithm is proposed that consists of the off-line textured architecture design and the on-line textured computation. Note that the off-line design is to determine the textured architecture while the on-line computation relies on the raw data or estimated value exchanges among neighboring entities through the communication in the textured architecture, fully takes advantage of the existing local state estimators and ultimately determines the system state over the whole system.

A heuristic approach is proposed to search for beneficial data exchanges for off-line design. Numerical tests on IEEE-14 and IEEE-30 bus system demonstrate that properly selected data exchange improves the estimator reliability of all entities. It is also shown that the benefit of different data exchanges can be quite different. Properly selected data exchanges will enable the estimation reliability of the local estimators to be as good as one estimator on the whole system. On the other hand, poorly designed data exchanges not following our design principles may be harmful to local estimators.

The textured DSE is non-recursive, asynchronous and avoids central controlling node. Furthermore, numerical tests verify that the performance of the new textured DSE algorithm improves greatly compared with existing DSE algorithms in respect of bad data detection and identification.

CHAPTER V

A SEMANTIC BASED SOFTWARE ARCHITECTURE FOR POWER MARKET INFORMATION INTEGRATION

5.1 Introduction

As discussed in Topic 4 of Chapter I, a semantic based software architecture is proposed in order to meet the three major concerns, where information sources are integrated into a regionally regulated semantic knowledge warehouse (SKW). Based on the above SKW, a semantic reasoning software component (SRSC) is further developed to automatically deduct new information based on a deductive reasoning algorithm.

This chapter is organized as follows: the overall picture of the proposed semantic based software architecture is described in Section 2. The SKW and SRSC are studied in Section 3 and 4 respectively. In Section 5 the deduction credibility index is discussed to further refine the SRSC. We conclude in the last section.

5.2 Semantic based software architecture

A. Concept of component [66, 67]

Reusable software components are self-contained, clearly identifiable pieces that describe and perform specific functions. Software components have clear interfaces, appropriate documentation, and a defined reuse status. The term 'pieces' in the definition indicates that components can have a variety of different forms, such as source code, documentation, executable code, and test plans, etc. It can be used as the vehicle for encapsulation and data hiding. In other words, component provides a basic reusable unit, and it is better than a function in terms of software reusability. There are already many existing commercial components.

B. Three classes of information sources

Information sources are classified into the following three classes in terms of how the information is structured:

● Class 1: Unstructured information sources

So far, most HTML web pages, such as personal homepages and the search result from a public search engine, are unstructured information sources, whose contents are difficult to be separated into data and their descriptions. Therefore, how to mine information from unstructured web pages remains challenging in the field of computer science [68, 69].

● Class 2: Syntax structured information sources

A syntax-structured information source is similar to a table with attributes name and tuples, where XML and frameworks supporting web services such as J2EE and .NET are widely applied. The content of these information sources has been separated into data and their descriptions.

In power system, XML has been widely utilized for addressing the operational data exchange needs of the industry. However, with syntax structured information sources, machines still do not know the semantic meaning of the attribute names. Therefore, the application is still tightly connected and limited to some specific information formats and therefore is hardly reusable and expandable. For example, the software for measurement data exchange is quite similar to estimated value exchange in terms of semantic reasoning; however, the software is hardly reusable because their table names, attribute names and the internal relationship in the information source are quite different in terms of syntax. In order to further improve the reusability of the software, semantic structured information sources are proposed as the following.

● Class 3: Semantically structured information sources

A semantically structured information source includes a semantic description of the data in an overall scale, where the knowledge is represented in a semantic based computer language such as RDF, SHOE or DAML+OIL [70, 71, 72] and therefore the information is understandable to computers. In other words, the distributed information sources are turned into a regionally regulated machine-readable and machine-understandable database, and the computers make their own reasoning process based on the given semantic represented data.

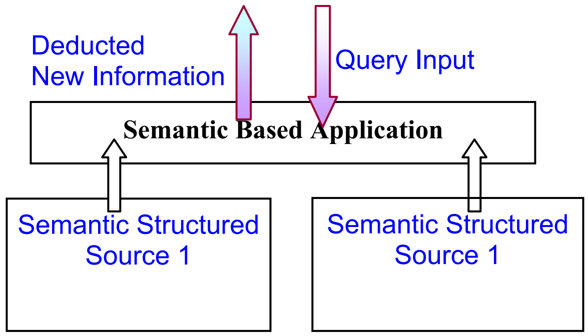


Fig.5.1 Traditional semantic-based software architecture

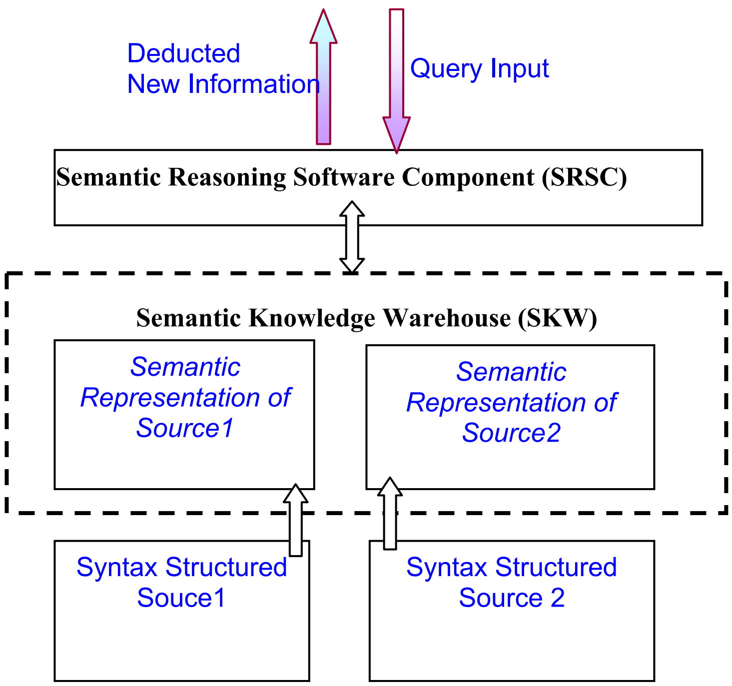


Fig.5.2 Refined semantic based software architecture

So far, XML has been widely utilized in power industry to implement the syntax-structured information sources, but few efforts are made on the semantic structured information sources (class 3).

C. The semantic based software architecture

The software architecture in Fig.5.1 is a typical semantic based software architecture [73], where the information source is in semantic structured (class 3) and semantic based application can be developed further.

However, since there is not yet a widely accepted language for semantic information source and application, very few information sources are semantically structured. In addition, the power industry is still unsure about the benefit of being semantically structured; therefore they prefer to keep the existing information sources as syntax structured. Furthermore, most current research in computer science focuses on how to develop and standardize a language to implement a semantic web; the detailed application and benefit of semantic information source are not well demonstrated, even though many conceptual comments on these benefits have been discussed [73, 74, 75].

Considering the above issues, we propose a refined semantic based software architecture as shown in Fig.5.2. It consists of a semantic knowledge warehouse (SKW) and a semantic reasoning software component (SRSC). Such an architecture is more suitable for gradual commercial evolution because:

● The SKW is regulated regionally in the power market domain and consists of numerous semantic representations of syntax structured information sources. In a power market, either a for-profit or nonprofit organization may initiate such a global SKW. It can also be part of the market protocol design and therefore regulated by ISO/RTO. ISO/RTO can further set up some rules within SKW to carefully determine whether the information is accessible as a typical problem of information security.

● Most information sources are already syntax structured with a commercial database system running in the background. Instead of waiting for the information sources to become semantic structured by a standard semantic language, the implementation of the semantic representations for the SKW is much easier since they are within the same domain and serves only for SKW.

Note that an information source possibly has more than one semantic representation. For examples, a specific utility accounting department as an information resource can be used to check the price and quantity of the generation. It can also be used to judge if the load data in the EMS is reasonable; and it can also be used to tell the daily lowest bidding price.

● SRSC is based on SKW as a virtual machine to handle reasoning requests. SRSC can be applied in different SKWs of different domains while the code remains the same. Therefore, the software vendors are more willing to work out such a generalized SRSC.

5.3 Semantic Knowledge Warehouse (SKW)

A. Overall construction process of SKW

The process to construct a SKW consists of three stages.

In stage 1, the global semantic dictionary is designed for the whole domain.

In stage 2, the input is the syntax structured information sources and the output is their semantic representations under the global dictionary set up in stage 1.

In stage 3, the output of stage 2 is further organized to enrich the knowledge connection, where term 'knowledge connection' refers to the logical relationship among the numerous semantic representations.

B. Stage 1 of SKW construction process

In stage 1, term 'ontology' [74] serves as the atom to set up the global semantic dictionary for the whole domain.

Ontology is a description of the concepts and relationships that exist for a model item. Based on the analysis of common elements in the multiple ontology languages for semantic web, the following terms [71, 72] are proposed to describe the ontology in the SKW:

Attribute definition: The attributes of the ontology are defined, where term 'attribute' is also known as 'property' in RDF/DAML+OIL [71].

Assertion knowledge: The ontology has many assertions, where the information on

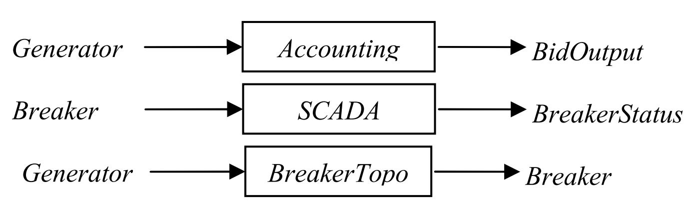


Fig. 5.3 The global semantic dictionary for the SKW

some attributes of the ontology serves as the input and the information on some other attributes are derived and serves as the output accordingly.

For example, as shown in Fig.5.3, a semantic dictionary is set up, where ontologies Accounting, BreakerTopo and SCADA (Supervisory Control and Data Acquisition) are defined.

Ontology Accounting has attributes Generator and BidOutput. In addition, an assertion knowledge that based on the information of Generator we can get the information of the corresponding BidOutput is set up as shown in Fig.5.3.

Ontology BreakTopo is from the network modeling application, where the connected breaker can be determined given the generator name.

Ontology SCADA is based on the measurement value and is able to give the status of a breaker with a given breaker name. Since measurements can have errors while so far most SCADA does not have topology detection capability, the information from ontology SCADA can thus be wrong, which motivates the concept of Deduction Credibility as discussed later in Section 5.5.

C. Stage 2 of SKW construction process

In the second stage, the contents of syntax web sites are transformed into their semantic representations under the overall dictionary.

Note that these semantic representations will be utilized only inside the SKW, and therefore one can choose any semantic languages to represent the information semantically. One can even develop its own semantic language without waiting for a standard language, and thus is more suitable for gradual commercial evolution.

The following terms [71, 72] are applicable to describe the semantic representations in SKW:

Information source instance knowledge: A specific ontology contains more than one information source instance. For example, every individual utility company is an instance of Utility Ontology in the power market domain.

Attribute mapping knowledge: The data of the information sources are mapped into the attributes of the ontology. Even if the syntaxes of the data for different information sources are different, their mapped ontology attribute can be the same.

With the same example in Fig.5.3, even if there are more than one SCADA systems in a power market, all of them are semantically integrated under a unified ontology *SCADA*. The syntax to represent the name of breaker can be different in different SCADA systems, but all of them are mapped into the same attribute *Breaker* in ontology *SCADA*.

D. Stage 3 of SKW construction process

In stage 3 the numerous ontologies in SKW are analyzed and organized based on 'Ontology Connection Knowledge'.

Ontology connection knowledge: that tells the equivalent relationship between the attributes of different ontologies in the SKW and bridges the numerous ontologies accordingly.

With the same example in Fig.5.3, the attribute Generator in ontology Accounting and the attribute Generator in ontology BreakerTopo are equivalent with each other since they are semantically the same, i.e. the information for attribute Generator in ontology Accounting leads to the information for attribute Generator in ontology BreakerTopo, and vice versa. Similarly the attribute Breaker in ontology BreakerTopo and the attribute Breaker in ontology SCADA are equivalent. Finally the SKW with the ontology connection knowledge is given in Fig.5.4.

E. SKW in power market information integration

Fig.5.5 [8] shows a typical system entity relationship in a power market. Every entity consists of one or numerous independent information source. There are basically two types of data flow: market data flow and operation data flow. Accordingly, for many

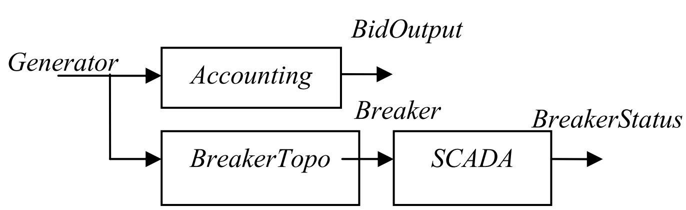


Fig. 5.4 The SKW with ontology connection knowledge

entities we need at least two different semantic representations: one for market information integration and the other for system operation.

Considering the numerous entities with different roles in Fig.5.5, how to successfully design a global semantic dictionary for SKW in electric power market is a big challenge, which is still under discussion in IEC TC57 Working Group 16 [75].

The SKW based software architecture in Fig.5.2 has the following advantages for power market information integration:

● The entities in power market may be reluctant to implement the semantic wrappers. Therefore, a centralized SKW is more feasible to be initiated by the control area authority as part of the standard market design.

● The configurations and rules of power market and operation are evolving fast; and thus it is not realistic for all entities to modify their semantic representation whenever the data flow or knowledge changes due to the rule changes. Instead, only the SKW level needs to be changed in Fig.5.2 while the information sources in the entities remain the same.

● The market data flow and operation data flow are tightly related logically. Though

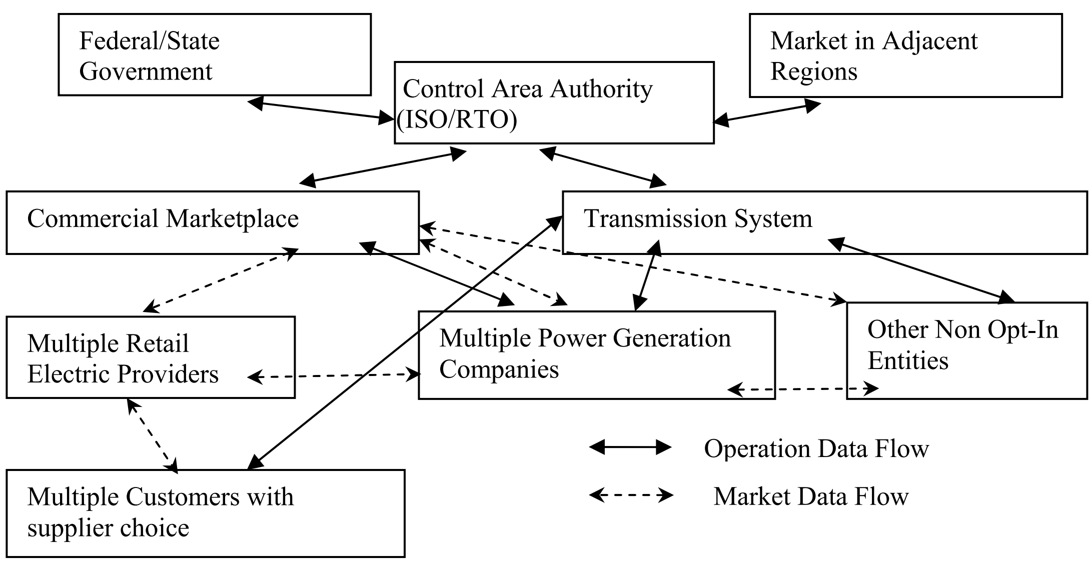


Fig.5.5 System entity relationships in a typical power market

they are quite different in terms of syntax, many applications may need both of them. The software architecture in Fig.5.2 can have two centralized SKWs, one for market data flow and one for operation data flow. Then a bridge among them can be developed to define their logical relationships.

● If the syntax of the information sources changes, say, a utility changes their EMS platform, then only the attribute mapping needs to be changed. No matter how the syntax changes, their semantic representations in SKW remain the same, which indicates the proposed software structure is flexible.

● New emerging information sources are able to be integrated into the SKW easily since only the information source instance knowledge needs to be updated. In other words, the proposed software structure is expandable.

5.4 Semantic Reasoning Software Component (SRSC)

Semantic reasoning software component (SRSC) is to derive new information based on the SKW via deduction reasoning.

A. Deduction reasoning algorithm

A deduction reasoning algorithm is fulfilled via the reasoning function: $V_D=f(O_S, A_S, V_S, O_D, A_D)$

Where O_S: Starting point ontology

A_S: attribute of O_S

V_S: value for A_S

O_D: Destination ontology

A_D: attribute of O_D

V_D: value for A_D

In other words, for a given ontology attribute A_S in the starting point O_S and its value V_S, the function can derive the value V_D for ontology attribute A_D in the destination ontology O_D.

Since SKW has given the necessary information on O_S, A_S, O_D, and A_D, the reasoning function can be rewritten as: $V_D=f(V_S, SKW)$

The problem can be divided into two sub-problems: one is whether V_D can be deducted by V_S, and the other is how to deduct V_D.

Note that SKW in Fig.5.4 is actually a directed graph, where the node is the attributes and the directed path is the assertion in the corresponding ontology. Therefore, the first sub-problem is transformed to a typical path finding problem in SKW, which can be solved by the standard algorithms in graph theory to return the directed path that leads from A_S in SKW to A_D.

If there is no such a path, then it indicates that it is impossible to deduct from A_S to A_D within the given SKW. On the other hand, sometimes there are more than one deduction path and we need to choose the best, which motivates the concept of deduction credibility. Deduction credibility is discussed in detail in Section 5.

Suppose the first sub-problem is solved and we find out there exist a deduction path $A_S \rightarrow A_1 \rightarrow A_2 \rightarrow \ldots \rightarrow A_{n-1} \rightarrow A_n \rightarrow A_D$, then as defined in Section 3, an assertion in an ontology can deduct the information of the output attribute with the given information of the input attribute. Therefore, we can transform the information from one attribute to another via the assertion path along the deduction path and eventually get the information of A_D with the given information of A_S.

B. Discussions

Obviously SKW plays a critical role for the reasoning in SRSC since SKW determines the directed graph, over which the reasoning algorithm is applied. Even a minor change in SKW may lead to a different deduction conclusion. Therefore, even though the SRSC can be developed by the software vendor, the design of SKW heavily depends on the cooperation from all the participants of the power market.

SRSC is a general reasoning tool and can be applied on SKWs of different domains without code modification, which is very attractive for the software vendors.

Note that the detailed data are still stored and maintained in the independent information sources, which is preferred for the market participants because they want to keep the privacy of their business.

C. Example

SRSC can be used in power market to implement the dynamic input generation mechanism, which is discussed as follows.

● Motivation of Dynamic Input Generation

A non-semantic based application works similarly to a function, where both the attributes of input and output are fixed.

However, such a predefined and static way is not flexible and extendible for a dynamic environment in power market. For example, with the changing market rules, some originally necessary input data may become unnecessary and the software code needs to be modified. Furthermore, it is even more difficult for the applications to adapt the changes in the syntax of the independent and distributed information sources.

With the SRSC running over the SKW, once the wanted output attributes are given, the necessary input attributes can be determined dynamically to avoid the above disadvantages.

The key of dynamic input generation mechanism is that an input attribute should be excluded from the input attributes list if it can be deducted from other attributes in the list. In other words, the dynamic input generation mechanism determines the minimum necessary input attributes to get the output.

● Example of Dynamic Input Generation

Suppose a particular generator involves in the bidding and its output is nonzero, then surely the status of the generator's connected breaker is physically 'on'. However, the status of the breaker from SCADA is based on the measurement and thus can be 'off' when the measurement is anomalous. Therefore, in order to check whether the measurement works fine, a user is trying to check whether the status of the breaker from SCADA is 'on'.

First of all, the user needs to tell SRSC what are wanted, i.e. the semantic attributes of the expected output information under the context of SKW. Suppose all the attributes in the SKW are graphically displayed for selection, users can even simply click on them to choose the wanted.

With the SKW shown in Fig.5.4, what the user needs to know are the attribute 'Bidding output' in ontology 'accounting' and the attribute 'breaker status' of the connected breakers in ontology 'SCADA'.

Then SRSC is activated to find out the necessary input attributes as the followings:

Step1: Based on the given output attributes and SKW in Fig.5.4, SRSC finds out that attribute 'Generator' in ontology 'Accounting' and the attribute 'Breaker' in ontology 'SCADA' are able to derive all the output attributes.

Step2: Attribute 'Breaker' in ontology 'SCADA' is excluded from the necessary input attributes list because there exists an assertion in the ontology 'BreakerTopo' to derive attribute 'Breaker' from attribute 'Generator', where attribute 'Generator' belongs to the necessary input attributes list in Step 1.

As a conclusion, SRSC automatically deducts that the attribute "Breaker" is only an intermediate result and the user needs to provide information on attribute "Generator" only.

Though the reasoning in the above example is not difficult for a human being, it needs to be accomplished automatically by the computer, which may not be as easy. For more complicated reasoning, it can be very difficult for human being but the computer can still carry it out routinely.

After the dynamic input generation reasoning is activated and completed via SRSC, the minimum necessary input information is determined. Then the user needs to supply the information for the input attributes, i.e. the particular name of the generator in this example, to finally get the output information.

5.5 Deduction credibility

A. Motivation

As mentioned in last section, sometimes there are more than one deduction path from the same starting attribute to the same destination attribute in SKW. For example, the generator output can be determined via:

a) Real time measurement from ontology SCADA;

b) Estimation results from ontology State Estimation;

c) Bidding output from the ontology Accounting.

d) Fuel consumption based estimation from the ontology Logistical.

Obviously the credibility of the above deduction paths is different. Therefore, it is critical to evaluate the credibility for an information *b* with a deduction path *S*.

B. Determination of Deduction Credibility Index (DCI)

For information *b* with a particular deduction-reasoning path *S*, the credibility is determined as:

$$DCI(b,S) = 1 - P(C_1 \cup C_2 \cup \cdots \cup C_k)$$

$$= 1 - \sum_{i=1}^{k}\left((-1)^{i-1} \sum_{1 \le t1 < t2 < \cdots < ti \le k} P(C_{t1} \cap C_{t2} \cap \cdots \cap C_{ti}) \right)$$

$$= 1 - \Big(\sum_{1 \le t1 \le k} P(C_{t1}) - \sum_{1 \le t1 < t2 \le k} P(C_{t1} \cap C_{t2}) + \sum_{1 \le t1 < t2 < t3 \le k} P(C_{t1} \cap C_{t2} \cap C_{t3}) - \cdots + (-1)^{k-1} P(C_1 \cap C_2 \cap \cdots \cap C_k) \Big) \quad (1)$$

Where

$DCI(b,S)$ is the Deduction Credibility Index of information *b* with respect to the deduction path *S*;

S consists of *k* ontology assertions C_i, i=1,2,3,…*k*;

$P(C_i \cap Cj)$ is the credibility that both assertion C_i and C_j are incorrect;

$P(C_i \cup Cj)$ is the credibility that either C_i or C_j is incorrect.

If the assertion credibility is independent from each other, $P(C_i \cap Cj)$ can be determined by:

$$P(C_i \cap Cj) = P(C_i) \cdot P(Cj) \quad (2)$$

where $P(C_i)$ is the assertion credibility of C_i, and it is a real value attached to the assertion assigned by SKW as part of the SKW design.

$DCI(b,S)$ can be determined with given $P(C_i)$ according to (1) and (2).

C. Two types of knowledge

Knowledge *K* is called Discrete Knowledge if its possible values are discrete and finite. An example is the status of a breaker, which is either open or closed.

On the other hand, Knowledge *K* is called Continuous Knowledge if its possible values are infinitely many in a continuous range. An example is the output of a generator, which can be any real value between the upper output limit and lower output limit if the generator has been started up.

For knowledge K, we suppose K can be deducted from different deduction paths S_i, where i=1, 2, .., n. In addition, for the deduction path Si that leads to information bi, we assume the deduction credibility index is $DCI(b_i, S_i)$. Then the information b for knowledge K can be set as bm from the most reliable deduction path S_m, where $DCI(b_m, S_m)$ is the maximum of $DCI(b_i, S_i)$.

D. Dynamic assertion credibility determination

*T*he credibility of an assertion needs to be increased if it always gives the correct information.

With such a mechanism, the information sources are encouraged to maintain their information to improve the credibility so that their information will be more likely applied in the whole market. The ISO/RTO also can request the distributed information sources to maintain the credibility at a reasonable level. Furthermore, such a mechanism helps the information sources to detect and identify the potential error when the credibility is lower than the expected.

E. Example

A typical continuous knowledge is the generator's output value which can be deducted from the ontology 'State Estimator' or 'Accounting' respectively.

A typical discrete knowledge is the trend of the company's generator bidden output in the ontology Accounting with the 'Multiple Generation Companies' in Fig.5.5 as the information sources. This knowledge is either 'increased' or 'decreased'.

Another typical discrete knowledge is the trend of the company's generator output in the ontology State Estimator with the 'Control Area Authority' in Fig.5.5 as the information source. This knowledge is also either 'increased' or 'decreased'.

Generally speaking the trend information in ontology accounting and state estimator are consistent with each other, which is helpful for state estimator to detect and identify bad data in the physical measurement system from the 'Transmission System' in Fig.5.5.

5.6 Conclusions

Information integration has become very challenging during electric power deregulation. A new software architecture is proposed based on the concept of semantic ontology, where information sources are integrated into a semantic knowledge warehouse (SKW). A semantic reasoning software component (SRSC) is developed accordingly to automatically deduct new information. Such a software architecture has many advantages during the information integration in power market. In addition, the concept of deduction credibility is proposed to further refine the reasoning process in SRSC.

CHAPTER VI

CONCLUSIONS

6.1 On network observability evaluations under a contingency

A novel topological approach is proposed to identify critical measurements and to examine network observability under a contingency, where the contingency can be any single branch outage or measurement loss. To advance the classical topological observability analysis, a new concept of contingency observability graph (COG) is introduced and it is proven that a power system network maintains its observability under a contingency if and only if its COG satisfies some conditions. With little extra computation load, the proposed COG based approach is a natural extension of the classical topological observability analysis for online applications. The IEEE 30-Bus system is used to demonstrate the validity and efficiency of our approach.

6.2 On placement design for measurements and RTUs against a contingency

A two-stage topological approach is proposed to minimize the number of measurements and RTUs under the constraint that the system remains observable under any single contingency. The contingency can be a single branch outage from a pre-selected branch set or a single measurement loss. In addition, a single RTU loss is taken into account as a type of contingency. The approach is based on the concepts of qualified COG (QCOG) and preferred QCOG, and an algorithm to search for a preferred QCOG is developed and a heuristic measurement and RTU placement methodology is proposed based on the preferred QCOG. The IEEE 30-Bus system is used to demonstrate the validity and efficiency of our approach.

6.3 On a textured distributed SE algorithm based on data exchange

A recent trend for ISOs/RTOs is to further merge to become a bigger Mega-RTO grid. The determination of state over the whole system becomes very challenging due to its size. Instead of starting a totally new estimator over the whole grid, a concurrent textured distributed state estimation algorithm is proposed that consists of the off-line textured architecture design and the on-line textured computation. Note that the off-line design is to determine the textured architecture while the on-line computation relies on the raw data or estimated value exchanges among neighboring entities through the communication in the textured architecture, fully takes advantage of the existing local state estimators and ultimately determines the system state over the whole system.

A heuristic approach is proposed to search for beneficial data exchanges for off-line design. Numerical tests on IEEE-14 and IEEE-30 bus system demonstrate that properly selected data exchange improves the estimator reliability of all entities. It is also shown that the benefit of different data exchanges can be quite different. Properly selected data exchanges will enable the estimation reliability of the local estimators to be as good as one estimator on the whole system. On the other hand, poorly designed data exchanges not following our design principles may be harmful to local estimators.

The textured DSE is non-recursive, asynchronous and avoids central controlling node. Furthermore, numerical tests verify that the performance of the new textured DSE algorithm improves greatly compared with existing DSE algorithms in respect of bad data detection and identification.

6.4 On a semantic based software architecture for information integration

Information integration has become very challenging during electric power deregulation. A new software architecture is proposed based on the concept of semantic ontology, where information sources are integrated into a semantic knowledge warehouse (SKW). A semantic reasoning software component (SRSC) is developed accordingly to automatically deduct new information. Such a software architecture has many advantages during the information integration in power market. In addition, the concept of deduction credibility is proposed to further refine the reasoning process in SRSC.

REFERENCES

[1] F. C. Schweppe and E. J. Handschin, "Static state estimation in electric power systems," *IEEE Proc.*, Vol.62, pp. 972-983, July 1974.

[2] F. F. Wu, "Power system state estimation: A survey," *Int. J. Elect. Power Eng. System*, vol. 12, pp. 80–87, Jan. 1990.

[3] A. Monticelli, "Electric Power System State Estimation," *Proceedings of the IEEE*, Vol. 88, No.2, pp. 262-282, February 2000.

[4] G. R. Krumpholz, K. A. Clements and P. W. Davis, "Power System Observability: A practical algorithm using network topology," *IEEE Trans. PAS*, Vol. 99, No.4, pp. 1534-1541, July 1980.

[5] D. Shirmohammadi, B. Wollenberg, B. Vodjani, P. Sandrin, M. V. F. Pereira, F. Schneider, and B. Stott, "Transmission dispatch and congestion management in the emerging energy market structures," *IEEE Trans. Power Syst.*, vol. 13, pp. 1466–1474, Nov. 1998.

[6] Iraj Dabbaghchi, Louis S. VanSlyck, "Inter-Utility Data Exchange for State Estimation," *IEEE Trans. On Power systems*, Vol. 3, No. 3, August 1988, pp. 1254- 1262

[7] Z. Alaywan and J. Allen, "California electric restructuring: A broad description of the development of the California ISO," *IEEE Trans. Power Syst.*, vol. 13, pp. 1445–1452, Nov. 1998.

[8] Electrical Reliability Council of Texas, Inc., "Company Profile," *http://www.ercot.com*, accessed on April 2, 2007.

[9] Narayan S. Rau, "Issues in the Path Toward an RTO and Standard Markets," *IEEE Trans. On Power Systems*, Vol. 18, No. 2 May 2003, pp. 435-443

[10] M. A. Lamoureux, "FERC's Impact on Electric Utilities," *IEEE Power Engineering Review*, pp. 8-11, August 2001.

[11] Hyde M. Merrill, "The RTO debate," *IEEE Power Engineering Review*, Vol. 20, no. 7, p. 9, July 2000

[12] F. H. Magnago and A. Abur, "A unified approach to robust meter placement against loss of measurements and branch outages," *IEEE Trans. Power Systems*, Vol.15, No.3, pp. 945-949, August 2000.

[13] A. Abur and F. H. Magnago, "Optimal meter placement for maintaining observability during single branch outages," *IEEE Trans. Power Systems*, Vol. 14, No. 4, pp. 1273-1278, November 1999.

[14] A. Monticelli and F. F. Wu, "Network Observability: Theory," *IEEE Trans. PAS*, Vol. 104, pp. 1042-1048, May 1985.

[15] K.A. Clements, G.R. Krumpholz and P.W. Davis, "Power System State Estimation Residual Analysis: an algorithm using network topology," *IEEE Trans. Power Apparatus and Systems*, Vol. 100, No. 4, pp. 1779-1787, April 1981.

[16] A. Simoes Costa, T. S. Piazza and A. Mandel, "Qualitative methods to solve qualitative problems in power system state estimation," *IEEE Trans. Power Systems*, Vol. 5, No. 3, pp. 941 - 949, August 1990.

[17] J. B. A. London, L. F. C. Alberto and N. G. Bretas, "Network observability: a fast topological approach to identify critical measurements," *Proceedings of International Conference on Power System Technology 2000*, Vol. 2, pp. 583-588, December 2000.

[18] J. B. A. London, L. F. C. Alberto and N. G. Bretas, "Network observability: identification of the measurements redundancy level," *Power System Technology*, Proceedings 2000, pp.577-582.

[19] A. Al-atwan and H.-J.Koglin, "Two aspects of local redundancy in state estimation," *IEE Proc. Generation, Transmission and Distribution*, Vol. 145, No. 4, July 1998. pp 458-462.

[20] G. N. Korres, G. C. Contaxis, "Identification and Updating of Minimally Dependent Sets of Measurements in State Estimation," *IEEE Trans. On power systems*, Vol. 6, No.3, August 1991, pp 999-1005

[21] E. D. Crainic, H. P. Horisberger, X. D. Do. , D. Mukhedkar, "Power Network Observability: The assessment of the measurement system strength," *IEEE Trans. On Power Systems*, Vol. 5, No. 4, November 1990, pp. 1267-1285

[22] K. A. Clements, "Observability methods and optimal meter placement," *International Journal of Electrical Power & Energy System*, Vol. 12, No. 2, pp. 88-93, April 1990.

[23] K. Clements, G. Krumpholz and P. Davis, "Power System State estimation with measurement deficiency – An observability measurement placement algorithm," *IEEE Trans. PAS*, Vol. 102, pp. 2012-2020, 1983.

[24] M. Park,, Y. H. Moon, J. B. Choo, and T. W. Kwon, "Design of reliable measurement system for state estimation," *IEEE Trans. Power Systems*, Vol. 3, No.3, pp. 830-836, August 1988.

[25] M. E. Baran, J. Zhu, H. Zhu, and K. E. Garren, "A meter placement method for state estimation," *IEEE Trans. Power Systems*, Vol. 10, No. 3, pp. 1704-1710, August, 1995.

[26] M. K. Celik and W.-H. E. Liu, "An incremental measurement placement algorithm for state estimation," *IEEE Trans. Power Systems*, Vol.10, No.3 , pp.1698-1703, Aug. 1995.

[27] G. N. Korres and G. C. Contaxis, "A tool for the evaluation and selection of state estimator measurement schemes," *IEEE Trans. Power Systems*, Vol.9, No.2, pp.1110-1116, May 1994.

[28] H. L. Smith and W. R. Block, "RTUs slave for supervisory systems," *IEEE Computer Applications in Power*, Vol. 6, No. 1, pp. 27-32, Jan. 1993.

[29] M. Yehia, R. Jabr, I. El-Bitar and R. Waked, "A PC Based State Estimator Interfaced with a Remote Terminal Unit Placement Algorithm," *IEEE Trans. Power System*, Vol. 16, pp. 210-215, May 2001.

[30] N. D. R. Sarma, V. V. Raju and K. S. P. Rao, "Design of Telemetering configuration for Energy Management Systems," *IEEE Trans. Power Systems*, Vol. 9, pp. 381-387, Feb. 1994.

[31] Reza Ebrahimian and Ross Baldick, "State Etimator Condition Number Analysis," *IEEE Trans. On power systems*, Vol. 16, No. 2, May 2001, pp 273-279

[32] D.M. Falcao, F. F. Wu, and L. Murphy, "Parallel and Distributed State Estimation," *IEEE Trans. On Power Systems*, Vol. 10, No.2, pp. 724-730, May 1995.

[33] K A-Rahman, L. Mili, A. Phadke, J.D.L. Ree and Y. Liu, "Internet based wide area information sharing and its roles in power system state estimation," *IEEE PES SM2001*, July, 2001.

[34] Chun-Lien Su and Chan-Nan Lu, "Interconnected Network State Estimation Using randomly Delayed Measurements," *IEEE Trans. On Power Systems*, Vol. 16, No. 4 November 2001, pp. 870-878

[35] I. O. Habiballah, M. R. Irving, "Multipartitioning of power system state estimation networks using simulated annealing," *Electric Power Systems Research* 34 (1995) 117-120.

[36] V C Ramesh, "On distributed computing for online power system applications," *Electrical Power & Energy Systems*, Vol. 18, No. 8, pp 527-533, 1996

[37] T. V. Cutsem and M. Ribbens-Pavella, "Critical Survey of Hierarchical Methods for State Estimation of Electrical Power Systems," *IEEE Trans. On Power Apparatus and Systems*, Vol. 102, No.10, pp. 3415-3424, October 1983.

[38] R. Ebrahimian and R. Baldick, "State estimation distributed processing for power systems," *IEEE Transactions on Power Systems*, Vol.15, No.4, pp.1240-1246, November 2000.

[39] Clements K.A., Denison O.J. and Ringlee R.J., "A multi-area approach to state estimation in power system networks," *IEEE PES Summer Meeting*, Paper C72 465-3, San Fransisco, Ca. On Power Industry Computer Appl., 1973.

[40] Kobayshi H., Narita S. and Hamman M.S.A.A., "Model coordination method applied to power system control and estimation problems," *Proc. Of the IFAC/IFIP 4th Int. Conf. On Digital Computer Appl. To Process Control*, 1974, pp. 114-128

[41] Van Cutsem Th., Horward J. L., and Ribbens-Pavella M. "A Two-level Static State Estimator for Electric Power Systems," *IEEE transaction on PAS*, Vol. PAS-100, Aug. 1981, pp 3722-3732.

[42] Wallach Y. and Handschin E., "An Efficient Parallel Processing Method for Power System State Estimation," *IEEE Trans. On PAS*, Vol, PAS-100, Nov. 1981, pp. 4402-4406.

[43] Garng M. Huang and S.-C. Hsieh, "Fast Textured Algorithms for Optimal Power Delivery Problems in Deregulated Environments," *IEEE Trans. On Power Systems,* Vol. 13, No. 2, pp. 493-500, May 1998.

[44] J.Zaborszky, Garng M. Huang, and K. W. Lu, "A textured model for computationally efficient reactive power control and management," *IEEE Trans. On Power Apparatus and Systems,* Vol. PAS-104, No. 7, pp. 1718-1727, July 1985.

[45] Garng M. Huang, and S.-C. Hsieh, "A parallel HAD-textured algorithm for constrained economic dispatch control problems," *IEEE Trans. On Power Systems,* Vol. 10, No. 3, pp. 1553-1558, August 1995.

[46] G. Huang and Shih-Chieh Hsieh, "Exact convergence of a parallel textured algorithm for constrained economic dispatch control problems," *Systems and control theory for power systems*, J. Chow, R. Thomas and P. Kokotovic, Ed. New York: Springer-Verlag, 1995, pp. 173-204.

[47] Garng M. Huang and W. Hsieh, "A parallel textured algorithm for optimal routing in data networks," *Proc. IEEE Global COM'91*, Phoenix, Arizona, Dec. 1991

[48] G. M. Huang, S. Zhu and W. Hsieh, "Parallel implementation issues of the textured algorithm for optimal routing in data networks," *Proc. Int'l Parallel Processing Symp*., Irvine, California, Apr. 1993

[49] E. Vaahedi, A. Y. Chang, S. Mokhtari, N. Muller and G. Irisarri, "A future application environment for BC Hydro's EMS," *IEEE Transactions on Power Systems*, Vol. 16 No. 1, pp. 9-14, February 2001

[50] B. Qiu,Y. Liu, Y. S. Ong, H. B. Gooi and S. Chen, "Managing Metadata over the WWW using eXtensible Markup Language (XML)," *IEEE Power Engineering Society Winter Meeting*, 2002. IEEE , Vol. 1, pp. 678 -683, January 2002

[51] X. Li, M. Gao, J. Liu, Z. Ding and X. Duan, "A software architecture for integrative utility management system," *IEEE Power Engineering Society Winter Meeting*, Vol.2, pp.476-480, 2001.

[52] D. T. Askounis and E. Kalfaoglou, "The Greek EMS-SCADA: from the contractor to the user," *IEEE Transactions on Power Systems*, Vol. 15, No. 4, pp. 1423-1427, Nov. 2000

[53] A. de Vos, S. E. Widergren and J. Zhu, "XML for CIM model exchange," *Power Industry Computer Applications*, pp.31-37, 2001.

[54] Abdel-Rahman Khatib, Xuzhu Dong, Bin Qiu, Yilu Liu, "Thoughts on Future Internet Based Power System Information Network Architecture," *IEEE Power Engineering Society Summer Meeting*, 2000, pp. 155-160

[55] Q. Zhao, H. P. In, X. Wu and G. M. Huang, "Transforming legacy energy management system (EMS) modules into reusable components: a case study," *Computer Software and Applications Conference*, pp.105-110, 2000.

[56] Jay P. Britton and Arnold N. deVos, "CIM-Based Standards and CIM Evolution," *IEEE Transactions on Power Systems*, Vol. 20, No. 2, pp. 758-764, May 2005.

[57] Stipe Fustar, "Towards Standard CIM-Based Services," *CIM Usage Presentations for CAISO*, http://www.cimuser.org/UserGroup, accessed on April 2, 2007

[58] Stipe Fustar, "Impact of Advanced Integration Technologies on the Next Generation of Energy Systems," *IEEE PES General Meeting 2006*, June 2006

[59] H. Walter Johnson, "SOA and Security," *InfoWorld SOA Executive Forum*, November 2005.

[60] Qizhi Chen, Hamada Ghenniwa and Weiming Shen, "Web-Services Infrastructure for Information Integration in Power Systems," *IEEE PES General Meeting 2006*, June 2006.

[61] InterPSS, "An Internet Technology-based Open-source Power System Simulation System," *http://www.interpss.org*, accessed on April 2, 2007

[62] Boris Lublinsky, "Defining SOA as an architectural style," *http://www.ibm.com /developerworks/architecture/library/ar-soastyle*, accessed on April 2, 2007.

[63] K. A. Clements, G. R. Krumpholz and P. W. Davis, "State Estimation Measurement System Reliability Evaluation," *IEEE Trans. PAS,* Vol. 101, No. 4, pp. 997-1004, Apr. 1982.

[64] J. F. Marsh, "Structure of measurement Jacobin matrices for power system," *IEE Proceedings Vol. 136*, Part C No. 6, November 1989, pp 407-413

[65] I.W. Slutsker and J.M.Scudder, "Network observability analysis through measurement Jacobian matrix reduction," *IEEE Transactions on Power Systems*, Vol.2, No.2, pp.331-337, May 1987

[66] J. Sametinger, *Software Engineering with Reusable Components*. New York, NY: Springer-Verlag, 1997.

[67] Clemens Szyperski, *Component software: beyond object-oriented programming*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1998

[68] Xiaofeng He, C.H.Q. Ding, Hongyuan Zha and H.D. Simon, "Automatic topic identification using webpage clustering," *IEEE International Conference on Data Mining*, pp.195-202, 2001.

[69] D. Tamiar, Y. Jiang, J.W. Rahaya and L. Bishay, "Structured Web pages management for efficient data retrieval," *Proceedings of the First International Conference on Web Information Systems Engineering*, pp.97 –104, 2000.

[70] Cheng, B.H.C. and Jun-Jang Jeng, "Reusing analogous components," *IEEE Transactions on Knowledge and Data Engineering*,Vol. 9, pp.341-349, 1997.

[71] M.Klein, "XML, RDF, and relatives," *IEEE Intelligent Systems*, Vol.16, pp. 26-28, 2001.

[72] A. Gomez-Perez and O.Corcho, "Ontology languages for the semantic web," *IEEE Intelligent Systems*, Vol. 17, pp.54-60, 2002.

[73] Tim Berners-Lee, "Semantic Web Road map," *http://www.w3.org/DesignIssues/Semantic.html*, accessed on April 2, 2007

[74] Tom Gruber, "What is an Ontology?", *http://www-ksl.stanford.edu/kst/what-is-an-ontology.html*, accessed on April 2, 2007

[75] IEC TC57 Working Group 16, "Framework for Deregulated Electricity Market Communications," *http://cimuser.org/WG16/WG16_info.htm*, accessed on April 2, 2007

[76] A.Simoes-Costa and V.H.Quintana, "An orthogonal row processing algorithm for power system sequential state estimation," *IEEE Trans. Power Apparatus and Syste*m, Vol.100, pp.3791-3800, Aug.1981.

[77] A. Monticelli, C. A. F. Murari, and F. F. Wu, "A hybrid state estimator: Solving normal equations by orthogonal transformations," *IEEE Trans. Power Sysem*, vol. 105, pp. 3460–3468, Dec. 1985.

[78] N. Vempati, I. W. Slutsker, and W. F. Tinney, "Orthogonal sparse vector methods," *IEEE Trans. Power Syst.*, vol. 7, pp.926–932, May 1992.

[79] K. A. Clements, P.W. Davis, and K. D. Frey, "Treatment of inequality constraints in power-system state estimation," *IEEE Trans. Power System*, vol. 10, pp. 567–573, May 1995.

[80] N. Vempati, I.W. Slutsker, andW. F. Tinney, "Enhancements to givens rotations for power system state estimation," *IEEE Trans. Power System*, vol. 6, pp. 842–849, May 1991.

[81] J. W. Gu, G. R. Krumpholz, K. A. Clements, and P. W. Davis, "The solution of ill-conditioned power system state estimation problems via the method of Peters and Wilkinson," *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-102, no. 10, pp. 3473–3480, Oct. 1983.

APPENDIX A

MATHEMATIC FORMULATION OF SE

SE problem [1, 2, 3] is based on the following nonlinear model:

$$\hat{z} = \begin{bmatrix} z_1 \\ \cdots \\ \cdots \\ z_m \end{bmatrix} = \begin{bmatrix} h_1(x_1, x_2, ..., x_n) \\ \cdots \\ \cdots \\ h_m(x_1, x_2, ..., x_n) \end{bmatrix} + \begin{bmatrix} e_1 \\ \cdots \\ \cdots \\ e_m \end{bmatrix} = \hat{h}(\hat{x}) + \hat{e}$$

Where

$\hat{z}$ is the measurements typically consisting of bus voltage magnitude measurements and pairs of active and reactive power injection and power flow measurements;

$\hat{e}$ is the measurement noise vector;

$\hat{x}$ is the state vector composed of the magnitudes and phase angles of voltages on buses throughout the entire network;

$\hat{h}$ stands for the nonlinear measurement functions relating $\hat{z}$ to $\hat{x}$ based on the given conditions such as the network and measurement configurations.

Though there are numerous algorithms to deal with the SE problem [76-81], most state estimation programs in commercial software packages for electric power industry are formulated as an over-determined system of nonlinear equations and solved as weighted least squares (WLS) problems. WLS based SE is formulated mathematically as an optimization problem with a quadratic objective function

$$\text{Minimize } J\left(\hat{x}\right) = \frac{1}{2}\sum_{i=1}^{m} w_i r_i^2 = \frac{1}{2}\sum_{i=1}^{m} w_i \left(z_i - h_i\left(\hat{x}\right)\right)^2 = \frac{1}{2}\hat{r}^T R^{-1}\hat{r}$$

Where

z_i is the measured value for telemetric measurement i;

w_i is the weight for telemetric measurement i;

$\hat{r}\left(\hat{x}\right) = \left[z_1 - h_1\left(\hat{x}\right), \quad \ldots \quad \ldots \quad z_m - h_m\left(\hat{x}\right)\right]^T = \hat{z} - \hat{h}\left(\hat{x}\right)$ is the residual vector,

R is the diagonal weighting matrix.

Accordingly, the first-order optimal condition is:

$$\frac{\partial J\left(\hat{x}\right)}{\partial \hat{x}} = -H^T R^{-1} \hat{r} = 0$$

Where H is the Jacobian matrix $\partial \hat{h} / \partial \hat{x}$.

Note that $\frac{\partial^2 J(\hat{x})}{\partial \hat{x}^2} = \sum_{i=1}^{m} R_i^{-1} \frac{\partial h_i}{\partial \hat{x}} \left(\frac{\partial h_i}{\partial \hat{x}} \right)^T - \sum_{i=1}^{m} R_i^{-1} r_i \frac{\partial^2 h_i}{\partial \hat{x}^2}$ consists of first-order term $\sum_{i=1}^{m} R_i^{-1} \frac{\partial h_i}{\partial \hat{x}} \left(\frac{\partial h_i}{\partial \hat{x}} \right)^T = H^T R^{-1} H$ and second-order term $\sum_{i=1}^{m} R_i^{-1} r_i \frac{\partial^2 h_i}{\partial \hat{x}^2}$ [3].

Instead of using Newton Raphson method where the second-order term is retained, Gauss Newton method is applied to find the root of the above equation based on a nonlinear iteration method procedure where the second-order term is ignored [3]:

$$\begin{cases} \hat{x}^{j+1} = \hat{x}^{j} + \Delta \hat{x}^{j} \\ G\left(\hat{x}^{j}\right) \Delta \hat{x}^{j} = H^T R^{-1} \hat{r}\left(\hat{x}^{j}\right) \end{cases}$$

Where

$G = H^T R^{-1} H$ is the gain matrix.

When enough measurements are well distributed throughout the network so that the gain matrix is nonsingular and the above iterative computation can be completed, the network is said to be observable with the given measurement configuration [3,4, 14].

To run a state estimator successfully, an appropriate measurement configuration to make the power network observable is an essential precondition. Accordingly, topological and numerical approaches to examine the network observability have been widely addressed [3,4, 14].

Regardless of the estimation algorithms, the configuration of measurements is one of the decisive factors for a successful state estimator [22].

APPENDIX B

SOME PROOFS

B.1 Descendant of a tree

The concept of descendant of a tree is introduced for the proofs in the appendix.

Definition A.1: Tree T_2, denoted as T_2=<T_1+$br_{c,1}$–$br_{t,1}$>, is termed as a *descendant* of a tree T_1 iff T_2 is derived from T_1 via substituting $br_{t,1}$ with $br_{c,1}$, where $br_{c,1}$ is a co-tree branch of T_1, i.e. $br_{c,1} \notin T_1$, and $br_{t,1}$ is a tree branch in the basic loop $L(T_1, br_{c,1})$.

Definition A.2: T_n is called T_1's *generalized descendant* iff T_n can be derived through a sequence of substitutions from T_1. Note that a descendant of T_1 is also a special generalized descendant of T_1.

Example 1: T_1, T_2 and T_3 are shown in Fig.A.1, A.2 and A.3, which are different spanning trees of the network in Fig.2.1. Obviously T_2=<T_1+b_2b_4–b_3b_4> and T_3=<T_2+b_5b_3–b_5b_7>. T_2 is a descendant of T_1, and T_3 is a descendant of T_2. Accordingly, T_3 is a generalized descendant of T_1.

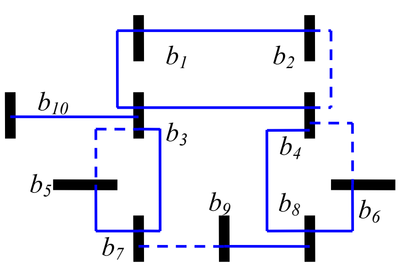
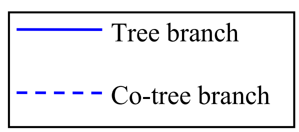


Fig.A.1 Spanning tree T_1



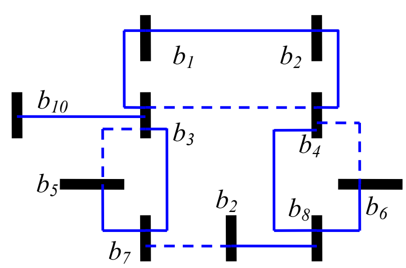


Fig.A.2 Spanning tree T_2

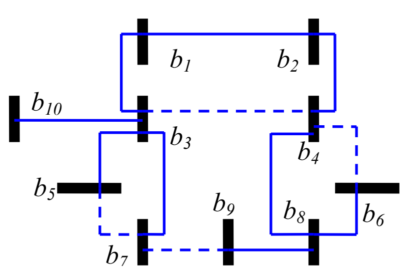


Fig.A.3 Spanning tree T_3

For T_2=<T_1+$br_{c,1}$–$br_{t,1}$>, the following observations holds:

Observation A.1: According to the definition of T_2, all co-tree branches of T_1 except $br_{c,1}$ are also co-tree branches of T_2.

Observation A.2: Suppose br_c is a co-tree branch of both T_1 and T_2, then the basic loops $L(T_1,br_c)$ and $L(T_2,br_c)$ have common tree branches as long as these tree branches are not in $L(T_1,br_{c,1})$, where the concept of basic loop and its notation $L(T, br_{cotree,T})$ are discussed in Definition 4 in Section 2.2.B. This observation can be proven by enumerating all the possible relationships between the two basic loops $L(T_1,br_c)$ and $L(T_1,br_{c,1})$, which is omitted here.

For example, for T_2=<T_1+b_2b_4–b_3b_4> in Example 1 where $br_{c,1}$ is b_2b_4 and $br_{t,1}$ is b_3b_4, the follows hold:

1) All co-tree branches of T_1 except b_2b_4, i.e. b_3b_5, b_7b_9 and b_4b_6, are also co-tree branches of T_2.

2) For a co-tree branch of both T_1 and T_2, say b_7b_9, basic loops $L(T_1, b_7b_9)$ and $L(T_2,b_7b_9)$ have common tree branches b_3b_7, b_9b_8, and b_8b_4 since these tree branches are not in $L(T_1, b_2b_4)=\{b_1b_2, b_2b_4, b_4b_3, b_3b_1\}$.

Suppose T_n is a generalized descendant of T_1, say T_2=<T_1+$br_{c,1}$–$br_{t,1}$>, T_3=<T_2+$br_{c,2}$–$br_{t,2}$ >, …, T_n=<$T_{(n-1)}$+$br_{c,(n-1)}$–$br_{t,(n-1)}$> where n>1, then the following observations holds:

Observation A.3: All co-tree branches of T_1 except $br_{c,1}$, $br_{c,2}$, … and $br_{c,(n-1)}$ are also co-tree branches of T_n.

Observation A.4: Suppose br_c is a co-tree branch of both T_1 and T_n, then the basic loops $L(T_1,br_c)$ and $L(T_n,br_c)$ have common tree branches as long as these tree branches are not in $L(T_1, br_{c,1}) \cup L(T_2, br_{c,2}) \cup \ldots \cup L(T_{(n-1)}, br_{c,(n-1)})$.

Based on Observation A.1 and A.2, Observation A.3 and A.4 can be proven via mathematical induction, which is omitted here.

For example, since T_2=<T_1+b_2b_4–b_3b_4> and T_3=<T_2+b_5b_3–b_5b_7> in Example 1, the followings hold:

1) All co-tree branches of T_1 except b_2b_4 and b_5b_3, i.e. b_7b_9 and b_4b_6, are also co-tree

branches of T_2.

2) For a co-tree branch of both T_1 and T_3, say b_7b_9, basic loops $L(T_1, b_7b_9)$ and $L(T_3, b_7b_9)$ have common tree branches b_9b_8 and b_8b_4 since these tree branches are not in either $L(T_1,b_2b_4)=\{b_1b_2, b_2b_4, b_4b_3, b_3b_1\}$ or $L(T_2,b_5b_3)=\{b_5b_3, b_3b_7, b_7b_5\}$.

B.2 Proof of Theorem 2

The following lemmas are introduced for the proof of Theorem 2:

Lemma A.1: Suppose measurement m_n is re-labeled from '*unknown*' to '*redundant*' during *n*th iteration of Stage5 in the *COG* construction algorithm, then there exists a valid measurement assignment $A^{(n)}:BR\text{-}T^{(n)}\rightarrow M_e^{(n)}$ satisfying the following four statements S1-S4:

S1) m_n is nonessential, i.e. m_n does not belong to $A^{(n)}:BR\text{-}T^{(n)}\rightarrow M_e^{(n)}$;

S2) $T^{(n)}$ is either equal to T or is a generalized descendant of T;

S3) If $T^{(n)}$ is a generalized descendant of T, say $T^{(n)}$ is derived through sn substitutions from T, i.e. $T_1=T$, $T_2=<T_1+br_{c,1}–br_{t,1}>$, $T_3=<T_2+br_{c,2}–br_{t,2}>$, …, and $T^{(n)}=T_{sn}=<T_{sn-1}+br_{c,sn-1}–br_{t,sn-1}>$, then $L(T_1, br_{c,1})$, $L(T_2, br_{c,2})$, …and $L(T_{sn-1}, br_{c,sn-1})$ are included in the *COG* before m_n is re-labeled to '*redundant*'.

S4) For a measurement being labeled as '*unknown*' at this time, say m_i, m_i's inverse mapped branches in A and $A^{(n)}$ are the same, i.e. $A^{(n)}(br\text{-}m_i)=m_i$ holds if $A(br\text{-}m_i)=m_i$.

Lemma A.1 is proven by mathematical induction as follows:

Clearly Lemma A.1 are true when n=1.

Suppose Lemma A.1 holds for $n=k$, , i.e. as long as measurements m_k is re-labeled from '*unknown*' to '*redundant*' during *k*th iteration of Stage5, there exists a valid measurement assignment $A^{(k)}:BR\text{-}T^{(k)}\rightarrow M_e^{(k)}$ satisfying S1-S4 in the lemma. Then we shall prove that the lemma holds also for $n=k+1$.

Suppose measurement m_{k+1} is re-labeled from '*unknown*' to '*redundant*' during ($k+1$)th iteration of Stage5 in the *COG* construction algorithm, where m_{k+1} is essential measurement in $A:BR\text{-}T\rightarrow M_e$ with $A(br\text{-}m_{k+1})=m_{k+1}$ and $br\text{-}m_{k+1}$ is a tree branch of T. Note that $br\text{-}m_{k+1}$ is not in any loop of current *COG*, otherwise m_{k+1} should have been relabeled as '*redundant*' in the previous iterations.

Then there are two possible cases that lead to such a re-labeling operation on m_{k+1}:

Case1: $br\text{-}m_{k+1}$ is in a basic loop $L(T, br\text{-}L_{k+1})$ where $br\text{-}L_{k+1}$ is a co-tree branch of T. In addition, during kth iteration of Stage5, measurement m_k is re-labeled from '*unknown*' to '*redundant*' and m_k's topologically related branches that contain $br\text{-}L_{k+1}$ are added into *COG* accordingly. Consequently, in $(k+1)$th iteration of Stage5, m_{k+1} is re-labeled as '*redundant*' because Condition 1 in Stage5 is satisfied.

Case2: m_k is re-labeled from '*unknown*' to '*redundant*' in kth iteration and m_k's topologically related branches contain $br\text{-}m_{k+1}$. Accordingly, m_{k+1} is re-labeled as '*redundant*' in $(k+1)^{\text{th}}$ iteration because Condition2 in Stage5 is satisfied.

Note that $br\text{-}m_{k+1}$, a tree branch in $L(T, br\text{-}L_{k+1})$, is also a tree branch in $L(T^{(k)}, br\text{-}L_{k+1})$ because:

$T^{(k)}$ is either equal to or a generalized descendant of T according to S2 for $n=k$. If $T^{(k)}$ is equal to T, then obviously $br\text{-}m_{k+1}$ is a tree branch in $L(T^{(k)}, br\text{-}L_{k+1})$. Otherwise, $T^{(k)}$ is a generalized descendant of T and $br\text{-}m_{k+1}$ is still a tree branch in $L(T^{(k)}, br\text{-}L_{k+1})$ because of the following two reasons:

First, since $br\text{-}L_{k+1}$ is not included in *COG* until m_k is re-labeled as '*redundant*', $br\text{-}L_{k+1}$ is not equal to $br_{c,1}$, $br_{c,2}$, … or $br_{c,sk-1}$ according to S3 for $n=k$. Hence $br\text{-}L_{k+1}$, a co-tree of T, is also a co-tree of $T^{(k)}$ according to Observation A.3 and therefore $L(T^{(k)}, br\text{-}L_{k+1})$ can be defined for $T^{(k)}$.

Second, since $br\text{-}m_{k+1}$ is not included in any loop of current *COG* and therefore is not in $L(T_1, br_{c,1})$, $L(T_2, br_{c,2})$, …or $L(T_{sk-1}, br_{c,sk-1})$. Hence $br\text{-}m_{k+1}$ is also a tree branch in $L(T^{(k)}, br\text{-}L_{k+1})$ according to Observation A.4.

Furthermore, $A^{(k)}(br\text{-}m_{k+1})=m_{k+1}$ because S4 holds for $n=k$ and $A(br\text{-}m_{k+1})=m_{k+1}$ where m_{k+1} is being labeled as '*unknown*'.

Now we check if Lemma A.1 holds for $n=k+1$.

For Case1, construct a measurement assignment $A^{(k+1)}:BR\text{-}T^{(k+1)}\rightarrow M_e^{(k+1)}$ as follows:

$T^{(k+1)}=T_{sk+1}=<T^{(k)}+br\text{-}L_{k+1}-br\text{-}m_{k+1}>=<T_{sk}+br_{c,sk}-br_{t,sk}>$;

$A^{(k+1)}=A^{(k)}$ except that, instead of $A^{(k)}(br\text{-}m_{k+1})=m_{k+1}$, we have $A^{(k+1)}(br\text{-}L_{k+1})=m_k$.

$A^{(k+1)}:BR\text{-}T^{(k+1)}\rightarrow M_e^{(k+1)}$ is valid because of the following two reasons:

First, $br\text{-}m_{k+1}$ is a tree branch in $L(T^{(k)}, br\text{-}L_{k+1})$ according to the previous analysis. Therefore, $T^{(k+1)}$ is a descendant of $T^{(k)}$ and also a valid spanning tree. Second, m_k does not belong to $A^{(k)}(T^{(k)})$ according to S1 for $n=k$. Therefore $A^{(k+1)}$ is valid for $T^{(k+1)}$.

Furthermore, such a valid measurement assignment $A^{(k+1)}{:}BR\text{-}T^{(k+1)}\rightarrow M_e^{(k+1)}$ satisfies S1-S4 as follows:

1) m_{k+1} is nonessential due to the definition of $A^{(k+1)}$;

2) Since $T^{(k+1)}=T_{sk+1}$ is a descendant of $T^{(k)}=T_{sk}$, $T^{(k+1)}$ is a generalized descendant of T considering S2 holds for $n=k$.

3) $L(T_{sk}, br_{c,sk})=L(T^{(k)},br\text{-}L_{k+1})$ are also included in the current *COG* because of the following two reasons:

First, $br\text{-}L_{k+1}$ has been added into *COG* at the end of kth iteration, which occurs before m_{k+1} is re-labeled to '*redundant*'.

Second, according to Observation A.4, tree branches in $L(T^{(k)},br\text{-}L_{k+1})$ are either tree branches in $L(T,br\text{-}L_{k+1})$ or branches in $L(T_1, br_{c,1})$, $L(T_2, br_{c,2})$, …and $L(T_{sk\text{-}1}, br_{c,sk\text{-}1})$, which are both included in current *COG*.

4) Note that m_k and m_{k+1} have been re-labeled as '*redundant*' at this time. Therefore, $A^{(k+1)}=A^{(k)}$ on those measurements that are still labeled as '*unknown*'. Hence S4 holds for $n=k$ and k+1.

For Case2, construct $A^{(k+1)}{:}BR\text{-}T^{(k+1)}\rightarrow M_e^{(k+1)}$ where

$T^{(k+1)}=T^{(k)}$;

$A^{(k+1)}=A^{(k)}$ except that, instead of $A^{(k)}(br\text{-}m_{k+1})=m_{k+1}$, we have $A^{(k+1)}(br\text{-}m_{k+1})=m_k$.

$A^{(k+1)}{:}BR\text{-}T^{(k+1)}\rightarrow M_e^{(k+1)}$ is valid because of the following two reasons:

First, $T^{(k+1)}$ is a valid tree because $T^{(k)}$ is a valid tree.

Second, m_k does not belong to $A^{(k)}(T^{(k)})$ according to S1 for $n=k$. Furthermore, since m_k's topologically related branches contain $br\text{-}m_{k+1}$ in Case2, $br\text{-}m_{k+1}$ can be assigned to m_k. Therefore $A^{(k+1)}$ is valid for $T^{(k+1)}$.

Accordingly, such a valid measurement assignment $A^{(k+1)}{:}BR\text{-}T^{(k+1)}\rightarrow M_e^{(k+1)}$ satisfies S1-S4 as follows:

1) m_{k+1} is nonessential due to the definition of $A^{(k+1)}$;

2) Note that $T^{(k+1)}$ is equal to $T^{(k)}$, S2 and S3 hold for $n=k+1$ since they hold for $n=k$.

3) Note that m_k and m_{k+1} have been re-labeled as '*redundant*' at this time. Therefore, $A^{(k+1)}=A^{(k)}$ on those measurements that are still labeled as '*unknown*'. Hence S4 holds for $n=k$ and $k+1$.

As a conclusion, Lemma A.1 holds for $n=k+1$ in both Case1 and Case2, which completes the proof.

Lemma A.2: For the *COG* associated with $A:BR\text{-}T\rightarrow M_e$ of (G, M) where G is an n-bus network, if there exists a redundant measurement, then at most (n-2) measurements are labeled as '*critical*'.

Proof:

Assume there is a redundant measurement in M.

If M consists of only (n-1) measurements and G is observable, then all these (n-1) measurements are critical because G with any (n-2) measurements is unobservable. Therefore, M consists of at least n measurements, i.e. there are (n-1) essential measurement for $A:BR\text{-}T\rightarrow M_e$, say m_1, m_2, …and m_{n-1}, and at least one nonessential measurement, say m_n.

In Stage1 of the *COG* construction algorithm, m_1, m_2 … and m_{n-1} are labeled as '*unknown*' and m_n is labeled as '*redundant*'. In Stage2, *COG* is initialized as T. In Stage3, m_n's topologically related branches, which contain at least one branch, say $br\text{-}m_n$, will be added into *COG*.

$br\text{-}m_n$ is either a tree branch or a co-tree branch of T. If $br\text{-}m_n$ is a tree branch whose A-mapped measurement is m_k, i.e. $A(br\text{-}m_n)=m_k$ where m_k is an essential measurement, then in Stage5 m_k will be re-labeled from '*unknown*' to '*redundant*' because Condition 2 in Stage 5 is satisfied. Otherwise $br\text{-}m_n$ is a co-tree branch and is added into *COG* in Stage3. Then a basic loop forms and at least one tree branch, say br_t, will be in the loop. Suppose br_t's A-mapped measurement is m_t, i.e. i.e. $A(br_t)=m_t$ where m_t is an essential measurement, then m_t will be relabeled from '*unknown*' to '*redundant*' because Condition 1 in Stage5 is satisfied.

As a conclusion, at least one of the (n-1) essential measurements will be relabeled from '*unknown*' to '*redundant*' in Stage5, i.e. there is at most (n-2) essential measurement still being labeled as '*unknown*' after Stage5. Hence at most (n-2) measurements are re-labeled from '*unknown*' to '*critical*' in Stage6, which complete the proof of Lemma A.2.

Lemma A.3: All measurements labeled as '*critical*' are actually critical measurements.

Proof of Lemma A.3 by contradiction is as follows:

Suppose an essential measurement m_1 is labeled as '*critical*' after the construction of *COG* for $A{:}BR\text{-}T \rightarrow M_e$ in (G, M) with $A(br\text{-}m_1)=m_1$, then $br\text{-}m_1$ is a radial branch of *COG*.

We assume that m_1 is redundant, i.e. the system is observable after removing m_1. Hence there exists a measurement assignment $A_1{:}BR\text{-}T_1 \rightarrow M_{e1}$ where T_1 is a spanning tree and m_1 is a nonessential measurement. Then a contradiction is found via the following argument:

First, since $br\text{-}m_1$ is a radial branch of *COG*, removing $br\text{-}m_1$ splits *COG* into two separate networks COG_1 and COG_1'. Considering T_1 is a spanning tree, there must be at least one branch in T_1, say $br1$, to connect COG_1 and COG_1'. Suppose $A_1(br_1)=m_2$.

Next, if m_2 is labeled as '*redundant*' in *COG*, then br_1 will be included in *COG*. Consequently, either there is a loop in *COG* containing $br\text{-}m_1$ (when $br_1 \neq br\text{-}m_1$) or TB(m_2) include $br\text{-}m_1$ (when $br_1=br\text{-}m_1$). Then m_1 should be labeled as '*redundant*' in *COG*, which contradicts our assumption. Therefore, m_2 is also labeled as '*critical*' in *COG*, i.e. there exist $A(br\text{-}m_2)=m_2$ in $A{:}BR\text{-}T \rightarrow M_e$.

Furthermore, since m_2 is labeled as '*critical*' in *COG*, $br\text{-}m_2$ must be a radial branch of *COG*. Therefore, removing $br\text{-}m_2$ splits *COG* into two separate graphs COG_2 and COG_2'. Considering T_1 is a spanning tree, there must be at least one branch in T_1, say $br2$, to connect COG_2 and COG_2'. Suppose $A_1(br_2)=m_3$.

If m_3 is labeled as '*redundant*' in *COG*, then br_2 will be included in *COG*. Consequently, either there is a loop in *COG* containing $br\text{-}m_2$ (when $br_2 \neq br\text{-}m_2$) or TB(m_3) include $br\text{-}m_2$ (when $br_2=br\text{-}m_2$). Then m_2 should be labeled as '*redundant*' in *COG*, which contradicts the conclusion in the previous reasoning. Therefore, m_3 is also labeled as '*critical*' in *COG*.

Note that m_1, m_2 and m_3 are distinct measurements because:

Measurement m_1 is not equal to either m_2 or m_3 because m_1 is a nonessential measurement for $A_1{:}BR\text{-}T_1 \rightarrow M_{e1}$ while m_2 or m_3 are essential measurement for $A_1{:}BR\text{-}T_1 \rightarrow M_{e1}$. Then $br\text{-}m_1$ and $br\text{-}m_2$ are different branches because their *A*-mapped measurements m_1 and m_2 are different in $A{:}BR\text{-}T \rightarrow M_e$. Hence br_1 and br_2 are different branches because these two branches are to connect the isolated networks after the removal of different radial branches $br\text{-}m_1$ and $br\text{-}m_2$ in *COG*. Therefore m_2 and m_3 are different because their A_1-1-mapped branches br_1 and br_2 are different in $A_1{:}BR\text{-}T_1 \rightarrow M_{e1}$. As a conclusion, m_1, m_2 and m_3 are distinct measurements.

Suppose there are totally *n* buses in the network *G*, we can continue the above argument to find distinct measurements m_4, m_5, m_6 … m_{n-1} that are also labeled as 'critical' in *COG*. In other word, all the (*n*-1) essential measurements in $A{:}BR\text{-}T \rightarrow M_e$ are labeled as '*critical*'.

On the other hand, since m_1 is redundant, at most (*n*-2) measurements are labeled as '*critical*' according to Lemma A.2, which leads to a contradiction and complete the proof of Lemma A.3.

Since a measurement is labeled as either '*redundant*' or '*critical*' after the construction of *COG*, the following corollary immediately follows from Lemma A.3.

Corollary 1: All redundant measurements will be labeled as '*redundant*' after the construction algorithm of *COG*.

Then Theorem 2 follows.

Theorem 2: Measurement's redundant or critical property matches with '*redundant*' or '*critical*' labeling after the construction of *COG*.

Proof:

For a measurement, say m_i, being labeled as '*redundant*' after the construction of *COG*, m_i is redundant because:

m_i is either initialized as '*redundant*' in Stage1 or re-labeled from '*unknown*' to '*redundant*' in Stage5. If m_i is labeled as '*redundant*' in Stage1, m_i is nonessential for

$A{:}BR\text{-}T \rightarrow M_e$ and therefore is redundant according to Observation 1 in subsection 2.2.B. Otherwise m_i is re-labeled as '*redundant*' in Stage5, thus, m_i is nonessential for a certain valid measurement assignment according to Lemma A.1 and therefore is also redundant.

Furthermore, all redundant measurements will be labeled as '*redundant*' according to Corollary 1. As a conclusion, measurement's redundant property matches with '*redundant*' labeling.

Since a measurement is either redundant or critical and is labeled as either '*redundant*' or '*critical*', this concludes that measurement's critical property also matches with '*critical*' labeling, which completes the proof of Theorem 2.

B.3 Proof for some properties of COG

Proof of Property 1:

According to Definition 3, $A^{-1}(M_e)= BR\text{-}T$.

According to Observation 5 in subsection 2.2.B, $A^{-1}(m_i) \in TB(m_i)$ for $m_i \in M_e$.

Then $BR\text{-}COG= A^{-1}(M_c) \cup TB(M_r)$

$$\begin{aligned}
&\supseteq A^{-1}(M_e \cap M_c) \cup TB(M_e \cap M_r)\\
&\supseteq A^{-1}(M_e \cap M_c) \cup A^{-1}(M_e \cap M_r)\\
&= A^{-1}((M_e \cap M_c) \cup (M_e \cap M_r))\\
&= A^{-1}(M_e \cap (M_c \cup M_r))\\
&= A^{-1}(M_e \cap M) \} = A^{-1}(M_e) = BR\text{-}T
\end{aligned}$$

That is to say, *COG* contains all tree branches, which indicate that *COG* contains *T*. Furthermore, as a spanning tree of *G*, *T* covers all buses of *G*. Therefore, *T* is also a spanning tree of *COG* since *COG* is contained in *G*; and this completes the proof.

Proof of Property2:

For Case1, *br* is removable according to Observation 8 in subsection 2.2.D.

For Case2, since *br* is not part of *COG*, *br* is a co-tree branch in $A{:}BR\text{-}T \rightarrow M_e$. Therefore, the branch outage on *br* has no impact on $A{:}BR\text{-}T \rightarrow M_e$, which proves the system is still observable.

For Case3, if *br* is a co-tree branch in $A{:}BR\text{-}T \rightarrow M_e$, then *br* is removable as discussed in the above Case2; else if *br* is a tree branch br_t in $A:BR\text{-}T\rightarrow M_e$ and is contained in a basic loop $L(T, br_{c\text{-}t})$ in *COG*, then we suppose $br_{c\text{-}t}$ is added into *COG* in Stage5 in *k*th iteration as one of measurement *m*'s topologically related branches, where *m* is newly re-labeled as '*redundant*' in *k*th iteration. Based on the analysis in Case1 in the proof of Lemma A.1, we can find a valid measurement assignment $A^{(k+1)}:BR\text{-}T^{(k+1)}\rightarrow M_e^{(k+1)}$ for (*G*, *M*), where the tree $T^{(k+1)} = <T^{(k)}+br_{c\text{-}t}-br_t>$ does not contain br_t and $A(br_t)$ is nonessential. Consequently, when br_t is in outage, the network still maintains its observability because we still have a valid measurement assignment $A^{(k+1)}:BR\text{-}T^{(k+1)}\rightarrow M_e^{(k+1)}$.

As a conclusion, *br* is removable in the above three cases, which completes the proof.

Proof of Property 4: According to Property 3, *br* in *G* is not removable only if *br* is non-radial in *G* while radial in *COG*.

Now we need to prove that *br* is not removable if *br* is non-radial in G while radial in COG. Suppose *br* is still removable, which means (G, M) is observable as a whole after the removal of *br*, i.e. there exist a valid measurement mapping $A:BR\text{-}T\rightarrow M_e$ where tree T covers the two end buses of *br*. Then a branch loop forms after adding *br* into T.

On the other hand, since all injection measurements are redundant, COG is unique and therefore this tree T is a sub-graph of COG, thus both *br* and T belongs to COG, which contradicts with the given condition that *br* is radial in COG. As a conclusion, *br* is not removable if *br* is non-radial in G while radial in COG, which completes the proof.

B.4 Proof of Theorem 4

We will prove Theorem 4.

Theorem 4: (*G*, *M*) maintains its observability against any single contingency if and only if the associated *COG* satisfies both of the following conditions:

Condition 1: Radial branches in *COG* are radial in *G*;

Condition 2: Every measurement whose A^{-1}-mapped branch is radial in *COG* is redundant.

Proof:

First we prove the two conditions are sufficient as follows:

Suppose Condition 1 and 2 are both satisfied, then every measurement is redundant because:

A measurement is either an essential or a nonessential measurement. For an essential measurement, say m_e, m_e's A^{-1}-mapped branch $br\text{-}m_e$ is either non-radial or radial in G. If $br\text{-}m_e$ is non-radial in G, then $br\text{-}m_e$ is non-radial in *COG* according to Condition 1. In other words, $br\text{-}m_e$ is in a loop of *COG* and therefore m_e is redundant according to the construction algorithm of *COG*. Otherwise, $br\text{-}m_e$ is radial in G. Then m_e is still redundant according to Condition 2. As a conclusion, every essential measurement is redundant. Furthermore, every nonessential measurement is redundant according to Observation1. Therefore, all measurements are redundant suppose Condition 1 and 2 are satisfied.

In addition, if Condition 1 is satisfied, i.e. a radial branch in *COG* must be radial in G, then every branch in G is removable due to Property 3 of *COG*.

As a conclusion, the two conditions are sufficient.

Next we prove the two conditions are also necessary.

Condition 2 is necessary obviously. Otherwise there exists a critical measurements and the network is unobservable after the removal of the critical measurement.

Condiont1 is proved to be also necessary as follows:

Suppose (G,M) maintains its observability against any single contingency, then there should be no critical measurement. In addition, we suppose *br* is non-radial in G but radial in *COG*. Then *COG* is split into two isolated graphs COG_1 and COG_2 after removing *br*. Since *br* is non-radial in G, (G,M) is observable as a whole after removing *br*, i.e. there exists $A':BR\text{-}T'\rightarrow M_e'$ where T' is a spanning tree and there exists a branch in T', say br', to connect COG_1 and COG_2. Suppose $A'(br')=m\text{-}br'$. Since there is no critical measurement, $m\text{-}br'$ is redundant, which indicates that br' will be included in *COG* and hence a loop containing *br* is found in *COG*, which leads to a contradiction.

As a conclusion, the two conditions are also necessary, which completes the proof of

Theorem 4.

VITA

Jiansheng Lei received his B.S. and M.S. degrees in Electrical Engineering from Tsinghua University, Beijing, China, in 1998 and 2000, respectively. He joined Texas A&M University in January 2001 as a Ph.D. student under the guidance of Dr. Garng M. Huang. He worked in ABB Network Management as summer interns in 2002 and 2003, respectively. Then he joined Electric Reliability Council of Texas (ERCOT) in June 2004. His current research interests include power system analysis, software engineering, and their applications in electric power market.

Mr. Lei may be reached at ERCOT, 2705 West Lake Drive Taylor, Texas 76574. His email address is jiansheng@gmail.com.