2005

Slow coherency grouping based islanding using minimal cutsets and generator coherency index tracing using the continuation method

Xiaoming Wang
Iowa State University

NOTE TO USERS

Slow Coherency grouping based islanding using Minimal Cutsets and generator coherency index tracing using the Continuation Method

by

Xiaoming Wang

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Electrical Engineering

Program of Study Committee:
Vijay Vittal, Major Professor
James D. McCalley
Venkataramana Ajjarapu
Degang J. Chen
David A. Hennessy

Iowa State University

Ames, Iowa

2005

Graduate College
Iowa State University

This is to certify that the doctoral dissertation of

Xiaoming Wang

has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

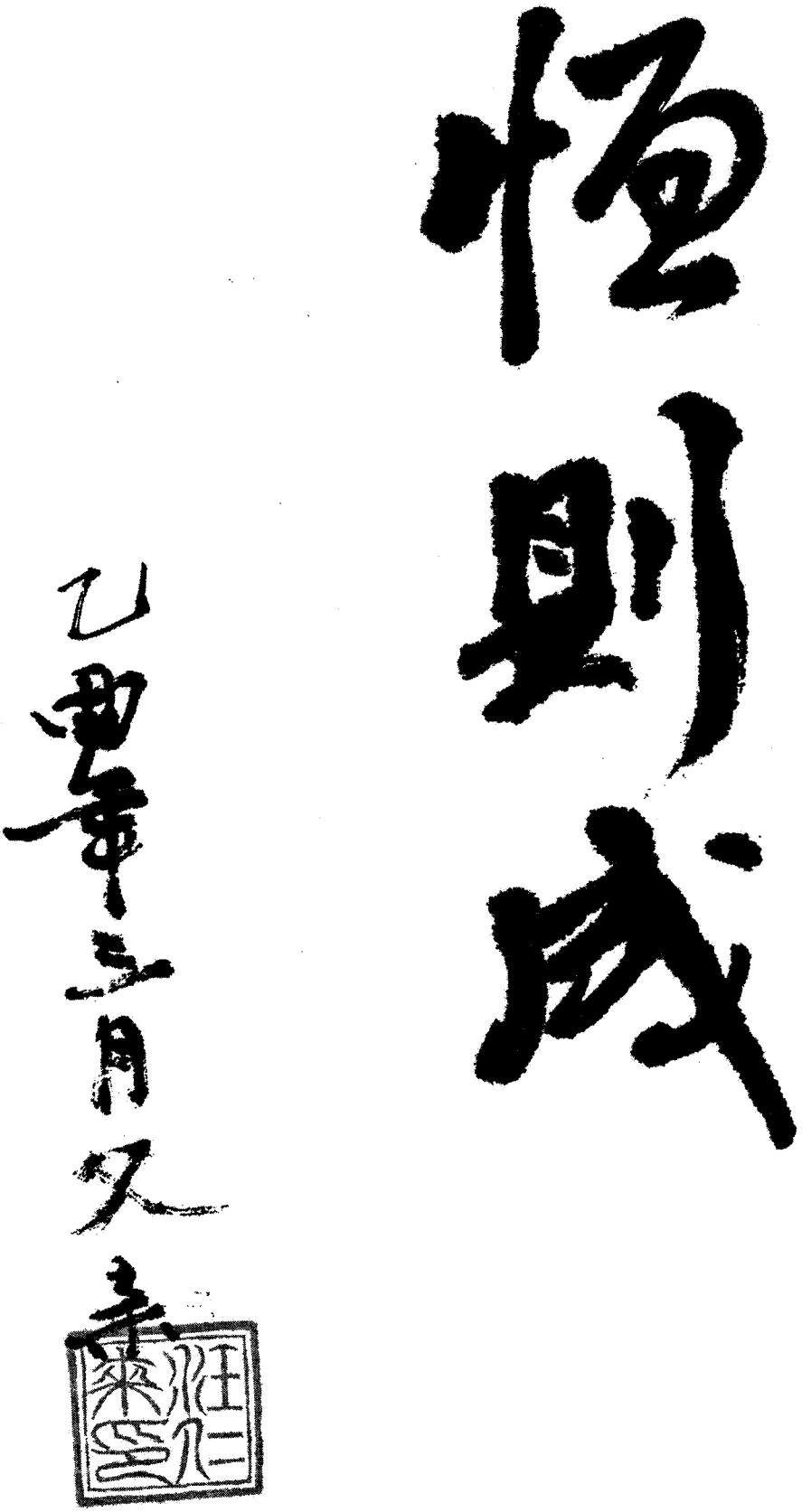
Major Professor

Signature was redacted for privacy.

For the Major Program

謹以此論文
獻給我的妻子，我的父母，還有我的弟弟。

This dissertation is dedicated to my beloved wife, Yichun Cho, my parents, Yueya Tang and Renlai Wang, and my little brother, Xiaofei Wang.
They are always there to encourage and support me.


恒則成
乙巳春三月又幸

學海無涯苦作舟

書山有路勤為徑

乙酉年五月 張福興

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am deeply indebted to my major advisor, Dr. Vijay Vittal, for his constant support. Without his guidance, this work would not be even possible. I would also like to thank the members of my committee Dr. James D. McCalley, Dr. Venkataramana Ajjarapu, Dr. Degang J. Chen, and Dr. David A. Hennessy (Economics). Their advice and patience is highly appreciated.

I also want to thank Dr. Gurpu M. Prabhu (Computer Science) for his invaluable advice on Graph Theory. A special thank also goes to Dr. Elgin H. Johnston (Mathematics) who greatly enriched my knowledge with his exceptional insights into mathematics. I also would like to acknowledge Vice President Joseph Bright and David Takach at Nexant who provided their valuable suggestion on my thesis.

I would like to acknowledge my comrades in the power group at Iowa State University who provided me with their outstanding advice throughout the years. Discussions with Dr. Kun Zhu paved the way for the first part of my research. Additionally, I would be amiss without particularly acknowledging the support of Xiaoyu Wen who provided invaluable suggestion at the beginning of the second part of my research. I would also like to thank one of my best friends, Dr. Weiguo Yang, for his constant encouragement and support.

I am also indebted to the fellow graduate students before me: Dr. Haibo You, Dr. Wenzheng Qiu, Dr. Qiming Chen, Dr. Jiang Huang, and Dr. Zhong Zhang.

Finally, I would like to thank Wei Shao, Shu Liu, Qian Liu, Yong Jiang, Ramanathan Badri, Ana Margarida Quelhas, Bo Yang, Weiqing Jiang and many others. Their great friendship accompanied me throughout my school years and made my life in Ames so meaningful. These memories will never be forgotten.

ABSTRACT

Power systems are under increasing stress as deregulation introduces several new economic objectives for operation. Since power systems are being operated close to their limits, weak connections, unexpected events, hidden failures in protection system, human errors, and a host of other factors may cause a system to lose stability and even lead to catastrophic failure. Therefore, the need for a systematic study and design of a comprehensive system control strategy is gaining more attention. Among these control methods, controlled system islanding is deemed as the final resort to save the system from a blackout.

In the literature, many approaches have been proposed to undertake this task. However, some of these approaches only take static power flow into consideration; others require a great deal of computational effort. It has been observed that following large disturbances, groups of generators tend to swing together. Attention has thus been drawn to the stability of inter-area oscillations between groups of machines. The slow-coherency based generator grouping, which has been widely studied in the literature, provides a potential method for capturing the movement of generators between groups under disturbance. The issue becomes on how to take advantage of slow coherency generator grouping and island the system by finding the set of lines to be tripped. Furthermore, through various simulations and analysis, it has been found that generator grouping indeed changes with respect to large changes in system load conditions.

In this dissertation, a comprehensive approach has been proposed to conduct slow coherency based controlled power system islanding using the minimal cutset technique from graph theory with the transition from calculating real power imbalance within the island to calculating the net flow through the cutset. Furthermore, a novel approach has been

developed to trace the loci of the coherency indices of the slow modes in the system with respect to variation in system conditions to obtain the updated coherency information between generators using continuation method. Finally, the approach has been applied to a 10 generator 39 bus New England system, and a 29 generator 179 bus model of the WECC system.

1 INTRODUCTION

With the advent of deregulation and restructuring, power systems have come under increasing stress as deregulation has introduced several new economic objectives for operation. Since systems are being operated close to their limits, weak connections, unexpected events, hidden failures in protection system, human errors, and a host of other factors may cause a system to lose stability and even lead to catastrophic failure. Therefore, the need for a systematic study and design of a comprehensive system control strategy is gaining more and more attention.

The work described in this dissertation is the extension of a portion of our work done under a EPRI/U.S. Department of Defense (DoD) project to design the Strategic Power Infrastructure Defense (SPID) system, supported by the project "Detection, Prevention and Mitigation of Cascading Events" granted by the Power Systems Engineering Research Center (PSERC). This work is directed at enhancing reliability of interconnected power systems and preventing cascading outages: "When a power system is subjected to large disturbances, and vulnerability analysis indicates that the system is approaching a potential catastrophic failure, control actions need to be taken to steer the system away from severe consequences, and to limit the extent of the disturbance"[1]. In this project three steps are proposed to address this problem:

1. Detect Major Disturbances and Protective Relay Operations Leading to Cascading Events.
2. Utilize Wide Area Measurement-Based Remedial action.
3. Initialize Controlled System Islanding with selective under frequency load shedding.

As a part of this project, Controlled System Islanding acts as the final resort to save the system from a blackout.

1.1 Power System Reliability

The North American Electric Reliability Council (NERC) Planning Standards define two components of reliability, a) adequacy of supply and b) transmission security:

Adequacy is the ability of electric systems to supply the aggregate electrical demand and energy requirements of customers at all times, taking into account scheduled and reasonably-expected unscheduled outage of system elements.

Security is the ability of electric systems to withstand sudden disturbances such as electrical short circuits or unanticipated loss of system elements.

Corrective control strategies contribute to solve the security problem in many such aspects, as circuit overload, voltage problems, and transient problems.

1.2 Power System Operating States

The bulk power grid is the largest and most complex interconnected network ever devised by man, which makes its control an extremely difficult task. Generally, the ability of a power system to survive a given disturbance depends on its operating condition at the time of occurrence, and any adaptive control scheme needs to be designed in such way that it will only be activated when the system is in an appropriate operating condition.

In order to facilitate investigation of power system security and design of appropriate control strategies, power systems can be conceptually classified into five operational states: Normal, Alert, Emergency, In Extremis, and Restorative [2]. Various preventive and corrective control strategies for coping with power systems in the different operational states have been studied. Figure 1-1 illustrates these operating states and the transitions which can take place between states. Figure 1-1 also shows the relative corrective control strategies according to different system operating conditions. [3]

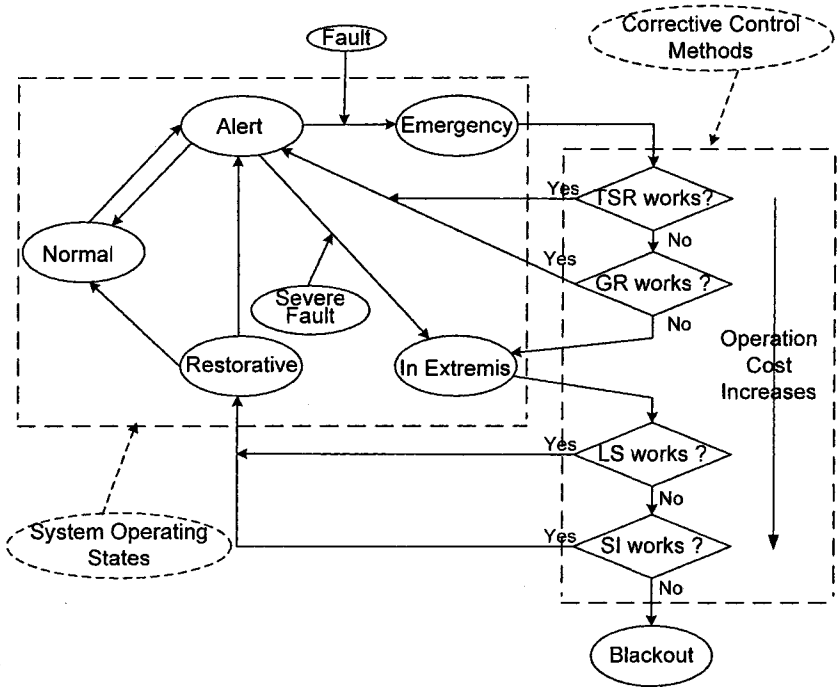


Figure 1-1 Power system operating states and relative corrective control strategies

Normal State

In this state, all the system variables are in the normal range and no system component is being overloaded. The system operates in a secure manner and is able to withstand a contingency without violating any constraints.

Alert State

The system enters the alert state when the system condition is degraded. In this state, all the system variables are still within the acceptable range and no constraints have been violated. However, the system components may overload when an N-1 contingency occurs and leads the system into an emergency state. The system may also directly transit from the alert state to the in extremis state if the disturbance is severe enough.

Emergency State

When the system is in the alert state, a sufficiently large contingency event may bring the system to the emergency state, where system voltages at many buses go below the normal range and the one or more system components may experience overloading. In this state, the system is in operation and may be restored back to the alert state by initiating corrective control strategies such as transmission system reconfiguration (TSR), generation rescheduling (GR), and load shedding (LS), etc.

In Extremis

The system enters the in extremis state if the relative corrective controls are not applied or are ineffective when the system is in the emergency state. Corrective control strategies in this state include Load shedding (LS) and controlled system islanding (CSI). These controls are intended to prevent total system blackout and preserve as much of the system as possible.

Restorative State

This state depicts a condition where control strategies are being deployed to reconnect all system components and to restore system load. Depending on the system condition, the system may transfer to the alert state, or directly transit back to the normal state.

In summary, when a severe fault occurs in a power system, the system may enter the emergency state or even the in extremis state, where the system may encounter an overloading condition, voltage violations, cascading failures, or even loss of stability requiring that system operators take appropriate corrective control actions. It is well known that transmission system reconfiguration (TSR) (including line switching and bus-bar switching) and controlled system islanding (CSI) are two effective corrective control strategies for various system operational states. When the system is in the emergency state, TSR may change the power flow distribution and voltage profiles and consequently solve

the problems of overloads and voltage violations caused by system faults. However, when the system is being operated close to its limits, TSR may not successfully relieve all the overloads and voltage violations for some severe faults, and consequently the system may lose stability or even suffer catastrophic failure. More aggressive corrective control strategies, such as LS and/or CSI must be used to prevent catastrophic failure.

1.3 Power System Restoration [4]

Today's bulk power system provides a highly-reliable supply of electric power. However, with this increased payoff has come increased risk. There is, for example, the potential possibility of system wide outage, making it necessary to provide preventive, corrective, and restorative actions to reduce the possibility, the extent, and the duration of an outage. During the past few decades, considerable effort has been directed toward studying the topics of preventive and corrective control. There is a critical need for an up-to-date, readily-accessible, and highly-reliable power system restoration plan that can quickly restore the system from an outage condition in an orderly fashion with minimal impact to the society being served.

Restoration plans differ in specificity based on the characteristics of different power systems. However, the restoration process can commonly be divided into three discrete stages: preparation, system restoration, and load restoration.

The goals and objectives of restoration can be mainly grouped into the following three categories:

1) To restore all loads safely
2) To restore smoothly and deliberately
3) To minimize the overall restoration time

With regard to these considerations, there are several issues that should be considered seriously during system restoration, among which a few can be listed as follows:

1) *Reactive power balance*

During the early stages of the restoration, system voltages should be maintained within an acceptable range, usually somewhat lower than the normal level, under a condition caused by a small amount of load pick-up at the receiving end of transmission lines in the early stage, which makes it easy to raise the system voltages due to the Feranti effects. There are several ways to solve this problem: energizing fewer high voltage lines, operating generators at minimum voltage levels, deactivating shunt capacitors/activating shunt reactors, or picking up load with a lagging power factor. In the literature, several problems related to reactive power balance have been described, such as sustained over-voltage and under-voltage, generator under-excitation, and switched-shunt capacitors/reactors.

2) *Load and Generation balance*

It is critically necessary to sustain the system frequency by determining the rate of response of the prime movers, from which the amount of load pick-up can then be determined. A potential change of frequency may occur as a function of: 1) Load-Generation mismatch, 2) the effect of underfrequency relays, and 3) the rate at which a generator can be loaded. The availability of real and reactive generation is based on prime movers' conditions just prior to an outage and their start-up times. By bringing those units online early in the restoration with fast rates of response and proper reactive absorbing capabilities, the process can be shortened by a significant time. During this process, more generating units will generally be available for re-start than for load, so load-generation balance is critical, taking into consideration small islands of load and generation.

Details of problems reported by the NERC include sudden increase in load and unnecessary UFLS.

3) *Load and Generation coordination*

In the early stage of power system restoration, one or more load and generation islands are formed to reduce the impact of an outage and to simplify the problem. The number of islands is limited by availability of resources (including operating teams), black-start capability within each island, and coordination between the various control centers.

Regarding the unit start-up (especially hot start-up), several critical time intervals such as the maximum down time and the minimum down time should be accurately determined. Many issues are in reality related to black start-up capacity, steam-unit start-up coordination, switching operations, overloading during restoration, and dispatch office coordination.

1.4 Problem Statement

On one hand, power system islanding is usually considered such a rare or improbable event that it seems not to merit special consideration. On the other hand, the significant impact of unintentional islanding on power system and electricity customers leads many individuals to have great concern about this situation. On November 9th, 1965, the largest power system blackout in history occurred. The northeast power system broke up 4 seconds after an initial disturbance, and 30 million people were without electricity for as long as 13 hours. On August 14th, 2003, widespread power blackouts occurred in the Northeastern United States and in Southeastern Canada, affecting eight states and two provinces with combined population of approximately 50 million people [5]. In reality, most intentional islanding schemes are based on engineering experience, and lack either theoretical analysis or reality validation. Therefore, it is our intention to develop an adaptive islanding scheme by taking into account not only system dynamic characteristics, but also the topology of the power network. This adaptive islanding approach breaks the system up into smaller islands at slightly reduced capacity, with an added advantage that the system can be restored very quickly.

Requirements and considerations in forming Islands:

1) Frequency deviation considerations: Active power imbalance between generation and load induces a frequency deviation from the nominal value. Low-frequency deviations especially cause many more problems in power systems than

high-frequency deviations. Therefore, in this dissertation, approaches have been proposed to only deal with islands with excess load.

2) Voltage stability considerations in reactive power balance: Under certain circumstances, the system within the island could collapse due to cascading events initiated by voltage instability. Therefore, it is important to consider the reactive power balance within the island.
3) Restoration considerations with regard to black-start capability or remote cranking power capability: All power systems require contingency arrangements to enable a restart in the unexpected event that all or part of the system is out of service. The process of restoring the power system is commonly referred to as "Black Start". It entails "islanded" power subsystems being started individually and then gradually being reconnected in order to restore system integrity.
4) Flexibility: automation as our goal for power system islanding. The islanding approach should be designed in such a way that it can provide the system operator a reasonable islanding solution without a great deal of human interaction. However, it should also be able to acquire and utilize information from human evaluation and prediction to improve performance.

Slow coherency has been widely used in industry power system dynamics studies to reduce system scale while not affecting accuracy. In this research, it has been a practice to group generators with similar behavior, referred to as coherent generators [6]. Slow coherency has very nice features, such as: 1) The coherent groups of generators are almost independent of the size of the disturbance. 2) The coherent groups are independent of the level of detail used in modeling the generating units. The first feature provides a theoretical background with which to design an islanding approach independent of disturbance, which would make it possible to design a controlled islanding scheme prior to the disturbance. The second feature simply states that classical generator model can be used in grouping

analysis, which may save the computation effort dramatically.

However, various studies indicate that generator coherency may change with respect to the change in the system operating point, and therefore grouping information at one particular operating point may not be suitable to use in another operating point. Since we believe that system dynamic characteristics should be considered in system islanding, and slow coherency is one of most widely-used approaches to group generators for processing system islanding in such a way that generators in one group shall be included in one island, further study may be needed to investigate the relationship between the generator coherency and system operating points.

1.5 Dissertation Organization

A brief introduction has been presented in Chapter 1. The remainder of this dissertation is organized as follows:

Chapter 2 details the concept of slow-coherency-based generator grouping and the motivation for introducing minimal cutsets into power system islanding. Starting with a review of relevant literature, slow coherency and some graph theory terminology are introduced in this chapter. A system islanding scheme using these concepts is also presented in this chapter. The proposed approach answers the following questions in detail: where is islanding initiated? How does it work? What are the advantages and disadvantages of the proposed method? Results obtained from the WECC 29-Generator and 179-Bus system are also given in this chapter. The summary for this chapter highlights its important aspects and provides a transition from the first part of this proposal to the second part, Chapter 3.

In order to investigate the relationships between generator coherency and system islanding and loading conditions, Chapter 3 presents a slow-mode tracing technique using the continuation method. Beginning with a brief introduction and motivation of this approach, a complete formulation is presented. The implementation of the approach is also discussed in detail. Results from the New England 10 Generator 39 Bus test system and

WECC 29 Generator 179 Bus test system are also presented.

Conclusions, future work, and contribution are presented in Chapter 4.

2 SLOW COHERENCY GROUPING BASED ISLANDING USING MINIMAL CUTSETS

2.1 Relevant Literature Review

Following large disturbances, groups of generators tend to swing together. Attention has thus been drawn to the stability of inter-area oscillations between groups of machines. These oscillations are lower in frequency than local oscillations between electrically-close machines. As a result, there is a separation in time scale between these two phenomena. Additionally, several comprehensive software packages for computing such low frequencies in large power systems are available with which to analyze the participation of the machines in these oscillations.

In References [7], [8], [9], [10], and [11], a slow-coherency approach based on the two-time-scale model has been successfully applied to the partitioning of a power system network into groups of coherent generators.

In the literature, there are some other approaches for the detection of islanding. In Reference [12], a spectral method for identifying groups of strongly connected sub-networks in a large-scale interconnected power system grid is presented as an alternative to long-standing singular perturbation-based coherency techniques. Reference [13] introduces an algorithm based on the breadth-first-search (BFS) algorithm from graph theory for island detection and isolation. In Reference [14], an interesting method based on the occurrence of singularity in Newton power flow is illustrated. Reference [15] gives an active technique based on the voltage-magnitude variation method of a distributed generation unit for detecting islanding, In [16], the authors present an interesting method for system splitting by using the OBDD technique. In the case of splitting a system into two islands, each load bus belongs either to one island or the other. This relationship can be captured by a Boolean

variable. A software package called 'BuDDY' [17] has been utilized to determine the value of these Boolean variables in order to cap the generation and load imbalance within limits in the island. However, for better system islanding, the dynamic characteristics of system, in particular dynamics of generators and loads, should be taken into consideration. The slow-coherency approach of generator grouping, which has been widely studied in the literature, provides a potential for capturing the movement of generators between groups under disturbance. It has the ability to capture both system dynamics and network topology. Therefore, in this approach, we use slow coherency as our grouping technique.

Based on slow coherency, the generators in the system may be divided into several groups. For two interconnected generator groups, reference [6] presents an islanding method for constructing a small sub-network using the center bus which is one of the buses in the group boundary. This sub-network is referred to as the interface network. A brute force search is then conducted on the interface network to determine the cutsets where the islands are formed. For each island candidate, the total load and generation are calculated, and the island with minimum load-generation imbalance is picked up as the optimal cutset if no other criteria have been considered. This approach converts the objective of finding the optimal cutset from that of searching the whole network into that of searching the interface network, making the searching space much smaller. However, this approach still involves considerable computational effort, particularly that of the brute force search applied in this approach. Furthermore, it is system-dependent since for some specific system, it may return fairly good results, while it may not for others. In this dissertation, a new slow-coherency-grouping based approach using minimal cutsets is presented to solve this type of problem.

Minimal cutsets have been previously investigated in communication, network topology, and network (particularly, power system) reliability analysis (maximum flow and connectivity) [18], [19], [20], [21]. As shown in this approach, it also has the potential for determining where to actually island the system.

In the remainder of this chapter, an introduction to the basic concept of slow coherency is provided,

2.2 Slow Coherency

In the controlled-islanding self-healing approach, it is critical to determine the optimum set of islands for a given operating condition. An elegant and flexible approach to islanding can result in significant benefit to the post-fault corrective control actions that follow the islanding, including a load-shedding procedure and a load-restoration procedure. Generally, islanding is system-dependent. Reference [22] indicates that the choice of islands is almost disturbance-independent, which makes it easy to implement a fairly general corrective control scheme for a given system.

Slow coherency was originally used in the development of dynamic equivalents for transient-stability studies. Several methods have been used to identify coherent groups of generators [8], [23]. In all these methods, there are two common assumptions:

The coherent groups of generators are almost independent of the size of the disturbance.

The coherent groups are independent of the level of detail used in modeling the generating unit.

The first assumption is based on the observation that the coherency behavior of a generator is not significantly changed as the clearing time of a specific fault is increased. A heuristic argument has also been given in Section 3.1. Although the amount of detail of the generator model can affect the simulated swing curve, it does not radically change the basic network characteristics such as inter-area modes. This forms the basis of the second assumption. In the following section, a brief introduction of slow coherency is given.

2.2.1 Modes and Time Scales in Power Systems

A power system can be modeled as a set of nonlinear differential equations and

algebraic equations. Small-signal stability analysis can be used to investigate system behavior under small disturbances. In this context, the system can be linearized for the purpose of analysis.

Suppose an unforced dynamic system is defined as the following,

$$\dot{x}(t) = Ax(t); \qquad x(0) = \zeta \tag{2.1}$$

The solution will be,

$$x(t) = \exp(At)\zeta = \sum_{i=1}^{n} \exp(\lambda_i t) v_i [w_i' \zeta] \tag{2.2}$$

where λ_i is the i^{th} eigenvalue of matrix A, while v_i and w_i are its right and left eigenvectors respectively.

The definition of the mode in this context is as follows: the i^{th} mode is $\exp(\lambda_i t)v_i$, which is defined by the direction of right eigenvector v_i and the time-domain characteristic of associated eigenvalue λ_i.

It can be seen that the dynamic behavior of state x is actually a linear combination of the dynamic behavior of modes in the linear system. The elements in the right eigenvector v_i quantifies the contribution of mode i on the particular state.

This concept is important for the understanding of the grouping algorithm of slow-coherency theory. Slow-coherency analysis shows that partitioning according to the r slowest modes will produce the weakest connection between areas. After the r slowest modes are selected, the corresponding columns of the modal matrix will determine the effect of the selected modes on the state variables. If two rows of the eigenvector matrix have the same entries corresponding to the r modes, the corresponding machines will be coherent with each other with respect to the selected modes.

Models of large-scale systems involve interacting dynamic phenomena of widely-differing speeds. To analyze the various stability problems, power system dynamics are usually modeled into the following four time scales:

- Long-term dynamics (several minutes and slower): Boiler dynamics, daily load cycles, etc.
- Mid-term dynamics (1-5 min): Load Tap Changers (LTC), Automatic Generation Control (AGC), thermostat-controlled loads, generator over-excitation limiters, etc.
- Transient dynamics (seconds): Generators, Automatic Voltage Regulators (AVR), governors, induction motors, HVDC controllers, etc.
- Practically instantaneous (less than a msec): Electromagnetic and network transients, various electronically controlled loads, etc.

In models of large-scale interconnected systems, dynamics of different speeds are frequently observed. With appropriate partitioning of a power system into areas, the motion of the center of angle associated with each area is much slower than the "synchronizing" oscillations between any two machines in the same area. A physical interpretation of this phenomenon is that the connections between the machines within an area are strong while those between the areas are weak. Therefore, the machines within the same areas interact on a short-term basis. On a long- term basis, when these fast dynamics have decayed, the machines in the same area move together, that is, they are "coherent" with respect to the slow modes. These slow dynamics, which are represented by the area centers of angle, are due to the interaction between groups of machines through the weak connections which may become important in the long term.

2.2.2 The Explicit Singular Perturbation Form

In the slow-coherency approach, singular perturbation techniques can be used to separate larger power systems into slow and fast dynamic sections. The low-frequency oscillations between coherent groups of stiffly-connected machines are referred to as the more relevant slow dynamics and the less significant fast dynamics are the higher frequency oscillations between machines within the coherent groups. [24]

Assume that the state variables of an n^{th} order system can be divided into r "slow" state

y and n-r "fast" state z, that is

$$\begin{aligned} dy/dt &= f(y,z,t), \qquad y(t_0) = y_0 \\ dz/dt &= G(y,z,t), \qquad z(t_0) = z_0 \end{aligned} \tag{2.3}$$

The quasi-steady state approach assumes that the only states used for long-term studies are y, while the differential equations for z are reduced to algebraic or transcendental equations by setting $dz/dt = 0$. The quasi-steady state model is thus

$$\begin{aligned} dy_s/dt &= f(y_s,z_s,t), \qquad y_s(t_0) = y_0 \\ 0 &= G(y_s,z_s,t). \end{aligned} \tag{2.4}$$

An inconsistency of this approach is that the requirement that z_s must be constant due to the assumption made above, is violated by equation (2.3) which defines z_s as a time-variant variable. A rigorous approach is to treat the situation as a two-time scale singular perturbation problem.

A new time variable τ is introduced to express the fast phenomena, defined by,

$$\tau = (t - t_0)/\varepsilon,$$

where t_0 is the initial value of t and ε is a ratio of time scale.

By rescaling G as $g=\varepsilon G$, we get the **explicit** singular perturbation form.

$$\begin{aligned} dy/dt &= f(y,z,t), \qquad y(t_0) = y_0 \\ \varepsilon dz/dt &= g(y,z,t), \qquad z(t_0) = z_0 \end{aligned} \tag{2.5}$$

To investigate quasi-steady state models, it is assumed that

$$dy_f/dt = 0, dz_f/dt = 0$$

It is known that $y_f(t)$ can be any value, and here we assume that $y_f(t)$ is 0 and $y_s(t_0)=y_0$. In the limit as ε is approaching 0, this model defines the quasi-steady states $y_s(t)$, $z_s(t)$ as

$$\begin{aligned} dy_s/dt &= f(y_s,z_s,t), \qquad y_s(t_0) = y_0 \\ 0 &= g(y_s,z_s,t). \end{aligned} \tag{2.6}$$

The value of $z_s(t_0)$ can be obtained from the above equations if $\partial g(y_s)/\partial z_s$ .is non-singular.

To obtain the fast parts of y and z, equation (2.5) is rewritten in terms of the fast time-scale τ,

$$\begin{aligned} dy/d\tau &= \varepsilon f(y,z,t_0+\varepsilon\tau) \\ dz/d\tau &= g(y,z,t_0+\varepsilon\tau) \end{aligned} \tag{2.7}$$

which leads to $dy/d\tau = 0$ as ε approaches 0, which means that the slow variable y is constant in the fast time scale. Also, it is assumed that $dz_s/d\tau = 0$, which yields the following dynamics model in fast time scale,

$$\begin{aligned} &dy/d\tau = 0 \\ &dz_f/d\tau = g(y_0, z_f(\tau)+z_s(t_0), t_0), \qquad z_f(0) = z_0 - z_s(t_0) \end{aligned} \tag{2.8}$$

The separated lower-order models are in error because they assume ε is approaching 0, instead of the actual positive ε. This parameter perturbation is called singular, since the dependence of the solutions of (explicit form) on ε is not continuous. However, in power systems, it is expected that slow state y will be continuous in ε and the discontinuity in fast state z can be corrected by z_f, if we assume that

With well-damped fast modes, the state z rapidly reaches its quasi-steady state z_s.

When the state z exhibits high-frequency oscillations, the state y is still approximated by $y_s(t)$ due to the "averaging" or filtering effect.

An $0(\varepsilon)$ perturbation form of y, z is therefore given by the following, based on the slow model in equation (2.6) and the fast model in equation (2.8).

$$\begin{aligned} y(t) &= y_s(t) + 0(\varepsilon), \\ z(t) &= z_s(t) + z_f(\tau) + 0(\varepsilon) \end{aligned} \tag{2.9}$$

2.2.3 Equilibrium and Conservation Properties in LTI systems

When the model of a two time scale system is expressed in terms of physical variables such as those in power systems, it is often not in an explicit form, which requires that $\partial g(y_s)/\partial z_s$ be nonsingular along $y_s(t)$ and $z_s(t)$. When this condition is violated, the explicit form of the two-time scale model cannot be obtained.

Consider the n-dimensional system,

$$\varepsilon dx/dt = dx/d\tau = A(\varepsilon)x = (A_0 + \varepsilon A_1(\varepsilon))x \tag{2.10}$$

If A_0 is nonsingular, $x \to 0$ as $\varepsilon \to 0$, no slow phenomenon would exist and the system would not have two-time scales. If A_0 is singular with rank p, by letting $\varepsilon \to 0$, the following equation is obtained,

$$dx/d\tau = A_0 x \tag{2.11}$$

It is observed that A_0 has a v-dimensional equilibrium subspace or manifold, as follows,

$$S = \{x : A_0 x = 0\} \tag{2.12}$$

where v is the rank of null space of A_0 and $v+p=n$.

Equation (2.12) indicates that model (2.11) has the **equilibrium property**.

If the rows of a $p \times n$ matrix Q span the row space of A_0, then S can also be denoted as $S = \{x : Qx = 0\}$.

To investigate the conservation property of (2.11), a $v \times n$ matrix P is defined such that it spans the left null space of A_0, that is, $PA_0 = 0$. Therefore,

$$Pdx/d\tau = d(Px)/d\tau = PA_0 x = 0$$

which induces that,

$$Px(\tau) = Px(0), \text{ for all } x(0) \text{ in } R^n. \tag{2.13}$$

This means that for each value of *x(0)*, the trajectory of equation (2.11) is confined to a translation of a *v*-dimensional subspace, defined in (2.13). Therefore, the system has the conservation property. This *v*-dimensional subspace is orthogonal to the rows of *P* and contains the initial point *x(0)*, defined as follows,

$$F_{x(0)} = \{x : Px = Px(0)\} \tag{2.14}$$

Based on these two properties, time scales in nonexplicit models can be examined to make them explicit by defining a set of coordinates. In the fast time scale, slow motions of a two time scale system remain constant (interpreted as an equilibrium property) while fast motions are restricted to a linear manifold (interpreted as a conservation property).

2.2.4 Time Scale Separation in Non-Explicit Models

For the models shown in (2.10), we can define a transformation matrix,

$T = \begin{bmatrix} P \\ Q \end{bmatrix}$, and its inverse, $T^{-1} = [V \quad W]$.

and define new states *y* and *z*, such that

$$\begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} P \\ Q \end{bmatrix} x. \tag{2.15}$$

Therefore, equation (2.10) has been transformed into

$$\begin{aligned} Tdx/dt &= \begin{bmatrix} dy/dt \\ dz/dt \end{bmatrix} \\ &= T(A_0/\varepsilon + A_1(\varepsilon))x \\ &= T(A_0/\varepsilon + A_1(\varepsilon))T^{-1}\begin{bmatrix} y \\ z \end{bmatrix} \\ &= \begin{bmatrix} PA_1(\varepsilon)V & PA_1(\varepsilon)W \\ QA_1(\varepsilon)V & QA_0W/\varepsilon + QA_1(\varepsilon)W \end{bmatrix}\begin{bmatrix} y \\ z \end{bmatrix} \end{aligned} \tag{2.16}$$

that is,

$$\begin{aligned} dy/dt &= A_s(\varepsilon)y + A_{sf}(\varepsilon)z \\ \varepsilon dz/dt &= \varepsilon A_{fs}(\varepsilon)y + A_f(\varepsilon)z \end{aligned} \tag{2.17}$$

where,

$$A_s(\varepsilon) = PA_1(\varepsilon)V, \qquad A_{sf}(\varepsilon) = PA_1(\varepsilon)W$$
$$A_{fs}(\varepsilon) = QA_1(\varepsilon)V, \quad A_f(\varepsilon) = QA_0W + \varepsilon QA_1(\varepsilon)W$$

Equation (2.17) shows an **explicit** form for model (2.10), because $A_f(0)=QA_0W$ is non-singular.

It is of the interest to mention that the concept of equilibrium and conservation can be extended to a non-linear system to induce the explicit form. More detailed information may be obtained in [24].

2.2.5 Coherency and Grouping Algorithms

As mentioned in previous sections, it has been observed that in multi-machine transients after a disturbance some synchronous machines have the tendency to "swing together". Such coherent machines can be grouped into "coherent areas". A coherency-based grouping approach requires the states to be coherent with respect to a selected set of modes σ_α of the system. This approach allows coherency to be examined in terms of the rows of an eigenvector matrix V which can be used to find coherent groups of states.

Most grouping criteria result in coherency states that are disturbance-dependent because they simultaneously treat the following two tasks:

1) Select the modes which are excited by a given disturbance or a set of disturbances,
2) Find the states with the same content of disturbed modes.

The slow-coherency-based approach only addresses the second task, that is, how to find coherency states for a given set of the r slowest modes. The selection of the slowest modes results in slow coherent groups such that the areas of the system are partitioned along the

weakest boundaries. Detailed information may be obtained in [24].

In this approach, disturbances are modeled as initial conditions. Therefore, a linear system may be modeled as the following form,

$$\dot{x} = Ax, \qquad x(0) = x_0 \tag{2.18}$$

where the state x is an n-vector.

Suppose $\sigma_\alpha = \{\lambda_1, \lambda_2, ..., \lambda_r\}$, where λ_i is an eigenvalue of A associated with a dominant mode. The definition of **Coherency** is that the states x_i and x_j are coherent with respect to σ_α if and only if the σ_α-modes are unobservable from z_k, where z_k is defined as x_j-x_i.

This definition implies that coherent states have the same impact as dominant modes on dynamics, which means the relative rows of V are identical. Modes with high frequency and high damping are neglected in long-term studies. By concentrating only on the σ_α-modes the coherency study will be independent of the location of disturbance.

For an n-machine power system, the classical model is defined as the following,

$$\begin{aligned} \dot{\delta}_i &= (\omega_i - \omega_0)\omega_R \\ 2H_i\dot{\omega}_i &= -D_i(\omega_i - \omega_0) + (P_{mi} - P_{ei}) \end{aligned} \tag{2.19}$$

where,

δ_i Rotor angle of machine i in radians,

ω_i Speed of machine i, in per unit (pu),

ω_0 Reference speed, in per unit (pu). Here $\omega_0 = 1$

P_{mi} Mechanical input power of machine i, in pu,

P_{ei} Electrical output power of machine i, in pu,

H_i Inertia constant of machine i, in seconds,

D_i Damping constant of machine i, in pu,

ω_R Base frequency, in radians per second. (376.99 rad / s)

In this model, the mechanical input power P_{mi} is assumed to constant. The electrical output power is

$$P_{ei} = \sum_{j=1, j\neq i}^{n} V_i V_j [B_{ij} \sin(\delta_i - \delta_j) + G_{ij} \cos(\delta_i - \delta_j)] + V_i G_{ii}^2$$

$$i = 1, \ldots, n$$

where V_i is behind transient reactance the machine per unit voltage, which is assumed to be constant. Loads are modeled by constant impedance, such that load buses may be eliminated from the Y_{bus} matrix. G_{ij} and B_{ij} are the real and image entries of Y_{bus}.

Linearizing the model about the an equilibrium operating point,

$$\begin{aligned} \Delta\dot{\delta}_i &= \omega_R \Delta\omega_i \\ 2H_i\dot{\omega}_i &= -D_i\Delta\omega_i + \sum_{j=1}^{n} \bar{k}_{ij}\Delta\delta_j \end{aligned} \tag{2.20}$$

where

$$\bar{k}_{ij} = V_i V_j [B_{ij} \cos(\delta_i - \delta_j) - G_{ij} \sin(\delta_i - \delta_j)], \quad j \neq i$$

$$\bar{k}_{ii} = -\sum_{j=1, j\neq i}^{n} \bar{k}_{ij}$$

Neglecting the damping constants which do not significantly change the mode shape and the line conductance which are relatively small compared with the line reactance, a second order dynamic model can be obtained,

$$\ddot{X} = M^{-1}KX = AX, \quad X(0) = X_0 \tag{2.21}$$

where

$$x_i = \Delta\delta_i$$

$$m_i = 2H_i / \omega_R$$

$$M = diag(m_1, m_2, ..., m_n)$$

$$k_{ij} = V_i V_j B_{ij} \cos(\delta_i - \delta_j), \quad \text{j} \neq \text{i}$$

$$k_{ii} = -\sum_{j=1, j\neq i}^{n} k_{ij}$$

It has been observed that matrix K has a zero eigenvalue with eigenvector u where $u = [1\ 1\ ...\ 1]^T$. Furthermore, K is symmetric if B is symmetric which is true for transmission networks without phase shifters. In general, B_{ij} are positive and $\delta_i - \delta_j$ are small, which implies that K is a negative semi-definite matrix and the eigenvalues of A are non-positive.

Similar to the first order dynamic system, same implication is applicable in the second order dynamic system.

Starting with (2.21), assuming,

$$x_1 = x, x_2 = \dot{x}$$

Equation (2.21) may be rewritten as,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & I_n \\ A & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{2.22}$$

Assume V to be a σ_α-eigenbasis matrix of A, and $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_r)$. Based on $AV = V\Lambda$, it is easy to obtain

$$\begin{bmatrix} 0 & I_n \\ A & 0 \end{bmatrix} \begin{bmatrix} V & 0 \\ 0 & V \end{bmatrix} = \begin{bmatrix} 0 & V \\ AV & 0 \end{bmatrix} = \begin{bmatrix} V & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} 0 & I_r \\ \Lambda & 0 \end{bmatrix} \tag{2.23}$$

which means that

$$\begin{bmatrix} V & 0 \\ 0 & V \end{bmatrix}$$

is a σ_α-eigenbasis matrix of

$$\begin{bmatrix} 0 & I_n \\ A & 0 \end{bmatrix}$$

From the definition, x_i and x_j are coherent if and only if the i^{th} and j^{th} rows of V are identical. This implies that to examine the coherency of the second order system such as that of (2.21), only the σ_α-eigenbasis matrix of A is required.

Usually in the real dynamic network of a real system, the coherency definition may not be exactly satisfied. Thus, if this definition is applied to a real system, there will be, in general, more coherency groups than the number of modes in σ_α, which means that there are too many groups to be used in islanding. As a result, an approach to finding near-coherent groups will be presented such that the total number of near-coherent groups is equal to the number of modes in σ_α. The areas formed by these near-coherent groups are still coherent with small perturbation.

The coherency based grouping algorithm has been summarized as follows: [24]

1) Choose the number of groups and the set of the slowest modes σ_α.
2) Compute a basis matrix V of the $\sigma\alpha$-eigenspace for a given ordering of the x variables containing slow modes.
3) Apply Gaussian elimination with complete pivoting to V and obtain the set of reference machines. Each group will then have one and only one reference machine. V_1 is the matrix composed of the rows of the matrix V related to the reference machines.
4) Compute $L = VV_1^{-1}$ for the set of reference machines chosen in step 3).
5) Determine the group that each generator belongs to from the matrix L by comparing the row of each generator with the row of the reference machines.

A 3-machine system will be chosen to illustrate this coherency based grouping

algorithm. Suppose two slowest modes have been chosen and the σ_α-eigenspace matrix V has the following form:

$$\begin{array}{ccc} & \lambda_1 & \lambda_2 \\ x_1 & 0.577 & -0.287 \\ x_2 & 0.577 & 0.827 \\ x_3 & 0.577 & 0.483 \end{array} \tag{2.24}$$

The procedure of Gaussian elimination with complete pivoting will be shown in the following steps.

The largest number in (2.24) is 0.827. Therefore, the first and second rows and the first and second column can be exchanged to obtain,

$$\begin{array}{ccc} & \lambda_2 & \lambda_1 \\ x_2 & 0.827 & 0.577 \\ x_1 & -0.287 & 0.577 \\ x_3 & 0.483 & 0.577 \end{array}$$

Then, the number 0.827 can used as a pivot to eliminate the remainder of the first column. The result can be shown as follows,

$$\begin{array}{ccc} & \lambda_2 & \lambda_1 \\ x_2 & 0.827 & 0.577 \\ x_1 & 0 & 0.831 \\ x_3 & 0 & 0.239 \end{array}$$

Excluding the first row and first column of the matrix V, the largest number is 0.831, and the procedure terminates because all the pivots have been found and the reference states are x_2 and x_1, shown as follows,

$$V_1 = \begin{bmatrix} 0.577 & 0.827 \\ 0.577 & -0.287 \end{bmatrix}$$

Therefore,

$$L = VV_1^{-1} = \begin{bmatrix} 0.577 & -0.287 \\ 0.577 & 0.827 \\ 0.577 & 0.483 \end{bmatrix} \begin{bmatrix} 0.577 & 0.827 \\ 0.577 & -0.287 \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} 0.577 & -0.287 \\ 0.577 & 0.827 \\ 0.577 & 0.483 \end{bmatrix} \begin{bmatrix} 0.4465 & 1.2866 \\ 0.8977 & -0.8977 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0.6912 & 0.3088 \end{bmatrix}$$

It can be concluded that machine 2 and machine 3 are with the same coherent group and machine 1 itself is another coherent group since the number 0.6912 is closer to 1 than the number 0.3088.

In summary, slow coherency assumes that the state variables of an n^{th} order system are divided into r slow states Y, and $(n\text{-}r)$ fast states Z, in which the r slowest states represent r groups with the slow coherency.

Slow coherency solves the problem of identifying the theoretically weakest connection in a complex power system network. Previous work shows that groups of generators with slow coherency may be determined using Gaussian elimination with complete pivoting on the eigensubspace matrix after selection of r slowest modes σ_a. In [8], it has been proven through linear analysis that with selection of the r slowest modes, the aggregated system will have the weakest connection between groups of generators.

The weak connection form best states the reason for islanding based on slow coherency grouping. That is, when the disturbance occurs, the slow dynamics in the transient time scale must be separated, which could propagate the disturbance very quickly, by islanding on the weak connections. The slow dynamics will mostly remain constant or change slowly on the tie lines between the areas.

Slow coherency is actually a physical manifestation of a weak connection, which is a

network characteristic. In many large-scale practical systems, there always exist groups of strongly interacting units with weak connections between groups. However, weak connections can become strong connections with significant interactions after a long time interval. When a large disturbance happens, it is imperative to disconnect the weak connections before the slow interaction becomes significant.

2.3 Graph Theoretic Terminology

Graph theory has developed as a branch of mathematics during the second half of 19^{th} century, and has boomed since 1930. The Swiss mathematician Leonard Euler (1707-1783) is undoubtedly the father of graph theory. His famous problem of the Bridge of Königsberg, has been viewed as the first problem in graph theory. Graph theory is basically the mathematical study of the properties of formal mathematical structures called graphs. Although mathematicians are responsible for much of its development and growth, sociologists and engineers alike are looking enthusiastically toward graph theory to solve problems in their fields [25], [26].

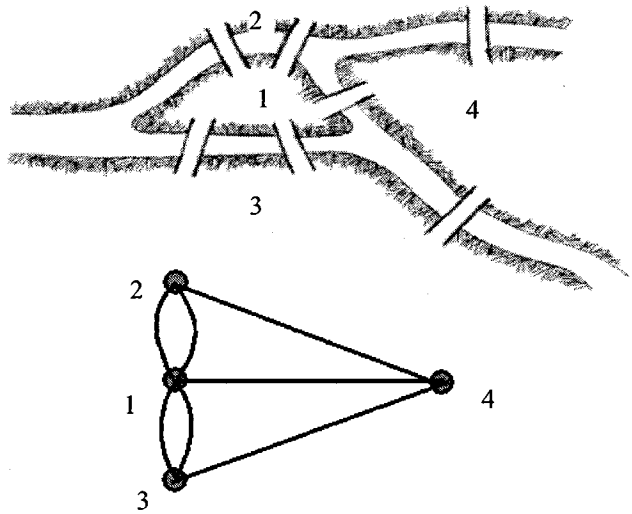


Figure 2-1 Konigsberg Bridge and its graph representation

Definition 2.1 ***Graph:*** A **graph** is a pair of sets (V, E) where V is the vertex-set and E the

edge-set is a family of pairs (possibly directed) of V. It is usually denoted as $G \equiv G(V, E)$. Graph are simple abstractions of reality. In this sense, graphs are diagrammatical models of systems. However, not every system can be represented in the form of a graph. As a general rule, any system involving **binary relationships** can be represented in the form of a graph.

Definition 2.2 ***Connected Graph:*** A graph is connected if there is a path connecting every pair of vertices. A graph that is not connected is said to be disconnected. Its vertices V can be divided into two nonempty subsets V_1 and V_2 such that vertices of V_1 are not adjacent to those of V_2. A subgraph $G_1(V_1,E_1)$ of a graph $G(V, E)$ is a graph with vertices V_1 and edges E_1 such that $V_1 \subset V$ and $E_1 \subset E$. A maximal connected subgraph is called a **component** of a graph. Clearly a graph is connected if it has only one component. The subgraph inside the dashed circle shown in Figure 2-2 one of the components of a graph E.

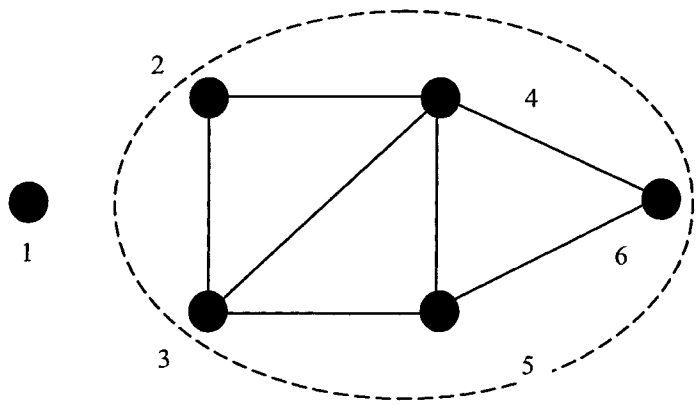


Figure 2-2 Illustration a component of a graph

Definition 2.3 ***Minimal Cutsets (MC)*****:** A **disconnecting set** of a connected graph $G(V, E)$ is a set of edges $E_1 \in E$ such that after the removal of E_1 the residual graph $G_1(V_1,E-E_1)$ is no longer connected. This set of edges is called **cutset.** For a given graph $G= (V, E)$, a subset of edges $E_1 \subset E$ is a minimal cutset if and only if deleting all edges in C would divide G into two connected components. It is also called **proper cutset**.

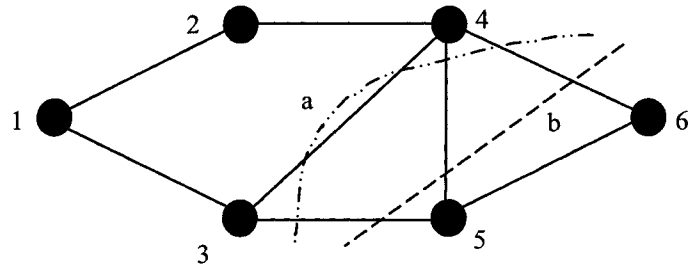


Figure 2-3 Illustration of a) a cutset and b) a minimal cutset

Definition 2.4 ***Vertices Contraction (VC)*****:** Given a graph G and one adjacent vertices pair $\{x, y\} \in V$, we define $G/\{x, y\}$, the vertices contraction of pair $\{x, y\}$, by deleting x and replacing each edge of the form $\{w, x\}$ by an edge $\{w, y\}$. If this process creates parallel edges, only one edge will remain in the graph. Any self-loops are also eliminated.

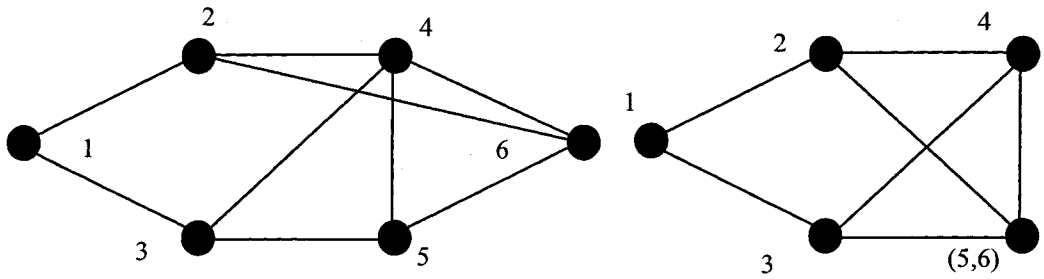


Figure 2-4 Illustration of vertices contraction on vertex 5 and 6

Definition 2.5 ***Depth first Search (DFS)*****:** *Depth first search* is a graph search algorithm which extends the current path as far as possible before backtracking to the last choice point and trying the next alternative path. Extremes are searched first. DFS tends to require less memory, as only nodes on the "current" path need to be stored. However, DFS may fail to find a solution if it enters a cycle in the graph. This can be avoided if we never extend a path to a node which it already contains. DFS can be easily implemented with a *recursion* process.

Definition 2.6 ***Breadth first Search (BFS):*** *Breadth first search* is a graph search algorithm which tries all one-step extensions of current paths before trying larger extensions.

This requires all current paths to be kept in memory simultaneously or at least their end points. Extremes are searched last. Compared to Depth first search, breadth first search does not have a cycling problem. Usually, BFS can be realized with a queue. Modified Breadth first search tree (BST), which is a tree constructed through BFS with specific properties.

Definition 2.7 ***Minimum Spanning Tree (MST)*****:** A connected, undirected acyclic graph is called a *tree*. *Spanning Trees* are trees that are subgraphs of G and contain every vertex of G. In a weighted connected graph $G = (V, E)$, it is often of interest to determine a spanning tree with minimum total edge weight – that is, such that the sum of the weights of all edges is minimum. Such a tree is called *Minimum Spanning Tree*.

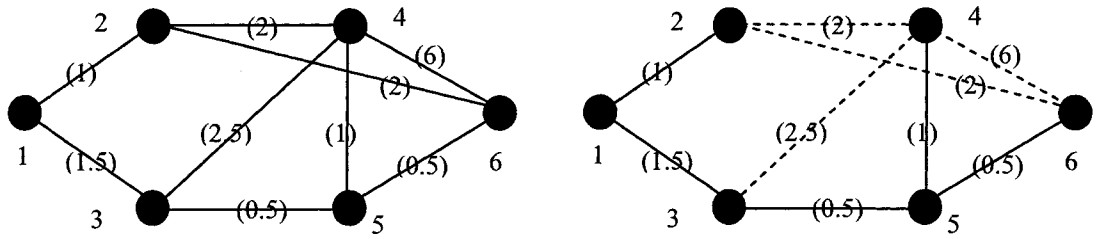


Figure 2-5 Illustration of the minimum spanning tree

Definition 2.8 ***Steiner Tree***: A minimum-weight tree connects a designated set of vertices, called terminals, in a weighted graph or points in a space. The tree may include non-terminals, which are called *Steiner vertices* or *Steiner points*. The Steiner tree problem is distinguished from the minimum spanning tree problem in that we are permitted to select intermediate connection points to reduce the cost of the tree.

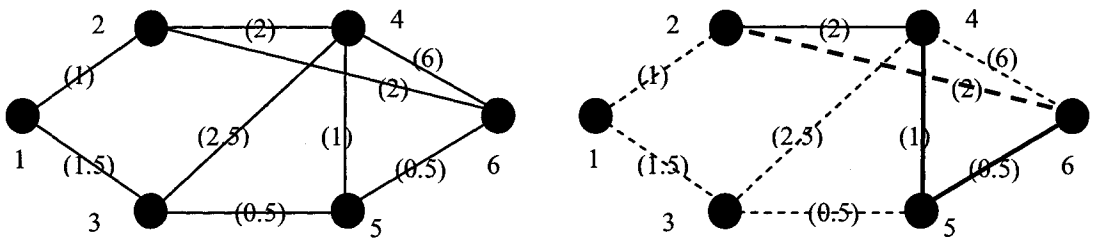


Figure 2-6 Illustration of the difference between a Steiner tree and a minimum spanning tree

To find a path connecting vertices 2, 4 and 6 with a minimum cost, a Steiner tree will consist of vertices 2, 4, 5 and 6, where vertex 5 is the Steiner point. However, a minimum spanning tree will consist of vertices 4, 2 and 6.

2.4 Realization in Power System

2.4.1 Motivation

Power systems are composed of buses and transmission lines connecting the buses. There are both generator buses and load buses with various capacities. Electrical power flows among those transmission lines in certain directions. Therefore, it is very convenient to consider a power system network as a directed graph with different weights at its vertices.

Figure 2-7 illustrates the diagram of one typical 3-generator 5-bus system and its graph representation. It can be seen that the graph is only the representation of the binary relationship between the pairs of buses in the system.

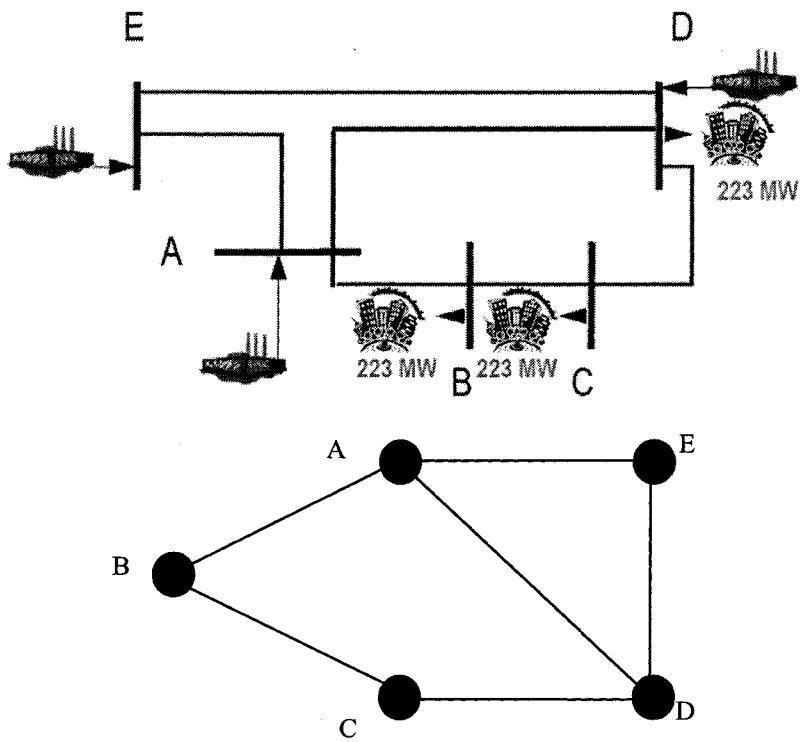


Figure 2-7 Illustration of a typical 3 generator 5 bus system and its graph representation

One of the most important requirements for islanding is to minimize the real power imbalance within the islands to benefit restoration. After an island is formed, the imbalance between the real power supply and load demand is usually calculated by computing all the generator vertices and load vertices, which requires a great deal of computation [6]. One may ask the question: What if we consider the branches connecting this island with other islands instead of browsing all vertices within this island? This intuitively makes sense, because most of the time, the number of tripping lines is limited in order to form an island.

The power flows in the transmission line also contain information about the distribution of generators throughout the system. Once the island is formed, the net flow in the tripping lines indicates the exact difference between the real generation and the load within the island (we assume that the losses can be ignored without the loss of generality).

Therefore, the problem can be converted into one of searching the minimal cutsets (MCs)

to construct the island with the minimal net flow. We can decompose the islanding problem into two stages:

1. Find Minimal Cutsets candidates
2. Obtain Optimal Minimal Cutset by various criteria

Generally, the edge-searching approach may result in inefficiency in computation since generally there are more edges than vertices in the network. However, most power systems, at the transmission system level, are sparse, which results in little difference between vertices and edges in terms of numbers.

The advantage of this method is that we can decompose the islanding problem into two stages: In the first stage, we find the cutsets disconnecting the sets of generators; in the second stage, we check the net flow on each cutset to obtain the optimal cutset. Another advantage is that, in the second stage, we can apply any additional criteria to formulate the optimization function under different conditions, such as the requirements for system restoration, while the first stage remains unchanged.

Other advantages of this method are that, besides the general criteria mentioned previously, other user-specified requirements can also be included during islanding, such as,

1. Specification as to which lines may not be disconnected. This is simply done by blocking such a line from the cutsets' candidates.
2. Specification as to which area will remain untouched. This can be done by aggregating such an area into one bus.

2.4.2 Software Structure

In order to demonstrate the applicability of this idea, an automatic power system islanding program has been developed to automatically determine where to create an island using minimal cutsets and a breadth first searching (BFS) flag-based depth-first searching (DFS) technique based on Graph Theory.

Figure 2-8 illustrates the software structure of this approach.

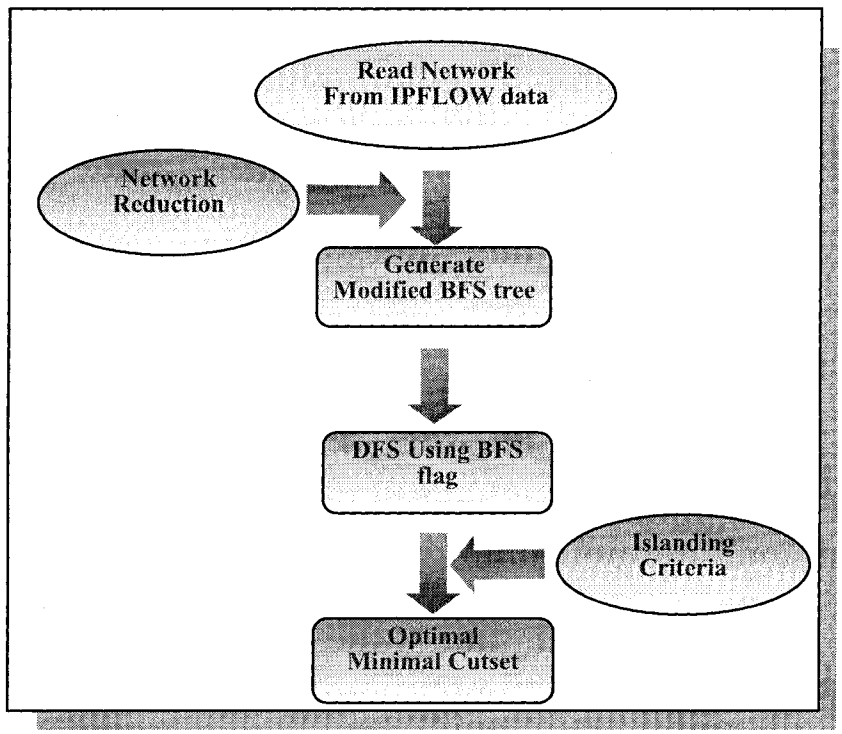


Figure 2-8 Software structure

The four main components are:.

1. Network reduction
2. Generation of a modified BFS tree with no offspring in sink vertex
3. Conduct of a DFS search with a BFS flag to enumerate all possible MCs
4. Application of islanding to select the optimal MC.

Network reduction

As one of the main components in this approach, network reduction plays a significant role in islanding performance and optimality as the network scale increases. Performance and Optimality are two goals that must be dealt with. Here, optimality means the minimum value for our objective function, and performance indicates the computational effort (how long it is

going to take to reach the optimality). It is our goal to achieve global optimality as closely as possible while maintaining high performance. However, there is sometimes a tradeoff between these two factors. As we keep reducing the network by a given criteria, we may also deviate from the globally optimal solution since the searching space has been reduced. A heuristic scheme using a *Contraction Factor* has been provided to make it possible to adjust both the performance and optimality level. The layered representation for Network Reduction is shown in Figure 2-9, which is composed of 3 layers, the original network, pre-reduction, and network reduction composed of first stage reduction and second stage reduction.

Original Network: In the stage, bus numbers are directly read from PSAPAC ipflow file.

Pre-Reduction: Bus numbers are re-ordered beginning with the number 1, since the index of the MATLAB vector variable begins with 1. The variable *bus* is used to record the actual bus numbers.

First Stage Reduction: Buses with degree one are reduced from the network. The network scale is reduced. Two functions ***fs2pr*** and ***pr2fs*** are used to convert the bus numbers from the first stage reduction index to pre-reduction

Second Stage Reduction: Vertices contraction is applied in this stage, in which the network will be greatly reduced to a reasonable scale. Vertices categorization basically investigates each system vertex and decides which generator group it belongs to. This is also referred to as the ***Generator Bus Extension Process***, after which the uncategorized vertices construct the area from which the minimal cutset will be obtained. The procedure can be illustrated as follows:

1. find buses to connect generator buses with each other. This can be done by searching using the Steiner tree technique.
2. for each remaining bus, find the shortest path to the generator buses in other areas. If the path crosses the extensive generator buses, it means that this bus is located inside the boundary Π formed by the extensive generator buses. Update the extensive generator buses boundary by adding part of the path.

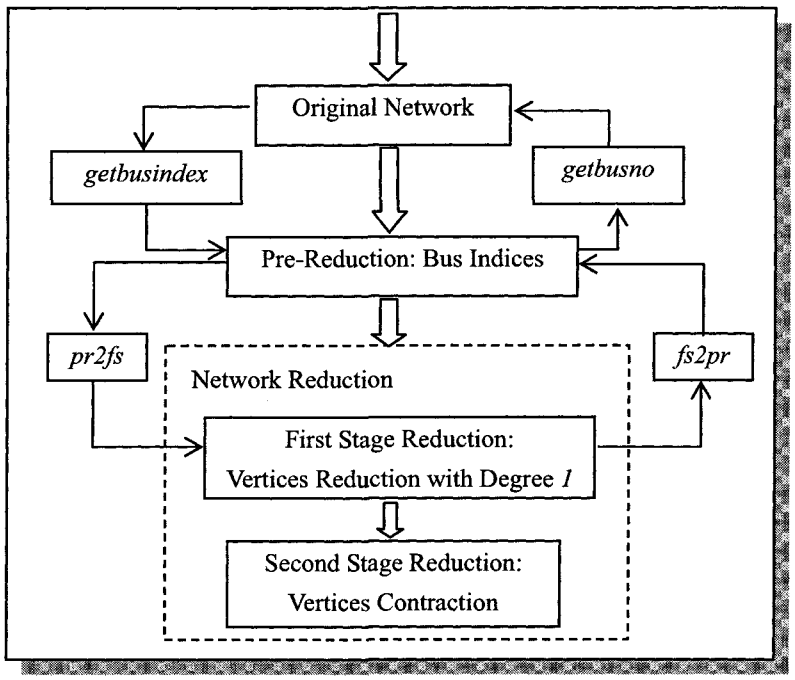


Figure 2-9 Layered representation of Reduction Component

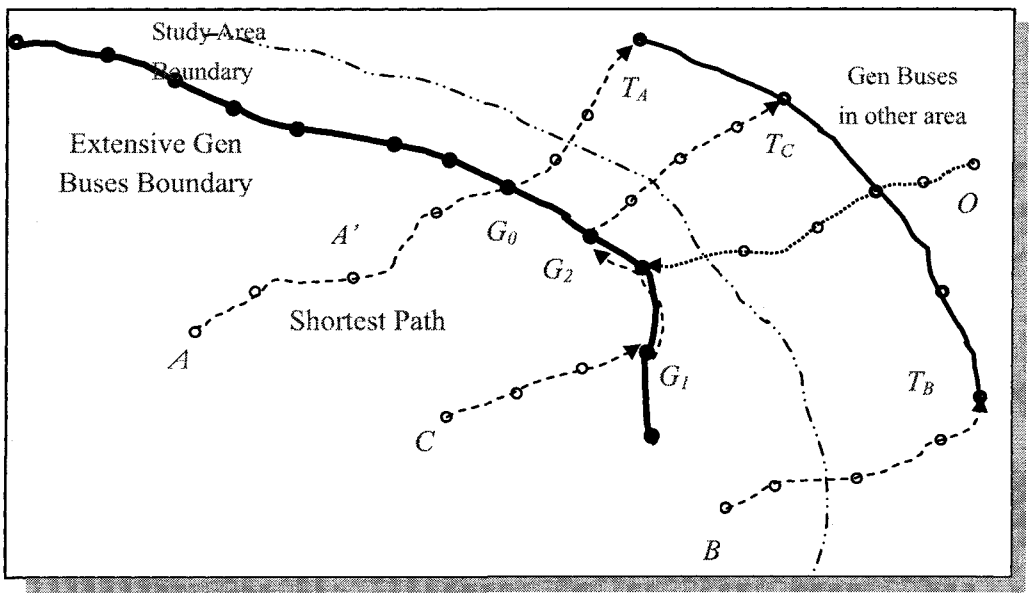


Figure 2-10 Generator Bus Extension Process

The Modified BFS tree based DFS approach

A modified BFS tree is generated such that there is no offspring for the sink vertex. It is

used as a flag to ensure that there is no reexamination of recurring subsets when a DFS is conducted to traverse the reduced graph generated from network reduction. This is because when the graph contains a cycle graph C_k, DFS will visit some vertices several times. Each vertex in the BFS tree has an ordered number. The inclusion-exclusion principle is used to organize the vertices according to their position in the BFS tree of the graph. To do this, all vertices in the outline that have a lower order BFS are omitted.

The following figure shows pseudo code of the BFS tree flag-based DFS searching algorithm.

```
/* FINDALLCUTSET (adjmatrix, v, t, F) */
/* adjmatrix is the adjacency matrix of the graph.
/* v and F are initialized as the source vertex, t is initialized as
/* the sink vertex.
If F has not been recorded
  Record F into cutsets set;
Find the outline of F excluding t;
Remove the lower order vertices than v which has been already taken from the
  outline;
For each vertex in the remaining outline of v
  Add this vertex into F;
  Update v to the vertex in F with the lowest order;
  FINDALLCUTSET (origmatrix, v, t, F);
```

Figure 2-11 Pseudocode of the BFS tree flag based DFS searching algorithm

2.5 Two Comprehensive Approaches for system wide islanding

As addressed above, by using the proposed approach a feasible solution to the islanding problem can be found. Without loss of generality, consider the islands formed in Figure 2-12. H_1, H_2, and H_3 are the total inertia of the load-rich islands; H_4 is the total inertia of a generation rich island.

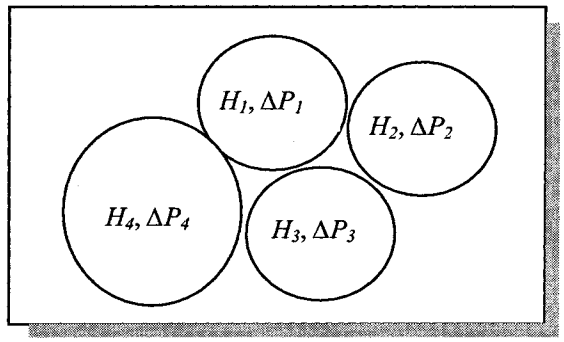


Figure 2-12 Islands with feasible cutsets

From the load-generation balance point of view, the optimal solution is to minimize the net flow of each of the islands H_1, H_2, and H_3, while maintaining $\Delta P_i/H_i$ constant among the islands H_1, H_2 and H_3. This means that the average real power imbalance per inertia should kept the same as nearly as possible among those load-rich islands. Here reactive power requirements and other restoration criteria have not been taken into consideration.

Two applicable approaches to deal with this optimization are presented in the following:

1. Tuning Trial-Error Iterations

Generator rotor speed deviation is captured by the swing equation, shown in (2.19), if the damping effect is ignored. It is observed that the change of generator rotor speed deviation is determined by the ratio of the difference between mechanical input and electrical output over the machine inertia, denoted as $\Delta P_{m,i} / H_{m,i}$. Furthermore, it is our intention to maintain the frequency decline almost the same across all the islands. Therefore, by extending this concept from the generator to the island, a Tuning Index *TI* is first defined to indicate the degree to which each island needs to be tuned,

$$TI = \frac{\Delta P}{H} \tag{2.25}$$

Obviously, islands with higher values of $\Delta P/H$ have a higher potential to be tuned if real power imbalance is of concern. These values are expressed as a vector $[\Delta P_i/H_i]$ denoted as the *TI* vector.

The algorithm will then expand the islands having the smallest *TI* among those which have intersections with the islands having the largest *TI*. The aim is to reduce the largest *TI*, which increases the smallest *TI*.

An island can be expanded by including its outline. However, one should keep in mind that the expansion should exclude the generators in other islands. Minimum spanning tree (MST) or Steiner tree techniques can be used to keep the generator buses from being included. This would also give maximal space for neighboring islands to expand.

For the example considered in Figure 2-12, suppose H_1 has the largest *TI*, and H_2 has the smallest *TI* among those islands which intersect with H_1. H_2 will be expanded by including its outline.

In general this approach will not reach the optimal solution in a single tuning procedure. Several iterations are needed until the error (as computed by Equation 2.26), is less than a specified tolerance.

$$\varepsilon = \sqrt{\sum_i \frac{\left(\frac{\Delta P_i}{H_i} - \frac{\overline{\Delta P}}{H}\right)^2}{n}} \tag{2.26}$$

where, $\frac{\overline{\Delta P}}{H} = \frac{1}{n}\sum_i \frac{\Delta P_i}{H_i}$

2. Aggregated Island Approach

An alternative for finding the optimal cutset for all islands will be addressed below:

1) Based on the Tuning Indices, find the reasonable cutsets for all the generator groups.
2) Determine the load-rich islands.
3) Consider all those generators in interconnected load-rich islands as one group, and determine the minimal cutsets for this aggregated group with minimal net flow, which corresponds to the aggregated islands.

4) Assume that once the minimal cutset for the aggregated group is acquired the optimal cutset for these individual groups can always be found.
5) Calculate the load-generation imbalance within the aggregated islands. If only the load-generation imbalance is considered, index $\Delta P_i/H_i$ among those individual islands should be maintained to be equal. By applying this principle, the load-generation imbalances within each individual island can be calculated.
6) Taking other criteria associated with restoration into account; and based on appropriate priority indices, the islanding procedure can be re-run again with an estimation of the load-generation imbalance within each island

If some load-rich islands are interconnected, the minimal cutsets for the aggregated island is the combination of the minimal cutsets. Here only one aggregated island is taken into consideration. For a system which is comprised of multiple aggregated islands, method A should be used. First, the number of islands existing in the system should be determined. Second, by using method (B), connected islands are considered as one island, and only isolated islands are taken into account. Next, using method (A), tune each isolated to reach the condition where equation (1) in [27] holds. The procedure is shown schematically in Figure 2-13.

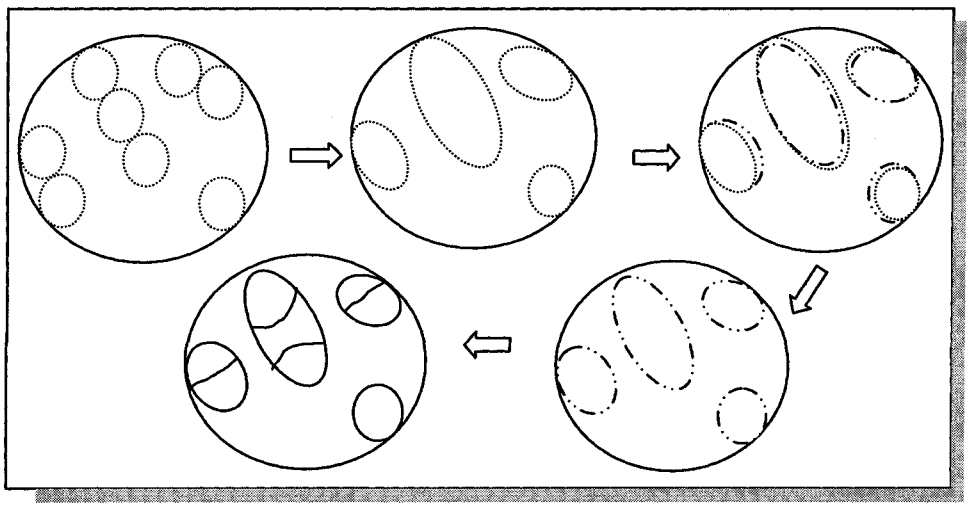


Figure 2-13 Final approach to system islanding

For an aggregated island containing less than two individual islands, separation is much simpler However, in the case where there are more than two islands within an aggregated island, each separate island needs to be calculated and identified; consequently. we need to first specify the source vertices *S* which represent the generators to be identified within the island, and sink vertices *T*, which will be generators in the remainder of the aggregated island, as shown in step 1, Figure 2-14. This procedure will continue as shown in step 1, Figure 2-14, until no more islands need to be identified.

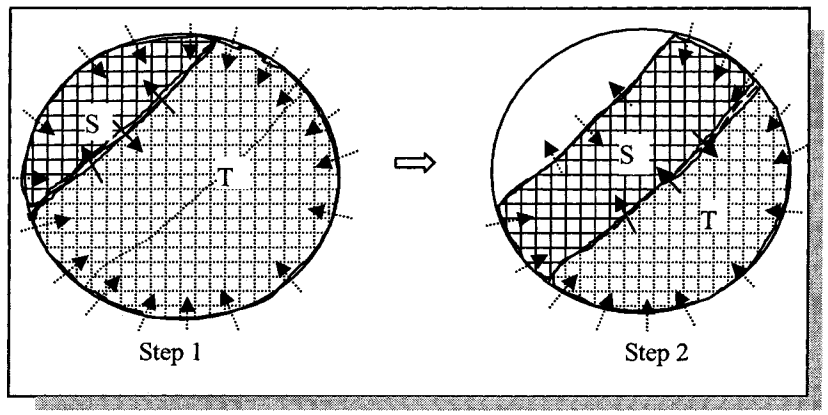


Figure 2-14 Separate individual islands in one aggregated island

2.6 New Governor Model

So far, a novel approach has been presented for performing power system islanding. This method is independent of the models of generators, exciters and governors. However, more accurate models are needed to represent a real system when time-domain simulation is conducted. In these simulations, frequency response needs to be identified and a load shedding scheme will also be designed.

Studies have demonstrated that representing base loading of generators and generator load controllers has a dramatic positive effect on simulation results, not only in frequency deviation studies (reserve, under frequency load shedding, etc.), but also on the results of many system stability studies, such as those used to set transfer limits, remedial action, etc.

Simulations of real-time events, including staged and random generator trips in the WECC system [28], have indicated that there is a wide difference in the frequency response values produced by simulations and those recorded by disturbance-monitoring equipment. Differences of the order of 50% to 60% have been noted in both transient peaks and "settling" frequencies. Governor and load-modeling issues were highlighted in previous work during 2000 by the Task Force of the WECC's Modeling & Validation Work Group (M&VWG) for further investigation.

In [28],[29], the WECC Modeling & Validation Work Group has proposed a new governor model for the WECC system to resolve wide differences in the frequency response values produced by simulations and those recorded by disturbance-monitoring equipment.

2.6.1 Model description

Two new models have been developed for use in WECC studies. The *ggov1* model referenced in [29] is a generic thermal governor/turbine model that incorporates base loading and a load controller, as shown in Figure 2-15. The model, *lcfb1* [29], is identical in structure to the load controller portion of *ggov1*, and can be used in tandem with any governor model currently defined in any power system transient stability program, as shown in Figure 2-16.

Thermal plants not currently modeled with a governor in the WECC database should be added using the *ggov1* model. All gas turbine units should use the *ggov1* model. Hydro units that operate under load control should use the *lcfb1* model in addition to the appropriate hydro governor model.

Existing *ieeeg1* models may be used with the addition of the *lcfb1* load-controller model if it applies. Alternatively, the new *ggov1* model may be used for such units with appropriate data supplied for it.

Upon initialization, base-loaded units and load-controllers are assigned values equal to the generator dispatched value specified in the power flow data in the *ggov1* and *lcfb1* models. If

the effects of a load (or any set point other than frequency) controller are to be included, the output of the unit will be reset to the value of P_{MWSET}. The speed at which the resetting takes place is controlled by the value of K_{IMW} (K_I in model *lcfb1*).

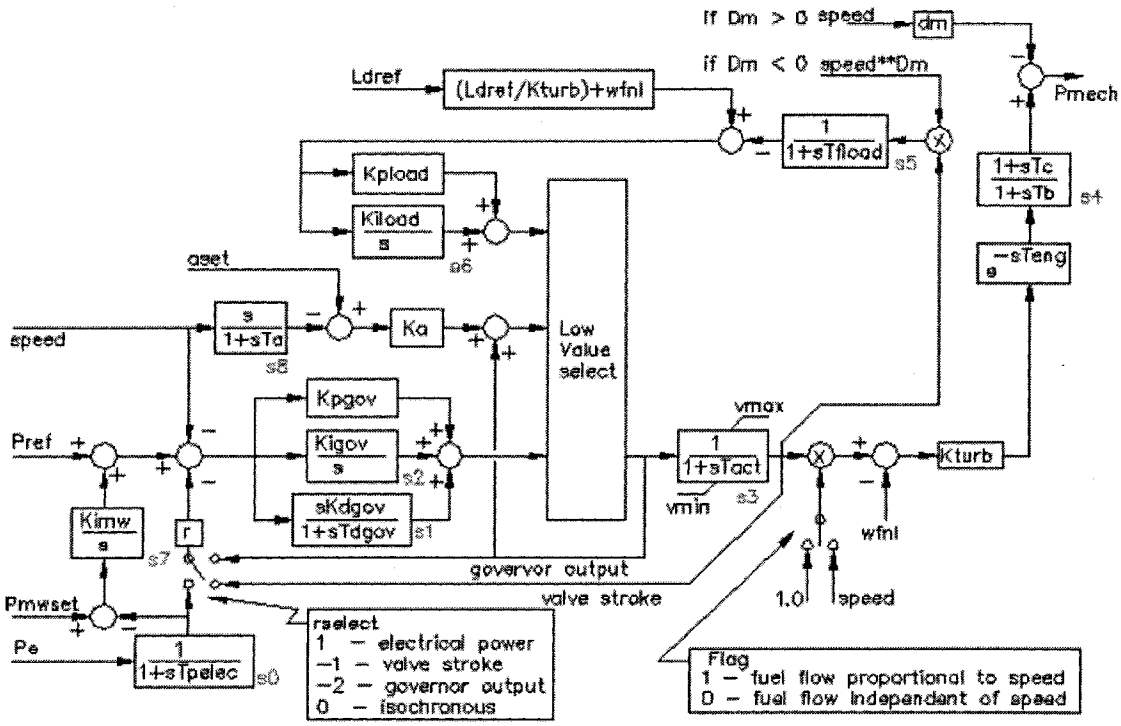


Figure 2-15 Generic thermal governor/turbine model

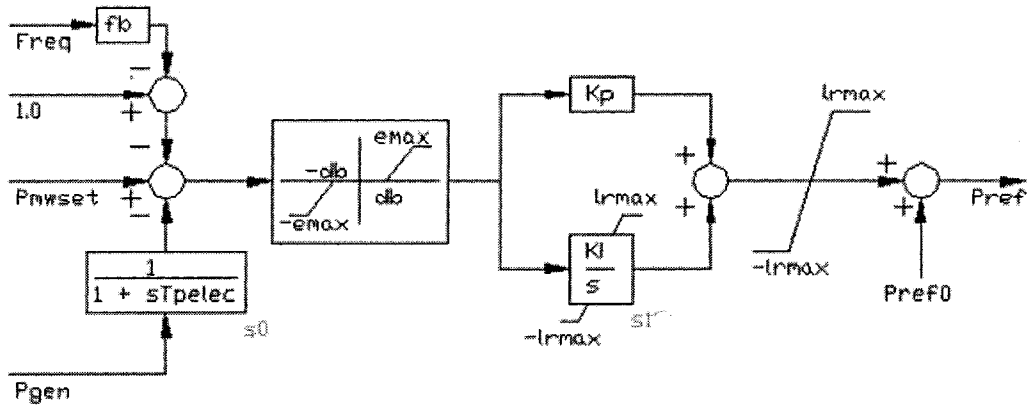


Figure 2-16 Load controller potion of governor model

2.7 Adaptive Load Shedding

An adaptive load shedding scheme is required to help preserve the security of generation and interconnected transmission systems during major system frequency declining events. Such a program is essential to minimize the risk of total system collapse; to protect generating equipment and transmission facilities against damage; to provide for equitable load shedding (interruption of electric supply to customers), and to help ensure the overall reliability of the interconnected systems.

Load shedding resulting from a system under-frequency event should be controlled so as to balance generation and customer demand (load), to permit rapid restoration of electric service to customer demand that has been interrupted, and, when necessary, to re-establish transmission interconnection ties.

In our approach, controlled system islanding divides the power system into islands. Some of these islands are load-rich and others may be generation-rich. Generally, in a load-rich island, the situation is more severe. The system frequency will drop because of the generation shortage. If the frequency falls below a certain set point (e.g., 57.5 Hz), the generation protection system will begin operation and trip the generator, further reducing the generation in the island and making the system frequency decline even more. In the worst case, the entire island will experience blackout.

2.7.1 Definition of load shedding

According to the North American Electric Reliability Council (NERC) definition, load shedding is the process of deliberately removing (either manually or automatically) preselected customer demand from a power system in response to an abnormal condition to maintain the integrity of the system and minimize overall customer outages [30].

2.7.2 General requirements of the automatic under-frequency load shedding

As already pointed out, controlled system islanding is the last resort to prevent a system from total collapse. A load-shedding scheme is the ultimate strategy to prevent total blackout in load-rich islands. This is due to the facts that a continuous generation shortage leads to persistent low frequency, which may activate the unit's protection scheme to trip units out of the system and further decrease the frequency. This may reach to the point that all the units trip out and the system shuts down.

Based on the characteristic and nature of under-frequency load shedding (UFLS), the following aspects should be the major considerations for designing a load shedding scheme: amount of load to be shed at each step, frequency threshold, step size and number of steps, time delay, and priorities.

The literature describes two types of load-shedding schemes: load shedding based on frequency decline and load shedding based on rate of frequency decline. Load-shedding schemes used before the 1980s were almost all based on frequency decline (UFLS). This conventional load shedding scheme has the following disadvantages: 1) longer low-frequency system operation caused by slower UFLS action; 2) possible excess of load shed and associated frequency overshooting.

An adaptive load-shedding scheme which takes the rate of frequency decline into consideration has been proposed.

A threshold value (M_0) is defined in each island, such that, if the rate of frequency decline after islanding at one load exceeds M_0, a new load-shedding scheme will be deployed. Otherwise, a conventional load shedding scheme will be deployed. [31]

$$M_0 = \frac{60 \times P_{L\Delta}}{2\sum H_i}, \quad where \;\; P_{L\Delta} = 0.3 \times P_{sys} \tag{2.27}$$

where $P_{L\Delta}$ is the minimum load deficit that could drive the system frequency down to $57Hz$, which is the minimal operational frequency.

2.8 Results for the Test System

2.8.1 Grouping Results from DYNRED

In this section we will demonstrate the efficacy of slow-coherency-based grouping and automatic islanding by applying minimal cutsets on the WECC 29-Generator 179-Bus test system. The system has a total generation of 61410MW and 12325Mvar. It has a total load of 60785MW and 15351Mvar. The Dynamic Reduction Program (DYNRED) from EPRI's Power System Analysis Package (PSAPAC) [32] was chosen to form groups of coherent generators based on an improved slow-coherency method developed by GE [33] to deal with large systems and achieve more precise results. The user can specify the tolerance value, the number of slow modes, and the number of eigenvalues being calculated. Then, with the help of the automatic islanding program, the optimal minimal cutset of the island may be determined, taking into account the least generation load imbalance and topological requirements.

The DYNRED program has been employed to find groups of generators with slow coherency on the 179-Bus system as a base case. The 29 generators are divided into 4 groups by the slow-coherency program as shown by the dotted lines in Figure 2-17. The dashed lines indicate these four groups. The detailed grouping information has been shown in TABLE 2-1. Fast dynamics are propagated through the weak connections determined by the boundary between groups of generators. To develop a better understanding of the proposed approach, the minimal cutsets between the south island and the rest of the system are first determined. Once the minimal cutset of the south island is found, we can, if necessary, continue to find other islands by removing the south island from the network and treating the rest of network as the whole network.

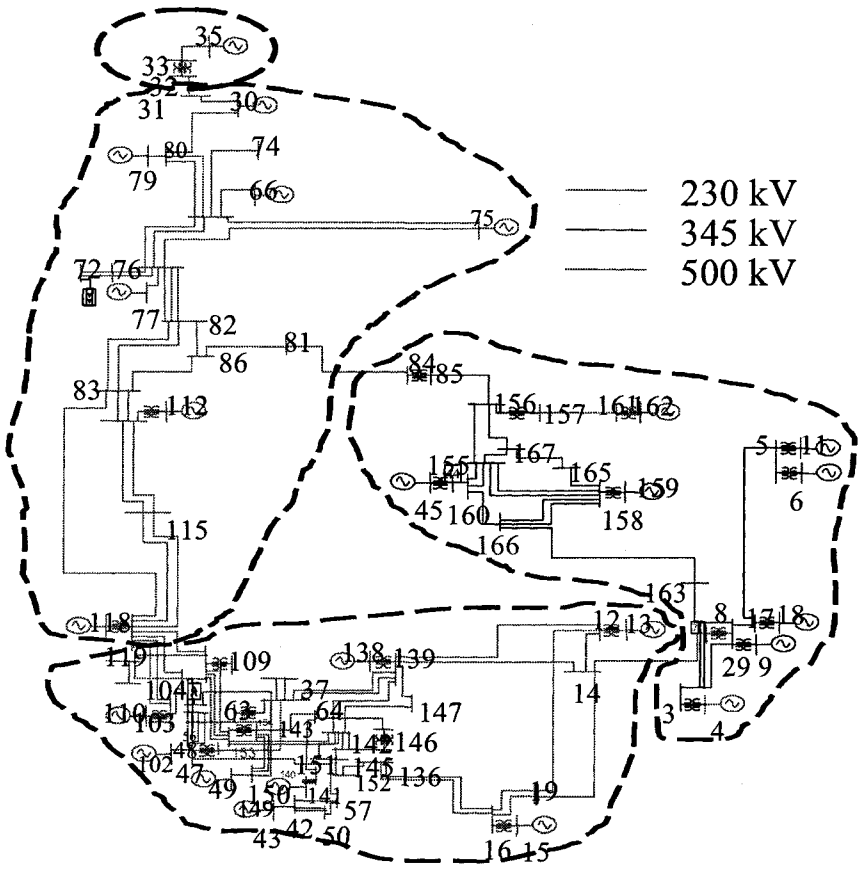


Figure 2-17 Generator groups formed by slow coherency

TABLE 2-1 GENERATOR GROUPING RESULTS FROM DYNRED

Group No.	Generator No.
1	140, 40, 103, 138, 43, 144, 148, 13, 47, 15, 149
2	11, 36, 4, 6, 159, 9, 45, 162, 18
3	35
4	79, 30, 70, 77, 65, 112, 116, 118

2.8.2 Graph Representation

Figure 2-18 denotes the graph representation of the WECC 29-generator, 179-bus system, where the largest font designates the generator buses in the south island and the middle-sized font designates the generator buses in other islands. It can seen that each double circuit is

considered as one edge in the graph, because the controlled system islanding action always disconnects both lines in the double circuit rather than just a single line. This is due to the fact that, other than transmission system reconfiguration, controlled system islanding is mainly intended to change the system topology rather than system flow. Therefore, this simplification will not affect the final result.

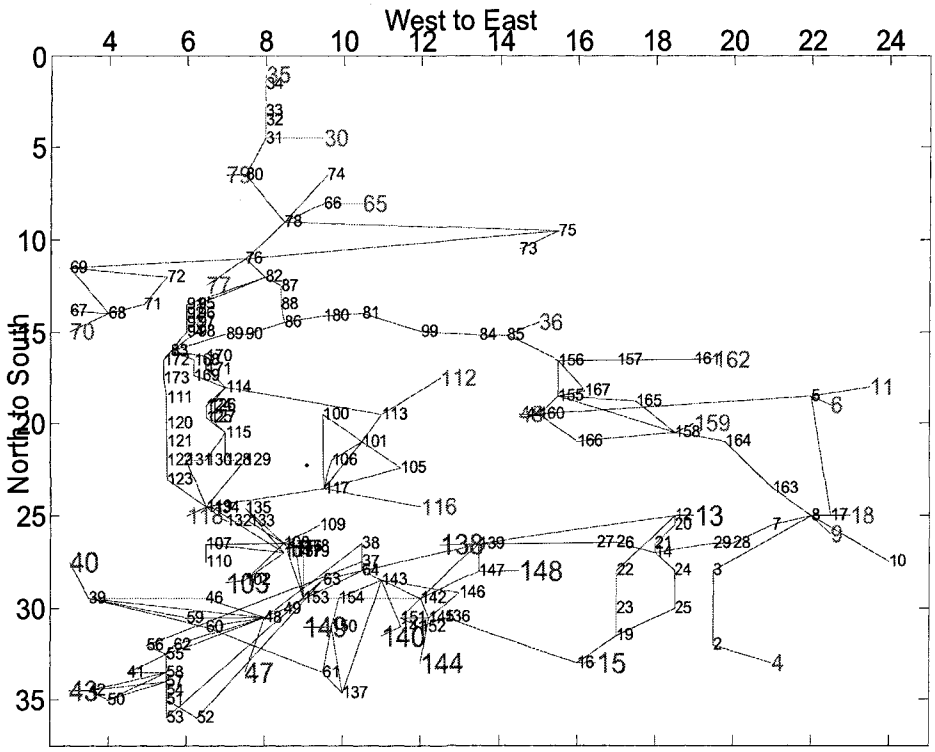


Figure 2-18 Graph of WECC 29-179 System

To demonstrate that controlled system islanding is almost independent of disturbance locations, two scenarios with different disturbances have been examined to illustrate the procedure of this approach.

Scenario 1

As indicated in Figure 2-8 and Figure 2-9, our approach starts with network reduction,

which can be divided into first-stage reduction and second-stage reduction.

After first-stage reduction, the WECC system has been reduced from 189 vertices and 222 edges to 132 vertices and 175 edges.

Based on the assumptions given earlier, the set of source vertices S and the set of sink vertices T should both be connected. To achieve this, other buses are included with the minimum spanning tree technique to make the set of generator buses in the south island and the set of generator buses in the rest of the area both connected. Then the network is reduced to a 21-vertex graph after applying vertices contraction, shown below.

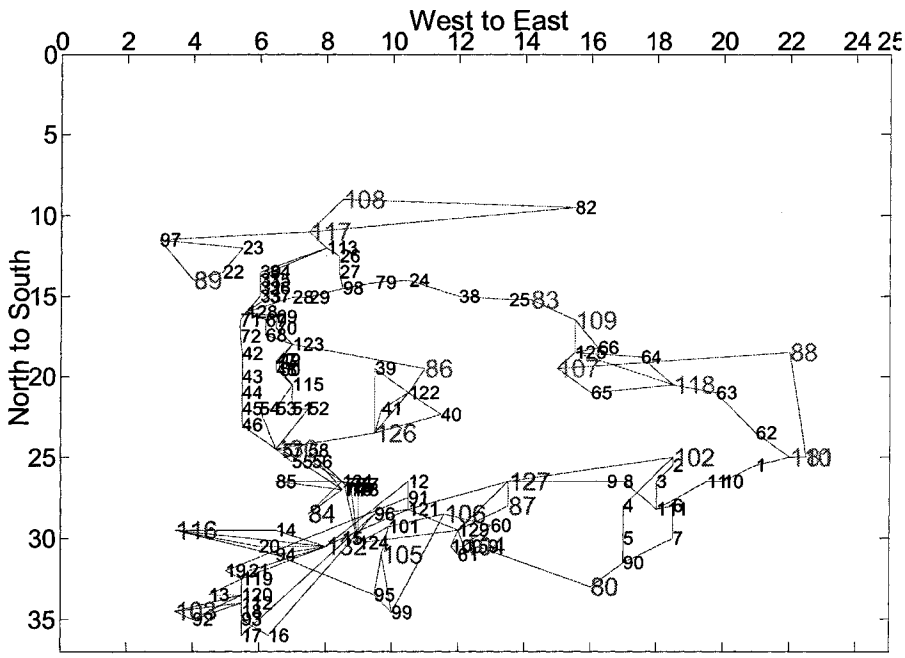


Figure 2-19 Graph of WECC 29-179 System after first stage reduction

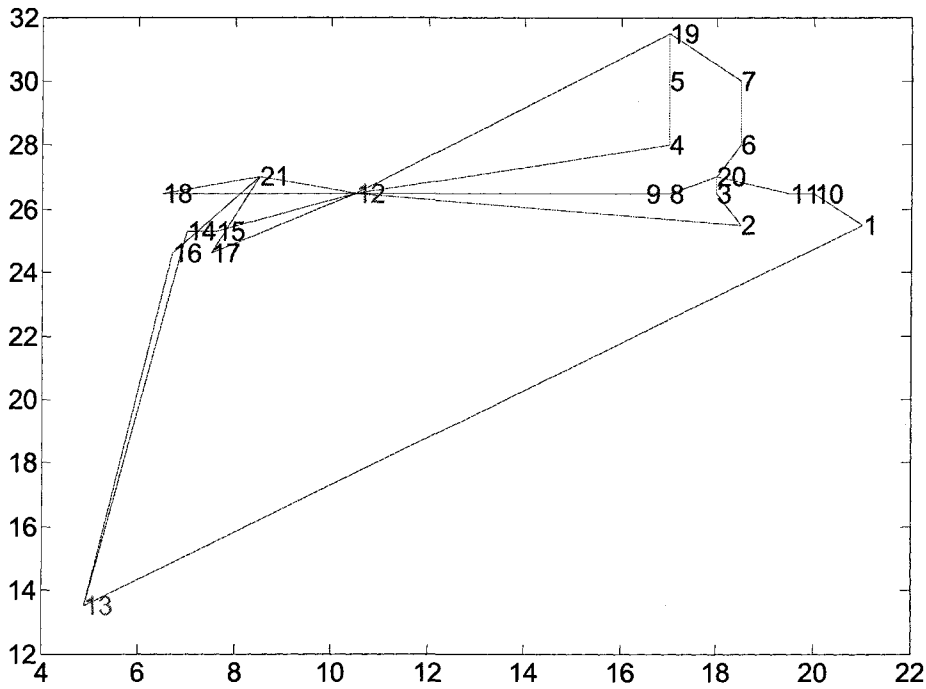


Figure 2-20 network representation after vertices contraction

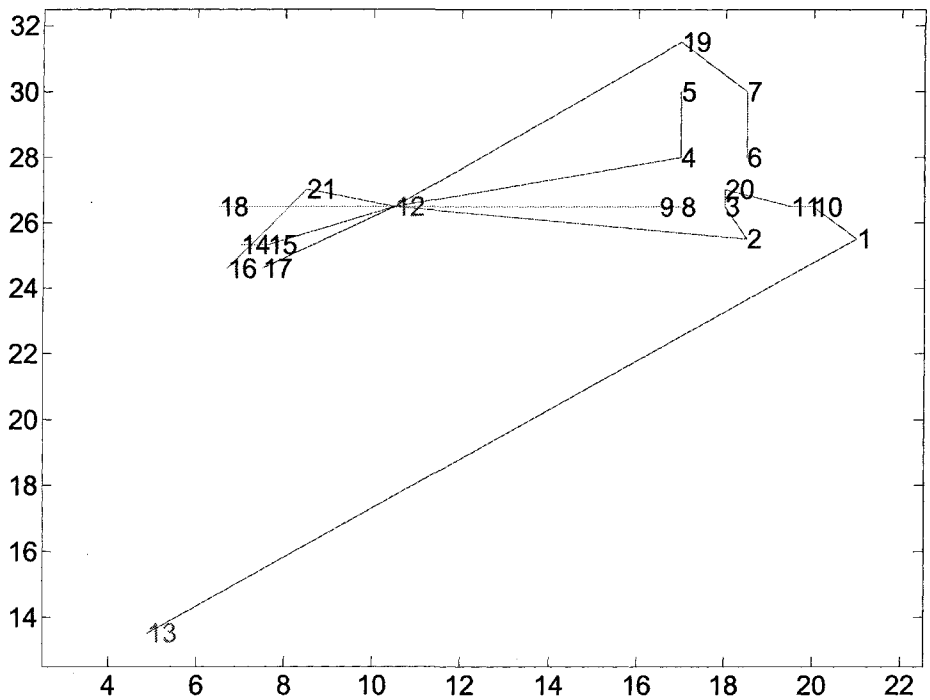


Figure 2-21 modified BFS tree

In Figure 2-20, vertex 12 is the source vertex, which is the aggregated vertex of the extensive generator buses in the south island, and vertex 13 is the sink vertex, which is the aggregated vertex of the extensive generator buses in the rest of the network. During the vertices contraction, other buses are included to make the set of generator buses in the south island and the set of generator buses in the rest of the network both connected.

Starting with the source vertex 12, the modified BFS tree is obtained as shown in Figure 2-21.

A recursive function with BFS tree flag-based DFS searching technique returns the following choices of 24 minimal cutsets with 3 lines, 210 cutsets with 5 lines, 162 cutsets with 6 lines, 324 cutsets with 7 lines, and 324 cutsets with 8 lines. TABLE 2-2 summarizes the minimal cutsets with different numbers of lines and minimal load-generation imbalance.

TABLE 2-2 MINIMAL CUSETS WITH DIFFERENT NUMBER OF LINES REMOVED

No. of lines removed	3	5	6	7	8
Cutsets number	24	210	162	324	324
Minimal Cutset with Minimal active power imbalance	14 29 104 134 108 133	102 104 14 29 108 133 108 135 108 107	16 19 12 20 12 22 104 134 139 27 108 133	102 104 19 25 12 20 139 27 108 133 108 135 108 107	16 19 102 104 12 20 12 22 139 27 108 133 108 135 108 107
Net Flow (MW)	-2076.35	-1464.98	-1434. 17	-1442. 28	-822.80

Figure 2-22 shows the relationship between the number of lines removed and load generation imbalance within the island. It is very clear that there is a trade-off; with more lines removed, there is less imbalance.

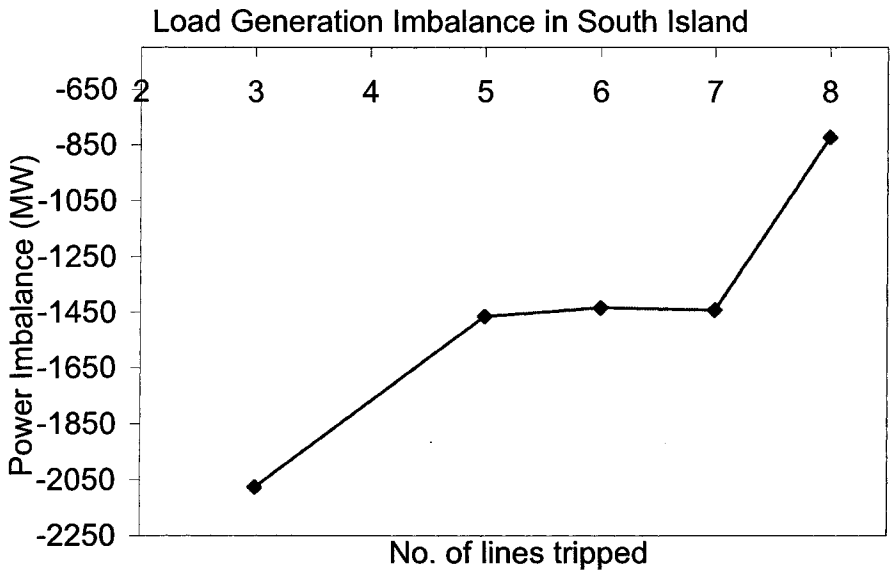


Figure 2-22 Relationship number of line removed v.s. active power imbalance

The situation is a bit more complicated when a major contingency is taken into account. A large contingency has been applied to the WECC system with characteristics such that transmission lines 83-168, 83-170, 83-172 are disconnected at the same time. This actually cuts the WECC system in the East. According to the method described in Section 2.5, in order to handle a system with more than two islands, either a Tuning Trial-Error or an Aggregated Island approach may be used to form the island in a systematic manner. In this case, the Aggregated Island approach is applied to island the system into two subsystems (one load-rich, the other generation-rich), along with the contingency. Once this is done, a Trial-Error approach is conducted in the aggregated load-rich island to form two islands.

TABLE 2-3 provides detailed information about the load-rich island after the Aggregated Island approach has been applied.

TABLE 2-3 AGGREGATED LOAD RICH ISLAND

	Generator (Bus No.)	Cutset (Bus No.)	Inertia (S)	Net Flow (MW)	TI (MW/S)
Aggregated Load Rich Island	15, 103, 148, 13, 43, 144, 149,140, 40, 138, 47, 112, 116, 118	168 83 170 83 172 83 14 29	1310.05	-4106.71	-3.1348

Two islands have been created by applying the Tuning Trial-Error approach to the aggregated island. TABLE 2-4 illustrates the detailed information of these two islands. The last column intuitively gives the idea of how fast the average rotor angle of generators in this island will move once the island is actually formed. It is ideally expected that the TI values for island 1 and 2 will be the same. However, depending on the topology of the real situation, these values are most likely not the same, although they are as close as possible. Figure 2-23 shows the final minimal cutset used to island the system.

TABLE 2-4 DETAILED INFORMATION FOR THE TWO SOUTH ISLANDS IN SCENARIO I

	Generator (Bus No.)	Cutset (Bus No.)	Inertia (S)	Net Flow (MW)	TI (MW/S)
Island 1	15, 103, 148, 13, 43, 144, 149,140, 40, 138, 47	**132 119** **134 119** 14 29	966.66	-2084.46	-2.1563
Island 2	112, 116, 118	168 83 170 83 172 83 **119 132** **119 134**	343.39	-2022.24	-5.8891

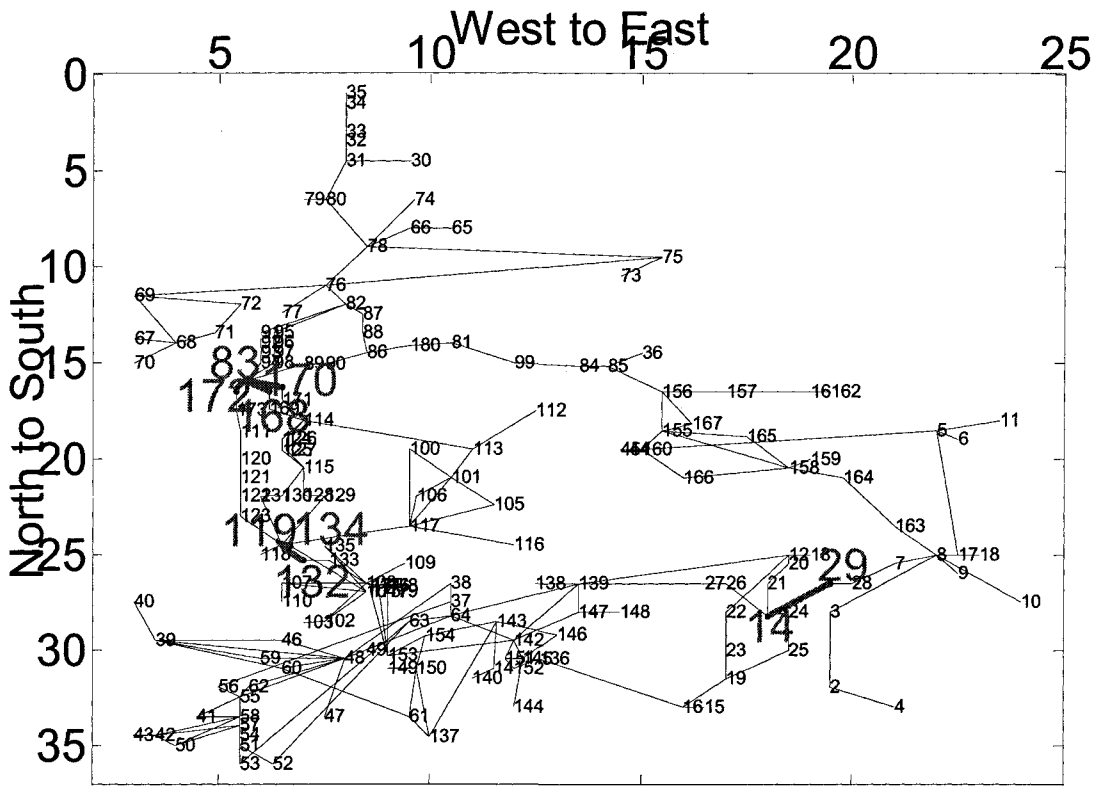


Figure 2-23 Final minimal cutset to island the system in Scenario I

Scenario 2

In this scenario, a contingency has been applied such that lines 139-27, 139-12, and 136-16 (dct) have been disconnected. This leads to the disconnection of the southern area from the east. The contingency is severe enough to make the system unstable if no self-healing strategies have been initiated.

The recursive function with BFS tree flag based DFS searching technique returns very similar cutest outcomes as shown in scenario 1. However, since the nominal south island has been split into two parts due to the contingency, the optimal cutest shown in TABLE 2-4 in scenario 1 may not be applicable any longer.

As stated in scenario I, all cusets candidates have been obtained from the recursive function before applying the contingency. With 3 lines tripped, the minimal cutset with

minimum net flow will be 14-29, 104-134, and 108-133 as shown in TABLE 2-4. However, after the contingency has occurred, the final cutset is line 132-119, 134-119, 136-16, 139-12, and 139-27 as shown in TABLE 2-5. Figure 2-24 shows the final cutset to island the system. One may find that the final cutset in scenario II is very similar to that in scenario I. This is due to the special network topology of the southern WECC system. This part of the system connects the rest of the system only through two independent paths: east path starting with bus 24, and west path near bus 108. Therefore, it becomes totally independent in searching minimal cutsets in east path and west path. That is the reason that part of the final cutset: line132-119 and line 119-134 does not change in both scenarios.

TABLE 2-5 DETAILED INFORMATION FOR THE SOUTH ISLAND IN SCENARIO II

	Generator (Bus No.)	Cutset (Bus No.)	Inertia (S)	Net Flow (MW)	TI (MW/S)
Island	103 148 43 144 149 140 40 138 47	**132 119** **134 119** 136 16 139 12 139 27	720.158	-4632.34	-6.4324

One may notice from Figure 2-24 that no line is tripped to form an island with generator 13 and 15; instead, these two generators connect to the rest of the system in spite of the fact that a weak connection exists, the connection to other generators, such as generator 11, is still relative tight (coherency index between generator 11 and 13 is 0.86 and coherency index between generator 11 and 13 is 0.74), as shown in TABLE 3-9. Also, the proposed island with generator 13 and 15 is in reality too small in scale to form an island.

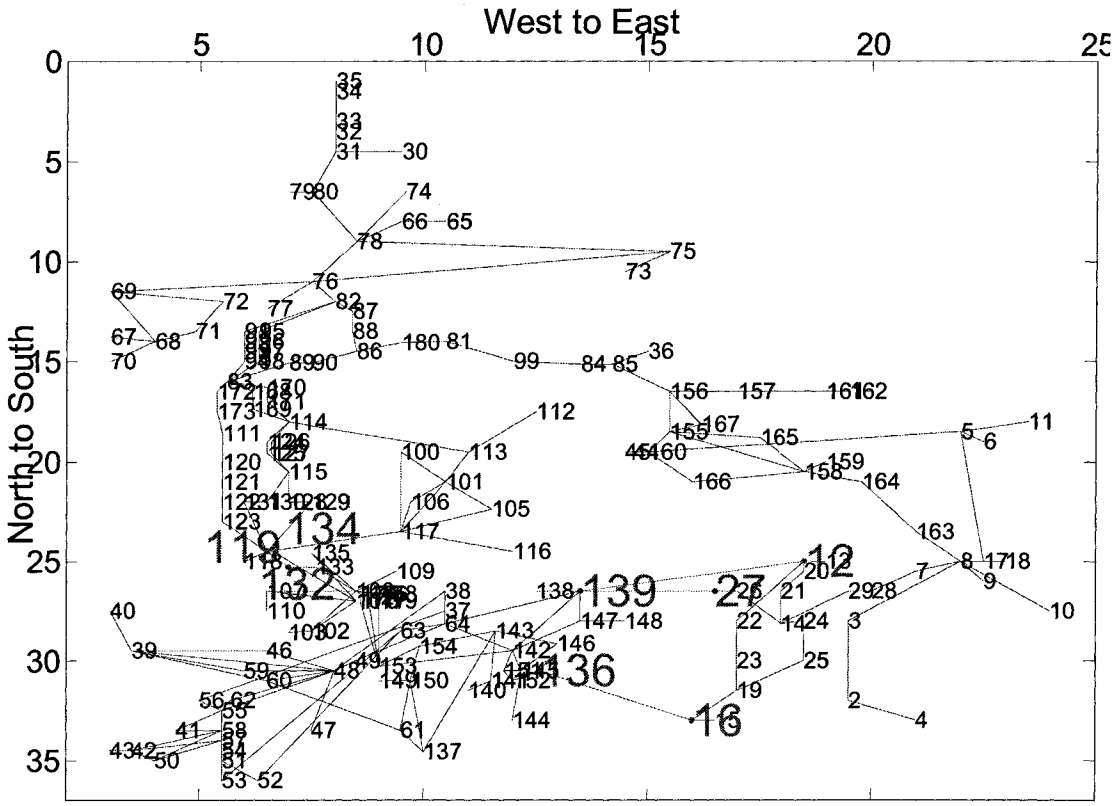


Figure 2-24 Final minimal cutset to island the system in Scenario II

2.9 Transient Simulation

To verify the advantages of the new islanding approach, it is necessary to conduct time-domain simulation and to investigate the system's transient performance after islanding. Two scenarios for accomplishing this task have been provided by to the grouping results in section 2.8.

Scenario 1

As a result of a severe fault, three 500KV transmission lines (83-168, 83-170, 83-172) are tripped at time 0 s, and the path from north to south along the east thus has been disconnected.

Four cases have been studied:

1. No self-healing strategy;
2. At time 0.087 s, form islands by tripping line 132-119, 134-119, and 14-29, without any load-shedding scheme installed;
3. At time 0.087 s, form islands by tripping line 132-119, 134-119, and 14-29 with conventional load-shedding scheme installed;
4. At time 0.087 s, form islands by tripping line 132-119, 134-119, and 14-29 with the new adaptive load-shedding scheme installed.

In cases 2, 3, and 4, two islands have been formed to prevent the cascading event addressed in section 0.

In Island1, there are 12 generators: 104, 149, 44, 145, 150, 141, 41, 139, 48, 113, 117 and 119. The total system inertia is 966.66s and total real power generation is 15477.7 MW. Therefore, the value of M_0 can be obtained as follows

$$M_{0,1} = 60 \times \frac{0.3P_{sys}}{2 \times \sum H} = 60 \times \frac{0.3 \times 154.477}{2 \times 966.66} = -1.441(Hz/s)$$

Similarly, in Island2, there are 3 generators: 112, 116 and 118. The total system inertia is 343.39s and total real power generation is 5118 MW. Therefore, the value of M_0 can be obtained as follows

$$M_{0,2} = 60 \times \frac{0.3P_{sys}}{2 \times \sum H} = 60 \times \frac{0.3 \times 51.18}{2 \times 343.39} = -1.341(Hz/s)$$

The new load-shedding scheme is developed as shown in TABLE 2-6. When the fault occurs, the rate of frequency decline at each bus is calculated and compared with the value from (2.27). If the rate of frequency decline at each bus is increased, the new load shedding scheme will be activated, shown as the second row in TABLE 2-6, in which 25 percent of the total load is shed with zero cycle delay in the first step. The character C in the table denotes cycle. Otherwise, the conventional load-shedding scheme will be activated, as shown in the

last row. The result from system transient simulation after applying contingency and islanding techniques indicates that the rate of frequency decline in the south island does not exceed the threshold value M_0. Therefore, the conventional load-shedding scheme has been applied at each bus in the south island. However, a large load deficit in the central island results in the application of the new load-shedding scheme.

TABLE 2-6 STEP SIZE OF THE NEW LOAD SHEDDING SCHEME

	59.5Hz	59.3Hz	58.8Hz	58.6Hz	58.3Hz
$M_i>M_0$	25% (0C)	5% (6C)	5% (6C)	4% (12C)	4% (18C)
$M_i<M_0$		15% (28C)	25% (18C)		

For the purpose of comparison with our new islanding scheme, islanding based on practical experience has been also studied, such that, after a fault at time 0 s, four lines are tripped to form the islands as follows: 139-12, 139-27, 136-16(dct). Simulation shows that the new islanding using both the conventional and the new load shedding scheme has the advantage of shedding fewer loads than that from islanding based on practical experience. Furthermore, there is less frequency oscillation detected at Generator 118 when new islanding is applied, compared to islanding based on practical experience as shown in Figure 2-25 and Figure 2-26.

TABLE 2-7 shows that new islanding method with new load-shedding scheme has the advantage of shedding fewer loads when compared with conventional load-shedding scheme, which indicates new load-shedding scheme indeed captures the frequency drop and sheds the loads ahead of the time based on the rate of the change of the frequency decline.

TABLE 2-7 COMPARISON OF NEW ISLANDING WITH TWO LOAD SHEDDING SCHEME

	Generation Load Imbalance (MW)	Inertia (S)	New Islanding with Load Shedding Scheme (MW)	
			Conventional	New
Island 1	**Generation:** 15477.70 **Load:** 17373.60	966.66	2220.84 (12.8%)	2220.84 (12.8%)
Island 2	**Generation:** 5118 **Load:** 7005.9	343.39	2439.51 (34.8%)	2081.19 (29.7%)

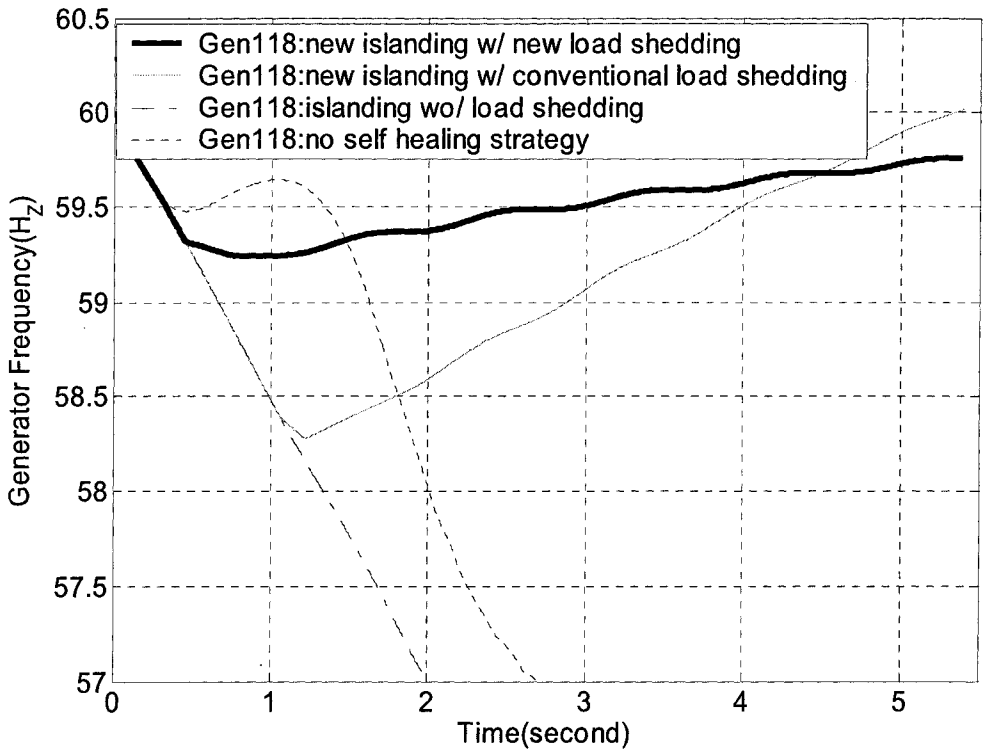


Figure 2-25 Generator frequency under different scenarios at Generator 118

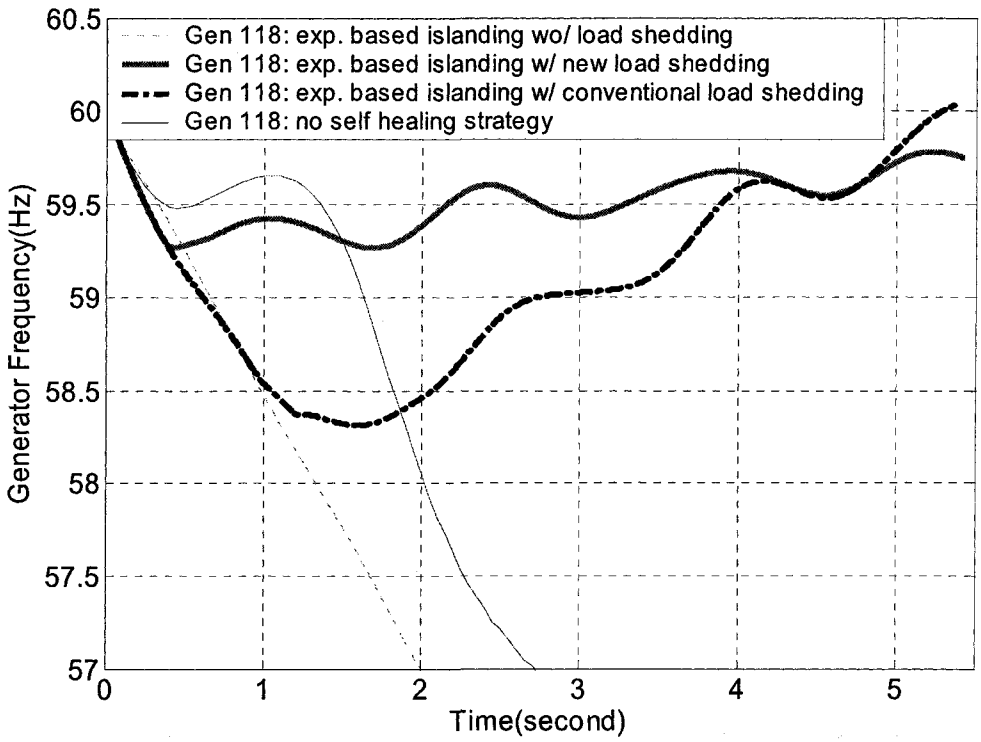


Figure 2-26 Generator frequency under different scenarios at Generator 118

Scenario 2

In this scenario, three 500KV transmission lines (12-139, 27-139, 16-136 dct) are tripped at time 0 s, and the path from north to south along east thus has been disconnected.

Four scenarios have been studied:

1. No self healing strategy;
2. At time 0.2 s, form the islands but without any load-shedding scheme installed;
3. At time 0.2 s, form the island with the conventional load-shedding scheme installed;
4. At time 0.2 s, form the island with the new adaptive load-shedding scheme installed

The final island in southern California has a total load of 15673.2 MW, and total generation of 11147.7MW. Therefore, the real power imbalance is 4624.23MW. The total inertia is 720.158 s. The value of M0 can be obtained as follows,

$$M_0 = 60 \times \frac{0.3 P_{sys}}{2 \times \sum H} = 60 \times \frac{0.3 \times 111.477}{2 \times 720.158} = 1.3932 (Hz/s)$$

TABLE 2-8 STEP SIZE OF THE NEW LOAD-SHEDDING SCHEME

	59.5Hz	59.3Hz	58.8Hz	58.6Hz	58.3Hz
$M_i>M_0$	24% (0C)	5% (6C)	5% (6C)	4% (12C)	4% (18C)
$M_i<M_0$		5% (28C)	24% (18C)		

TABLE 2-9 COMPARISON OF NEW ISLANDING WITH TWO LOAD-SHEDDING SCHEME

	Generation Load Imbalance (MW)	Inertia (S)	New Islanding with Load Shedding Scheme (MW)	
			Conventional	New
South Island	**Generation:** 11147.7 **Load:** 15673.2	720.158	5572.22 (35.6%)	4792.92 (30.6%)

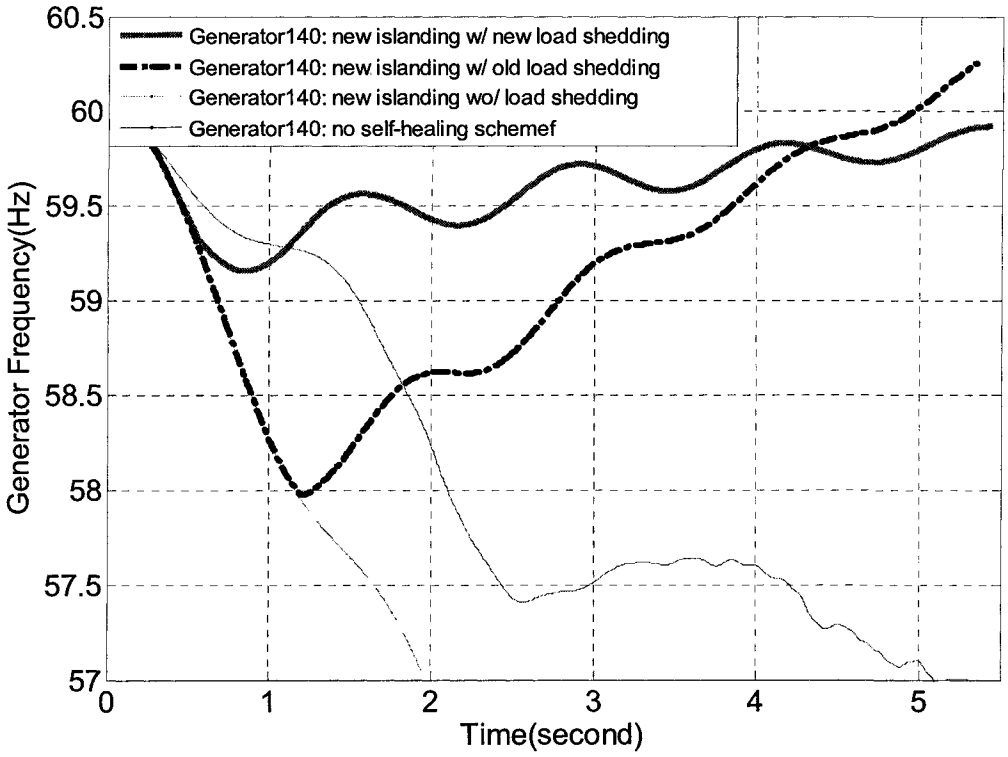


Figure 2-27 Generator frequency under different scenarios at Generator 140

2.10 Summary

In this chapter, an automatic power system islanding program has been described, and detailed descriptions of its motivation, functionality, and drawbacks have been addressed. It can be seen that this approach relies heavily on system reduction; since this BFS flag-based DFS searching technique requires significant computational effort for a large-scale system. It is therefore important to reduce the system scale by utilizing system topology information and system dynamic characteristics. To reduce system scale prior to using the minimal cutsets approach, one possible approach would be to eliminate each bus which is neither a generator bus nor a load bus, because those buses will not come into consideration under every situation. The system can thus be reduced to an equivalent system but with lower scale. The minimal-cutsets based islanding approach can then be applied to this reduced system, and the optimal cutset can be found. We may then map this optimal cutset back into the original system.

The final result of islanding may be affected by contingencies, especially when a 3-phase short-circuit fault occurs in the system. Sometimes such a fault shorts main transmission lines, which leads in turn to relay or breaker action and system isolation may not be avoidable. Some kinds of faults will also cut off the generator groups originally produced by the slow-coherency method. In either case, more islands will be produced as a result.

PSAPAC/DYNRED can be used to decide generator grouping based on the slow-coherency technique. Without consideration of contingencies, the number of groups may be specified by users. After the islanding program is executed, the same number of islands is generated, with each island including one generator group. However, when contingencies are taken into account, more islands will most likely make the original islanding less optimal. Therefore, the islanding program should be re-run.

The location of the contingency is very important for islanding. It may be located in one island, or at the boundary between two islands, or it may break one island into two.

Another issue requiring attention is the determination of line trips that may cause island separation. It is possible that only a subset of lines tripped may contribute to island separation (whether the contingency cut is minimal). Also, the contingency may influence more than two aggregated islands. For simplicity, the location of contingency is assumed to be limited within one island in our approach.

Various studies indicate that slow coherency may be affected by a change in system topology, which could also be due to the contingency. Slow coherency does not promise consistency if system topology is changed. If the change in system topology happens at a weak connection, however, slow coherency will not be affected. If the topology change occurs at a strong connection, slow coherency will be changed, and grouping may need to be be re-run.

A more critical question would be: Doesn't the coherency between generators vary with different load conditions? [22],[23] presents a new algorithm to compute the coherency index by using a matrix of eigenvectors corresponding to the small eigenvalues (slow modes) and row vectors associated with the generator rotor angles. Those slow modes change their time constants and contribution to system state with respect to the change of the system operating point. It would not be surprising if the coherency index indeed changes. To address this issue, one needs to investigate the correlation between the slow modes and different load condition, which leads us to Chapter 3.

3 GENERATOR COHERENCY INDICES TRACING USING THE CONTINUATION METHOD

3.1 Introduction

Slow coherency has been applied to group the generators before the islands are actually determined. The question may then be asked as to whether slow coherency is independent of disturbances or operating conditions. If slow coherency is independent of operating conditions, grouping information based on one particular operating point can be used for all other loading scenarios. Similarly, the same grouping information may be used under different contingencies if slow coherency is independent of disturbances. In [6], it has been pointed out that slow coherency among the groups of generators does not vary significantly with a change of initial condition or disturbance. A heuristic argument may be stated for investigating whether coherency is independent of initial conditions. The idea is to compute the correlation between contributions of slow modes on different generators. Slow modes affect the dynamics of system states through their eigenvalues and right eigenvectors. Therefore, the argument focuses on the effects on eigenvectors of slow modes due to initial conditions.

In linear control theory, a typical dynamic system can be represented by the following differential equation,

$$\begin{aligned} \dot{X} &= AX + Bu \\ X(0) &= \zeta \end{aligned} \tag{3.1}$$

where A is the system matrix.

Suppose $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ is the diagonal eigenvalues matrix of system matrix A, V is its relative right eigenvector matrix, and W is its relative left eigenvector matrix, which yields

$$AV = V\Lambda$$
$$WA = \Lambda W \tag{3.2}$$

Zero-Input state solution:

$$X(t) = \sum_{i=1}^{n} e^{\lambda_i t} v_i [w_i \zeta] \tag{3.3}$$

where v_i is ith column vector of V and w_i is i^{th} row vector of W.

Zero-State state solution:

$$X(t) = \sum_{i=1}^{n} e^{\lambda_i t} v_i [\sum_{k=1}^{m} (w_i b_k) \int_0^t e^{-\lambda_i \tau} u_k(\tau) d\tau] \tag{3.4}$$

where, v_i is i^{th} column vector of V, w_i is i^{th} row vector of W, and b_k the k^{th} column vector of B.

The scalar $w_i\zeta$ is considered to be the degree to which initial state ζ excites the i^{th} mode. The scalar $w_i b_k$ is considered to be the degree to which the k^{th} control, $u_k(t)$ influences the i^{th} mode.

Regarding each element in $X(t)$, from equation (3.1), the initial state ζ has the same impact on it through the inner product with W. This means that coherency is independent of the initial state ζ, which may vary according to the applied disturbance [22].

However, through various simulations and analysis, it has been found that generator grouping indeed changes with respect to large changes in system load conditions. This change in grouping is caused by generators having a loose coherency property. In this case some generators may switch from one grouping to another under certain system conditions. System modes may vary according to system operating conditions. Since the system matrix A varies with different operating conditions, eigenvalues λ_i and its relative right eigenvector v_i may also change, and as a result, the coherency between generators may change. Investigations in [6] indicate that grouping from slow coherency slightly changes with respect to different load conditions.

One thing that must be considered is that change in system topology may affect slow coherency. Slow coherency does not promise consistency if system topology is changed.

In the literature, the continuation method has been widely accepted and applied to compute system damping ratio margin, oscillatory stability margin, and voltage stability margin [34], [35], and [36]. Reference [36] presents an elegant approach to trace eigenvalues by involving a set of differential equations. The derivative in the differential equation denotes the eigenvalue and eigenvector elements' differentiation with respect to the system parameter. By integrating in the parameter domain, a curve of eigenvalue and eigenvector vs. parameter value can be traced.

3.2 Power System DAE Model

Generally speaking, power systems can be represented by a set of differential algebraic equations (DAE),

$$\begin{cases} \dot{X} = F(X, Y, \alpha) \\ 0 = G(X, Y, \alpha) \end{cases} \tag{3.5}$$

where the set of differential equations, F, represents the dynamics of system state variable X associated with the generators, the excitation systems, the prime movers and the speed governors (PSS model is not included in our approach without loss of generality). The set of algebraic equations, G, describes the relationship between those state variables through the network variables Y. In the power system, α represents the loading condition of the entire system.

3.2.1 Generator Model

A two-axis generator model as shown in [2] is used to represent the synchronous machines.

$$
\begin{aligned}
\dot{E}'_{qi} &= (E_{FDi} - E'_{qi} - (x_{di} - x'_{di})I_{di})/\tau'_{d0i} \\
\dot{E}'_{di} &= (-E'_{di} + (x_{qi} - x'_{qi})I_{qi})/\tau'_{q0i} \\
\dot{\omega}_i &= (-D_i(\omega_i - \omega_M) + P_{mi} - \\
&\qquad ((E'_{qi} - x'_{di}\, I_{di})I_{qi} + (E'_{di} + x'_{qi}\, I_{qi})I_{di}))/M_{gi} \\
\dot{\delta}_{iM} &= (\omega_i - \omega_M)\omega_R, i = 1,\ldots,M-1
\end{aligned}
\tag{3.6}
$$

where,

$$
\begin{aligned}
I_{di} &= [R_{si}E'_{di} + x'_{qi}\,E'_{qi} - R_{si}v_i \sin(\delta_{iM} - \theta_{iM}) \\
&\qquad - x'_{qi}\,v_i \cos(\delta_{iM} - \theta_{iM})]/A_i \\
I_{qi} &= [R_{si}E'_{qi} - x'_{di}\,E'_{di} - R_{si}v_i \cos(\delta_{iM} - \theta_{iM}) \\
&\qquad + x'_{di}\,v_i \sin(\delta_{iM} - \theta_{iM})]/A_i \\
A_i &= R^2{}_{si} + x'_{di}\,x'_{qi} \\
i &= 1,\ldots,M
\end{aligned}
$$

3.2.2 Excitation System Model

The simplified IEEE type DC1A [37] excitation system representation is used. This type of exciter has three states, represented as follows. Figure 3-1 shows the block diagram.

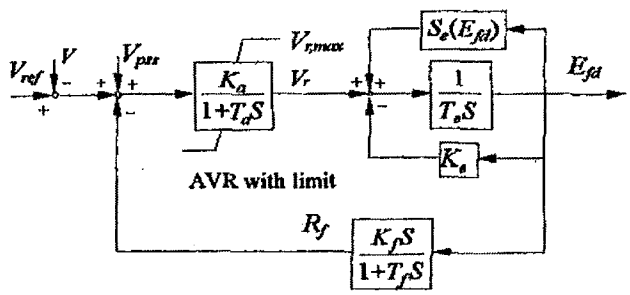


Figure 3-1 Block diagram of Type 1 Excitor (DC1A)

$$
\begin{aligned}
\dot{E}_{FDi} &= (V_{ri} - (S_{ei} + K_{ei})E_{FDi})/T_{ei} \\
\dot{V}_{ri} &= (-V_{ri} + K_{ai}(V_{Refi} - v_i - R_{fi}))/T_{ai} \\
\dot{R}_{fi} &= (-R_{fi} - (S_{ei} + K_{ei})K_{fi}E_{FDi}/T_{ei} + K_{fi}V_{ri}/T_{ei})/T_{fi} \\
&\qquad i = 1,\ldots,M
\end{aligned}
\tag{3.7}
$$

3.2.3 Governor System Model

A generic model [38] for a simplified prime mover and speed governor has been used in this approach. The mathematical representation of this governor is,

$$\begin{aligned} \dot{\mu}_i &= (P_{g0}(1+K_m\alpha)-(\omega_i-1)K_{1i}-\mu_i)/T_{gi} \\ \dot{P}_{mi} &= (\mu_i - P_{mi})/T_{chi} \qquad\qquad i=1,\ldots,M \end{aligned} \tag{3.8}$$

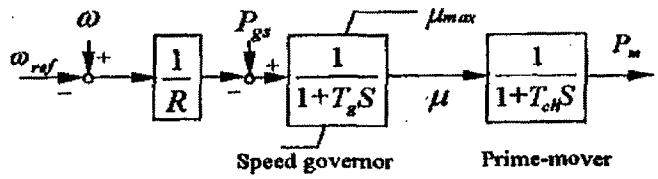


Figure 3-2 Block diagram of generic model for a simplified prime mover and speed governor

The description of the state variables is then given by,

$$X=[X_1, X_2 \cdots X_M]^T$$

where $X_i = (E'_{qi}, E'_{di}, E_{FDi}, X_{E1i}, X_{E2i}, \mu_i, P_{mi}, \omega_i, \delta_{iM})$. The notation δ_{iM} indicates that the Mth generator is chosen to be the reference machine, and δ_{iM} is the relative rotor angle of the ith generator. M is the total number of generators in the system. The size of X is 9M-1.

3.2.4 Load Model

A voltage and frequency dependent load model [38] is used. Parameters K_{IP} (K_{IQ}), K_{ZP} (K_{ZQ}), and K_{PP} (K_{PQ}) are the components of constant current, impedance, and power, respectively. Parameters K_{Pl} (K_{Ql}) indicate frequency dependent components,

$$\begin{cases} Pl_i = P_{0,i}\left[(\frac{V_i}{V_{0,i}})K_{IP,i}+(\frac{V_i}{V_{0,i}})^2 K_{ZP,i}+K_{PP,i}\right](1+K_{W,i}(\omega_M-1)(1+\alpha K_{Pl,i}) \\ Ql_i = Q_{0,i}\left[(\frac{V_i}{V_{0,i}})K_{IQ,i}+(\frac{V_i}{V_{0,i}})^2 K_{ZQ,i}+K_{PQ,i}\right](1+K_{W,i}(\omega_M-1)(1+\alpha K_{Ql,i}) \end{cases} \tag{3.9}$$

3.2.5 Network Representation

The network model is given by

$$\begin{cases} 0 = P_{Gi} - Pl_i - P_{ti} \\ 0 = P_{Gi} - Ql_i - Q_{ti} \end{cases} \tag{3.10}$$

where,

$$P_{Gi} = I_{di} v_i \sin(\delta_i - \theta_i) + I_{qi} v_i \cos(\delta_i - \theta_i)$$

$$Q_{Gi} = I_{di} v_i \cos(\delta_i - \theta_i) - I_{qi} v_i \sin(\delta_i - \theta_i)$$

$$i = 1, \ldots . M$$

$$P_{ti} = \sum_{j=1}^{N} y_{i,j} V_i V_j \cos(\theta_{jM} - \theta_{iM} + \gamma_{i,j})$$

$$Q_{ti} = -\sum_{j=1}^{N} y_{i,j} V_i V_j \sin(\theta_{jM} - \theta_{iM} + \gamma_{i,j})$$

$$i = 1, \ldots, N$$

The network variables Y can be defined as $Y = \{Yi\}$, where $Y_i = (V_i, \theta_{iM})$. If the total number of buses in the system is N, Y is a $2N$ vector.

3.3 Approach Formulation

3.3.1 Introduction to the Continuation Method

The concept of the continuation method has been documented and developed in mathematics literature for several decades [39]. However, it was much later when power system engineers began to realize the power and versatility of the continuation method [40], [38]. This method is ideally suited to solve the following engineering problems [39],

1. Given a mapping F: $R^n \rightarrow R^n$, find solutions to $F(x)=0$.
2. Given a mapping G: $R^n \rightarrow R$, find $\min_x G(x)$
3. Given a mapping F: $R^m \rightarrow R^n$, and a G: $R^m \rightarrow R$, find the $\min G(x)$ s.t. $F(x)=0$
4. Given F and G as in 3, and U as a subset of R^n, find $\min G(x)$ s.t $F(x) \in U$

The continuation method computes a portion of the solution manifold near one solution then selects another solution from this set and repeats the process. As long as the new portion covers some new part of the solution manifold the computation progresses. The basic computational steps are:

Compute the solution manifold near some point x_i, where a set of equations $F(x_i)$ is equal to 0.

Select a new point x_{i+1} in the neighborhood of x_i.

Check to see if this point leads to a new part of the solution manifold.

There are two approaches to addressing Step 1, and these approaches divide continuation methods into two types: simplicial (piece-wise linear) continuation methods and predictor-corrector methods. Predictor-corrector continuation is well-defined in one dimension. Experience shows that the predictor-corrector continuation method appears to be ideally suited for the case where there is only one deforming parameter so this technique is used below.

3.3.2 Single Mode Tracing

The Jacobian matrix of the system DAE model is,

$$A_{total} = \begin{bmatrix} F_X(\alpha) & F_Y(\alpha) \\ G_X(\alpha) & G_Y(\alpha) \end{bmatrix}$$

$$A_{sys} = F_X(\alpha) - F_Y(\alpha) G_Y^{-1}(\alpha) G_X(\alpha) \tag{3.11}$$

where, F_x, F_y, G_x, and G_y are the first derivatives of F and G with respect to X and Y.

The study of system dynamics, such as generator grouping, is often associated with the eigenspace of the matrix A_{sys}.

Suppose λ is one of the slow-modes-associated eigenvalues of A_{sys}, and v is its right eigenvector, resulting in the following equation,

$$A_{sys}(\alpha) v = \lambda v \tag{3.12}$$

Notice that there is an inverse matrix in A_{sys}. This is not acceptable not only because it is computationally time-consuming, but also because it is difficult to use in an explicit approach. Therefore, by introducing a companion eigenvector u [35], the inverse matrix may be eliminated from (3.11). The formulation of u may be given as,

$$u = -G_Y^{-1}(\alpha)G_X(\alpha)v \tag{3.13}$$

Equation (3.12) may extended to the following manner,

$$\begin{cases} F_X v + F_Y u = \lambda v \\ G_X v + G_Y u = 0 \end{cases} \tag{3.14}$$

It may be noticed that λ, v, and u, respectively, are defined in the complex space. It is convenient for computation if all variables are defined in real space.

$$\begin{cases} v = v_R + jv_I; \\ u = u_R + ju_I; \\ \lambda = \lambda_R + j\lambda_I; \end{cases} \text{ where } \begin{cases} v_R, v_I \in R^{7M-1} \\ u_R, u_I \in R^{2N} \\ \lambda_R, \lambda_I \in R^1 \end{cases}$$

Hence, (3.14) can be rewritten as follows:

$$\begin{cases} \lambda_R v_R - \lambda_I v_I - F_X v_R - F_Y u_R = 0 \\ \lambda_I v_R + \lambda_R v_I - F_X v_I - F_Y u_I = 0 \end{cases} \tag{3.15a}$$

$$\begin{cases} -G_X v_R - G_Y u_R = 0 \\ -G_X v_I - G_Y u_I = 0 \end{cases} \tag{3.15b}$$

From (3.5),

$$\begin{cases} F(X, Y, \alpha) = 0 \\ G(X, Y, \alpha) = 0 \end{cases} \tag{3.16}$$

The set of nonlinear equations obtained by combining (3.10a-b) and (3.11), generally does not possess a unique solution until three more constraints are added because we have 3*(9*M*+2*N*) variables but only 3*(9*M*+2*N*)-3 equations. One straightforward constraint on v will be, $v^* v=1$ (This constraint cannot make v unique. In fact, if v is the solution, $-v$ is (3.12) also the solution). The alternative may be to instead set $v^T v=1$, which can return two

equations if extended to real and imaginary parts.

$$\begin{cases} v_R{}^T v_R - v_I{}^T v_I - 1 = 0 \\ v_R{}^T v_I + v_I{}^T v_R = 0 \end{cases} \tag{3.17}$$

Therefore, one obtains a set of $3(9M+2N) - 1$ nonlinear equations, denoted as

$$H(Z,\alpha) = 0 \tag{3.18}$$

where Z is the set of DAE state variables X, and network variables Y, λ, v, and u: $Z = (v_R, v_I, u_R, u_I, \lambda_R, \lambda_I, X, Y)$. By applying the continuation method, we can possibly solve this slow-mode tracing problem.

The Jacobian matrix of (3.18) is needed. First of all, equation (3.16) can be differentiated with respect to x, y, and α. Note that only static transitions are considered, i.e., dynamics from one operating point to another will not be taken into consideration (these dynamics may be significant in some situations). Thus,

$$\begin{aligned} 0 &= F_X \Delta x + F_Y \Delta y + F_\alpha \Delta\alpha \\ 0 &= G_X \Delta x + G_Y \Delta y + G_\alpha \Delta\alpha \end{aligned} \tag{3.19}$$

From (3.15a),

$$\begin{aligned} 0 &= (\lambda_R I - F_X)\Delta v_R - \lambda_I \Delta v_I - F_Y \Delta u_R + v_R \Delta\lambda_R - v_I \Delta\lambda_I \\ &\quad + K_1 \Delta X + K_2 \Delta Y + K_3 \Delta\alpha \\ 0 &= \lambda_I \Delta v_R + (\lambda_R I - F_X)\Delta v_I - F_Y \Delta u_I + v_I \Delta\lambda_R + v_R \Delta\lambda_I \\ &\quad + K_4 \Delta X + K_5 \Delta Y + K_6 \Delta\alpha \end{aligned} \tag{3.20}$$

where,

$$\begin{aligned} K_1 &= F_{X,X} v_R + F_{X,Y} u_R, \quad K_4 = F_{X,X} v_I + F_{X,Y} u_I \\ K_2 &= F_{Y,X} v_R + F_{Y,Y} u_R, \quad K_5 = F_{Y,X} v_I + F_{Y,Y} u_I \\ K_3 &= F_{X,\alpha} v_R + F_{Y,\alpha} u_R, \quad K_6 = F_{X,\alpha} v_I + F_{Y,\alpha} u_I \end{aligned}$$

From Equation (3.15b),

$$\begin{aligned} 0 &= -G_X \Delta v_R - G_Y \Delta u_R + K_7 \Delta X + K_8 \Delta Y + K_9 \Delta\alpha \\ 0 &= -G_X \Delta v_I - G_Y \Delta u_I + K_{10} \Delta X + K_{11} \Delta Y + K_{12} \Delta\alpha \end{aligned} \tag{3.21}$$

where

$$K_7 = G_{X,X}v_R + G_{X,Y}u_R, \quad K_{10} = G_{X,X}v_I + G_{X,Y}u_I$$
$$K_8 = G_{Y,X}v_R + G_{Y,Y}u_R, \quad K_{11} = G_{Y,X}v_I + G_{Y,Y}u_I$$
$$K_9 = G_{X,\alpha}v_R + G_{Y,\alpha}u_R, \quad K_{12} = G_{X,\alpha}v_I + G_{Y,\alpha}u_I$$

From equation (3.17), we obtain the equality,

$$\begin{aligned} 0 &= v_R{}^T \Delta v_R - v_I{}^T \Delta v_I \\ 0 &= v_I{}^T \Delta v_R + v_R{}^T \Delta v_I \end{aligned} \tag{3.22}$$

It is not straightforward to compute the parameters K_i (i = 1,...,12) by simply taking partial derivatives. For example, the second term of K_1, is actually obtained by manipulating $F_Y u_R$, as shown below,

$$F_Y u_R(k) = \sum_{j=1}^{2N} F_Y(k,j) u_{R,j} \tag{3.23}$$

Therefore,

$$\begin{aligned} F_{Y,X}\Delta X u_R(k) &= \sum_{j=1}^{2N}\sum_{i=1}^{9M-1} \frac{\partial F_Y(k,j)}{\partial x_i}\Delta x_i u_{R,j} \\ &= \sum_{j=1}^{2N}\sum_{i=1}^{9M-1} \frac{\partial F(k)}{\partial y_j \partial x_i}\Delta x_i u_{R,j} \\ &= \sum_{j=1}^{2N}\sum_{i=1}^{9M-1} \frac{\partial F_X(k,i)}{\partial y_j}\Delta x_i u_{R,j} \\ &= \sum_{i=1}^{9M-1}\sum_{j=1}^{2N} \frac{\partial F_X(k,i)}{\partial y_j} u_{R,j}\Delta x_i \\ &= F_{X,Y} u_R \Delta X(k) \end{aligned} \tag{3.24}$$

where $k = 1,...,9M-1$

Other parameters can be easily obtained by following a similar procedure.

As mentioned before, the continuation method is composed of two consecutive steps, illustrated by the following.

Predictor

The predictor in continuation methods is shown in (3.25).

$$[J_{AUG}]\Delta U = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.25}$$

where,

$$J_{AUG} = \begin{bmatrix}
\lambda_R I - F_X & -\lambda_I I & -F_Y & 0 & v_R & -v_I \\
\lambda_I I & \lambda_R I - F_X & 0 & -F_Y & v_I & v_R \\
-G_X & 0 & -G_Y & 0 & 0 & 0 \\
0 & -G_X & 0 & -G_Y & 0 & 0 \\
v^T{}_R & -v^T{}_I & 0 & 0 & 0 & 0 \\
v^T{}_I & v^T{}_R & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & e^T{}_k & 0 & 0
\end{bmatrix}$$

$$\begin{bmatrix}
-(\frac{\partial F_X}{\partial X} v_R + \frac{\partial F_X}{\partial Y} u_R) & -(\frac{\partial F_Y}{\partial X} v_R + \frac{\partial F_Y}{\partial Y} u_R) & -(\frac{\partial F_X}{\partial \alpha} v_R + \frac{\partial F_Y}{\partial \alpha} u_R) \\
-(\frac{\partial F_X}{\partial X} v_I + \frac{\partial F_X}{\partial Y} u_I) & -(\frac{\partial F_Y}{\partial X} v_I + \frac{\partial F_Y}{\partial Y} u_I) & -(\frac{\partial F_X}{\partial \alpha} v_I + \frac{\partial F_Y}{\partial \alpha} u_I) \\
-(\frac{\partial G_X}{\partial X} v_R + \frac{\partial G_X}{\partial Y} u_R) & -(\frac{\partial G_Y}{\partial X} v_R + \frac{\partial G_Y}{\partial Y} u_R) & -(\frac{\partial G_X}{\partial \alpha} v_R + \frac{\partial G_Y}{\partial \alpha} u_R) \\
-(\frac{\partial G_X}{\partial X} v_I + \frac{\partial G_X}{\partial Y} u_I) & -(\frac{\partial G_Y}{\partial X} v_I + \frac{\partial G_Y}{\partial Y} u_I) & -(\frac{\partial G_X}{\partial \alpha} v_I + \frac{\partial G_Y}{\partial \alpha} u_I) \\
0 & 0 & 0 \\
0 & 0 & 0 \\
F_X & F_Y & F_\alpha \\
G_X & G_Y & G_\alpha \\
0 & 0 & 0
\end{bmatrix} \tag{3.26}$$

The last row in J_{AUG} denotes that the k^{th} element of the augmented vector $U \equiv [Z, \alpha]^T$ is increased by 1. The number k is chosen such that J_{AUG} is nonsingular. Usually, the element with the largest change in the last iteration will be chosen. The element $e^T{}_k$ is a 3(9M+2N)-dimensional row vector with all elements equal to zero except the k^{th} element, which is unity.

The last row in J_{AUG} shows that the k^{th} element of the augment vector $U \equiv [Z, \alpha]^T$ is

increased by 1. The number k is chosen such that J_{AUG} is nonsingular. Usually, the element with thelargest change in the last iteration will be chosen.

Corrector

Basically, the corrector in the continuation method is given by.

$$G(U) \equiv \begin{cases} H(Z,\alpha) = 0 \\ e_k^T U - \eta = 0 \end{cases} \tag{3.27}$$

Note that η *is* the value of k^{th} element of the augment vector U after the predictor procedure. In the corrector step the value of k^{th} element of U is fixed. The formulation of the conventional Newton-Raphson method is shown in the following equation,

$$[J_{AUG}]\Delta U = -\begin{bmatrix} H(Z,\alpha) \\ 0 \end{bmatrix} \tag{3.28}$$

Instead of the Newton-Raphson method, a globally convergent method for nonlinear systems of equations based on a quasi-Newton approach described in Section 9.7 of [41] is used here for obtaining the solution of a nonlinear system of algebraic equations. From the literature, the Newton-Raphson method can provide quadratically-convergent speed when the initial guess is near a root. However, the disadvantages of the general Newton-Raphson method are: 1) It is important that the initial guess be near the root; and 2) it is sometimes very difficult to choose the step length at each iteration. The second issue becomes very significant especially when the objective functions are a set of linear equations.

There are efficient general techniques for finding a minimum of functions with multiple variables. Under certain circumstance, the problem can be solved by finding the minimum of a function, instead of finding roots of a set of nonlinear functions. In this approach, a super function f is defined with positive definite (equivalent to identity matrix), such that,

$$f = 0.5 \times \|G\|_2^2 \tag{3.29}$$

At the root of G, f reaches its minimum, which is zero. However it may not be inversely true. When f attains its minimum, G may not be zero, which means that there is no root in the

neighborhood. Therefore, by minimizing f, root finding for G can be achieved. A special routine is created in our approach to return a flag indicating whether a minimum indicates the root, or instead whether another initial guess may be chosen.

The basic idea for the corrector is illustrated as follows: since the Newton-Raphson method always returns the Newton-Raphson step $\delta U=(-J_{AUG}^{-1}*G)$ indicating the descent direction for G, it is not difficult to show that the Newton-Raphson step also denotes the descent direction for f. Therefore, the strategy is, along the direction of the Newton-Raphson step, to choose an appropriate μ $(0<\mu\leq 1)$, such that,

$$f(U_{\text{old}} + \mu \times \delta U) \tag{3.30}$$

has diminished to an acceptable value.

3.3.3 Multiple modes tracing problem

Slow-coherency-based generator grouping involves certain slow eigenvalues. It is pertinent to ask whether it is possible to trace multiple slow modes in one single approach. Equation (3.14) sheds some lighs on the extension of the approach to the multiple modes tracing problem. Suppose in the system, there are r slow modes. Then for the set of eigenvalues $\Lambda = diag\{\lambda_1,...,\lambda_r\}$, the set of eigenvectors is $V=[V_1,...,V_r]^T$, and the set of companion eigenvectors is $U=[U_1,...,U_r]^T$.

According to this multiple modes tracing problem, (3.14) is extended to the following:

$$\begin{cases} F_X V + F_Y U = V\Lambda \\ G_X V + G_Y U = 0 \end{cases} \tag{3.31}$$

By substituting

$$\begin{cases} V = V_R + jV_I; \\ U = U_R + jU_I; \\ \Lambda = \Lambda_R + j\Lambda_I; \end{cases} \text{ where } \begin{cases} V_R, V_I \in R^{(9M-1)\times r} \\ U_R, U_I \in R^{2N\times r} \\ \Lambda_R, \Lambda_I \in R^{1\times r} \end{cases}$$

Equation (3.31) can be rewritten as

$$\begin{cases} V_R\Lambda_R - V_I\Lambda_I - F_X V_R - F_Y U_R = 0 \\ V_R\Lambda_I + V_I\Lambda_R - F_X V_I - F_Y U_I = 0 \\ -G_X V_R - G_Y U_R = 0 \\ -G_X V_I - G_Y U_I = 0 \end{cases} \tag{3.32}$$

As a result, along with

$$\begin{cases} F(X,Y,\alpha) = 0 \\ G(X,Y,\alpha) = 0 \end{cases} \tag{3.33}$$

and the constraint on V, where $V^TV=I$

$$\begin{cases} V_R{}^T V_R - V_I{}^T V_I - I = 0 \\ V_R{}^T V_I + V_I{}^T V_R = 0 \end{cases} \tag{3.34}$$

One obtains a set of nonlinear equations, denoted as

$$H(Z,a) = 0 \tag{3.35}$$

where Z is the set of DAE state variables X and network variables Y, and Λ, V, and U (Λ, V, and U now become matrices): $Z = [V_R, V_I, U_R, U_I, L_R, L_I, X, Y]^T$.

3.3.4 Generator Coherency Indices Tracing

Once the system DAE has been obtained, the continuation method can be applied to investigate the relationship between Z (refer to (3.18)) and system loading condition α. Also, the relationship between right eigenvectors V_i and α will be used to trace the generator coherency indices. Here, V_i is the right eigenvectors of the slow modes being selected.

Generator coherency indices are defined as [22], [23]

$$d_{ij} = \frac{W_i W_j^T}{\|W_i\|\|W_j\|} \tag{3.36}$$

where W_i and W_j are obtained from the right eigenvector transfer matrix $V=\{V_i\}$, by considering only the columns associated with the selected slow modes. W_i and W_j are row vectors corresponding to the rotor angle of the i^{th} and j^{th} generators.

3.3.5 Generator Coherency Index Sensitivity

One of the advantages of this approach is that the sensitivity of the coherency indices can be obtained from the Predictor in (3.25) and (3.36). Therefore, a clear picture of not only the generator coherency indices under the current operating condition, but also its sensitivity to changes in load condition are available without performing power flow and small signal analysis at each operating condition. The sensitivity information tells the operator how fast the coherency indices, or, in other words, the generator grouping, will change according to the load changes. The analytical formulation of the coherency index sensitivity may be derived from (3.36).

3.4 Results

3.4.1 New England 10 Gen - 39 Bus system

Figure 3-3 illustrates the one-line diagram of the test system.

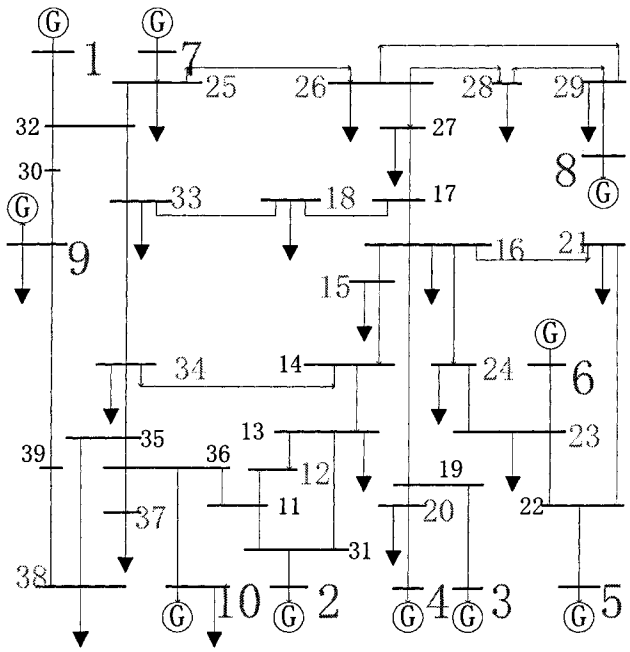


Figure 3-3 One line diagram of New England 10 generator 39 bus system

The load model used here in Scenarios I and II is a ZIP model with a combination of constant impedance (20%), constant current (30%), and constant power component (50%). A frequency component (with a frequency coefficient of 20%) has also been included.

Scenario I: The load scenario has been chosen such that the load at each load bus and generation at each generator bus uniformly decrease by a factor α from the base-case loading.

TABLE 3-1 shows all the slow modes for the base case which have the largest participation factors of the generator rotor angle and speed. It may be observed that, all modes except for Mode 7 (which participates in Generator 9 rotor speed the most) have very poor damping (around 1% ~ 5%). In this approach, the first four slow modes have been selected to calculate the coherency indices.

TABLE 3-1 SCENARIO I- SYSTEM MODES AT BASE CASE

Mode	Value	Frequency (Hz)	Damping (%)
7	-0.0566+0.1018i	0.016	-48.59
49	-0.1118+3.7587i	0.598	-2.97
56	-0.2697+6.0072i	0.956	-4.49
58	-0.1766+6.2450i	0.994	-2.83
61	-0.2968+6.8248i	1.086	-4.34
63	-0.1082+7.0908i	1.129	-1.53
65	-0.2066+7.5051i	1.194	-2.75
69	-0.3416+8.5007i	1.353	-4.02
72	-0.4019+8.7566i	1.394	-4.58
74	-0.4082+8.7647i	1.395	-4.65

The value of the generator coherency indices are shown in TABLE 3-2 for the base case. It can easily be concluded that Generators 2, 3, 4, 5, and 6 are coherent, and Generators 1, 7, and 8 are coherent. Generator 9 does not show tight coherency with other machines because the highest coherency index is 0.894, which is with Generator 1. Generator 10 is the reference machine. From TABLE 3-2, it is possible to generate TABLE 3-3, where the

coherency index among pairs of generators is listed in descending order.

Figure 3-4 illustrates how the coherency indices between any pair of generators change as the load factor α varies by taking the first four slow modes into consideration. Due to limited space, only selected loci of indices have been shown in the figure. The traces of the loci of coherency indices show that none of the selected indices change significantly in order to affect the grouping information.

TABLE 3-2 SCENARIO I- COHERENCY INDICES BETWEEN PAIRS OF GENERATORS AT BASE CASE

Index	G1	G2	G3	G4	G5	G6	G7	G8	G9
G1	X	0.83	0.88	0.81	0.87	0.88	***0.98***	***0.95***	0.89
G2	0.83	X	**0.96**	**0.94**	**0.98**	**0.96**	0.90	0.70	0.55
G3	0.88	0.96	X	**1.00**	**1.00**	**1.00**	0.90	0.59	0.70
G4	0.81	0.94	1.00	X	**0.99**	**0.99**	0.84	0.41	0.62
G5	0.87	0.98	1.00	0.99	X	**1.00**	0.91	0.66	0.64
G6	0.88	0.96	1.00	0.99	1.00	X	0.91	0.68	0.61
G7	0.98	0.90	0.90	0.84	0.91	0.91	X	***0.97***	0.81
G8	0.95	0.70	0.59	0.41	0.66	0.68	0.97	X	0.61
G9	0.89	0.55	0.70	0.62	0.64	0.61	0.81	0.61	X

TABLE 3-3 SCENARIO I- COHERENCY INDICES AT BASE CASE

Rank	Gen Pair	Rank	Gen Pair	Rank	Gen Pair
1	03-06	13	02-04	25	01-04
2	03-05	14	06-07	26	03-09
3	05-06	15	05-07	27	02-08
4	03-04	16	02-07	28	06-08
5	04-06	17	03-07	29	05-08
6	04-05	18	01-09	30	05-09
7	01-07	19	01-06	31	04-09
8	02-05	20	01-03	32	08-09
9	07-08	21	01-05	33	06-09
10	02-06	22	04-07	34	03-08
11	02-03	23	01-02	35	02-09
12	01-08	24	07-09	36	04-08

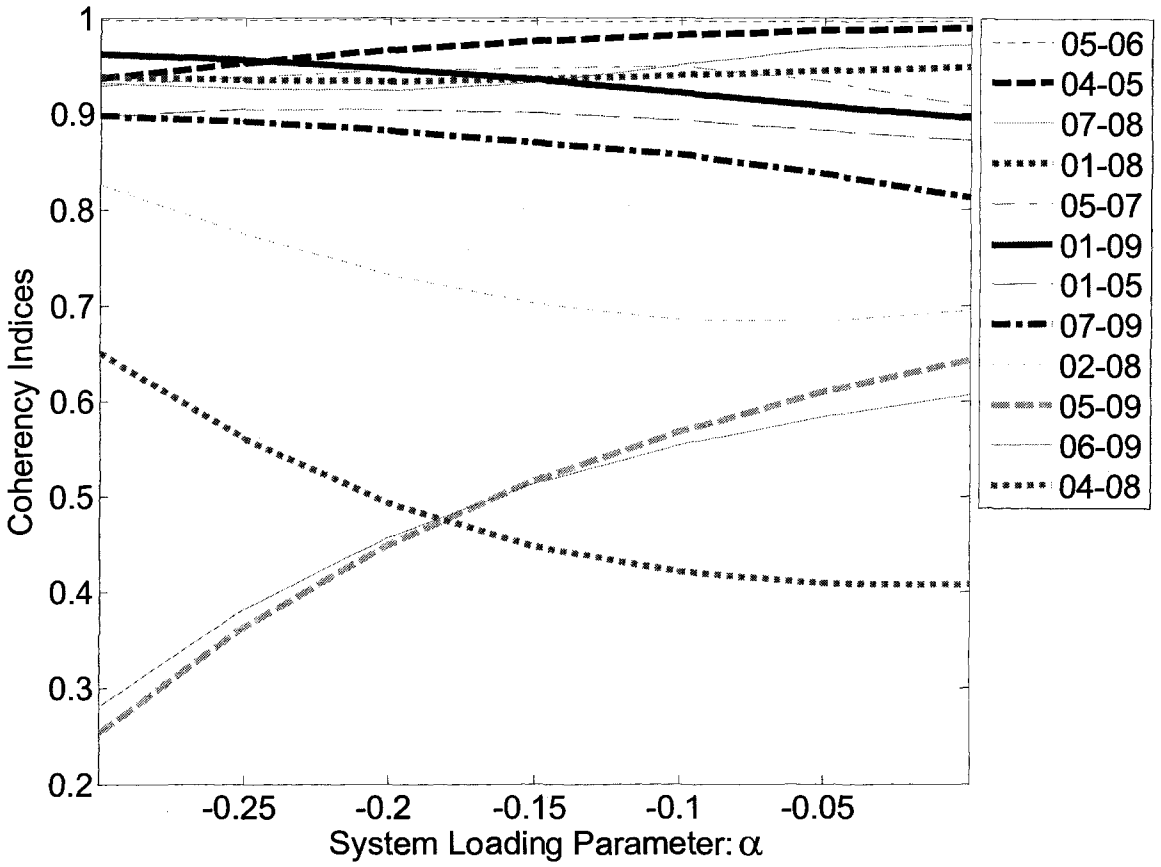
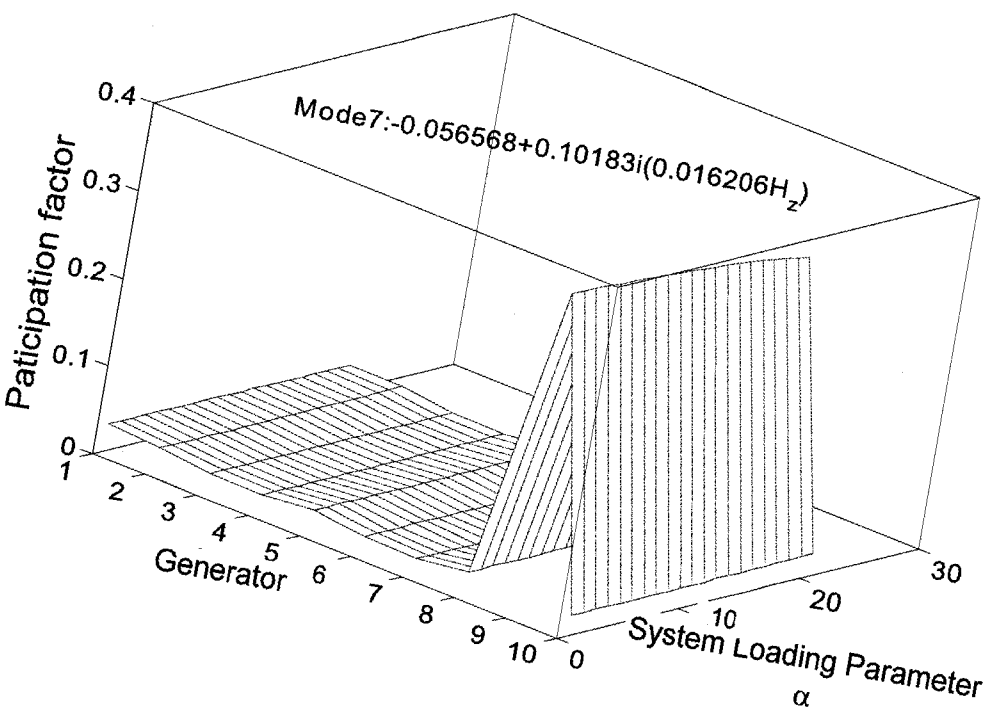


Figure 3-4 Coherency Indices under small operating range (Four Slow Modes) in Scenario I

The slow modes and their corresponding eigenvectors are calculated by the continuation method, and participation factors for the slow modes in the generator rotor angle over the range of operating points are evaluated. This is shown in Figure 3-5. It is observed that these slow modes continue to participate in generator rotor angle dynamics over the entire range of operating conditions.

Generator grouping is calculated by using the DYNRED package [32]. For the base case (α=0), the results of this analysis indicate that the generators should be assigned to the three groups shown in TABLE 3-4, which agrees with the results obtained using the continuation method in Figure 3-4.



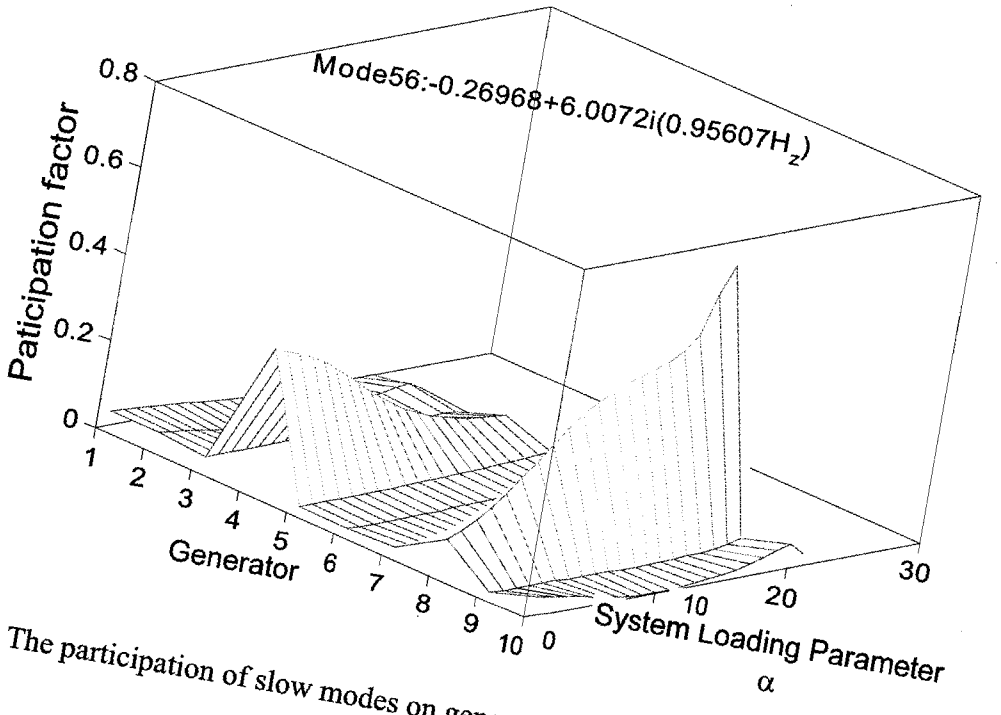


Figure 3-5 The participation of slow modes on generator angle over the range of operating points

TABLE 3-4 BASE CASE GENERATOR GROUPING FROM DYNRED

Group No.	Generator No.
1	2, 3, 4, 5, 6, 10
2	1, 7, 8
3	9

To verify the result using time-domain simulation, a small disturbance is applied to the system for Scenario I. For the base case, at 0.1s, lines 33-18 and 34-14 are removed, and at 0.5s, these two lines are reconnected. Figure 3-6 illustrates the generator relative rotor angle dynamics. It is observed that Generators 2, 3, 4, 5, and 6 are coherent when the disturbance occurs; Generators 1, 7, and 8 also have tight coherency, while Generator 9 does not show any coherency with other generators. From Figure 3-6, it can also be observed that, even within the coherent group-Generators 1, 7, and 8, Generator 8 has a slightly different behavior, which is captured by the continuation method.

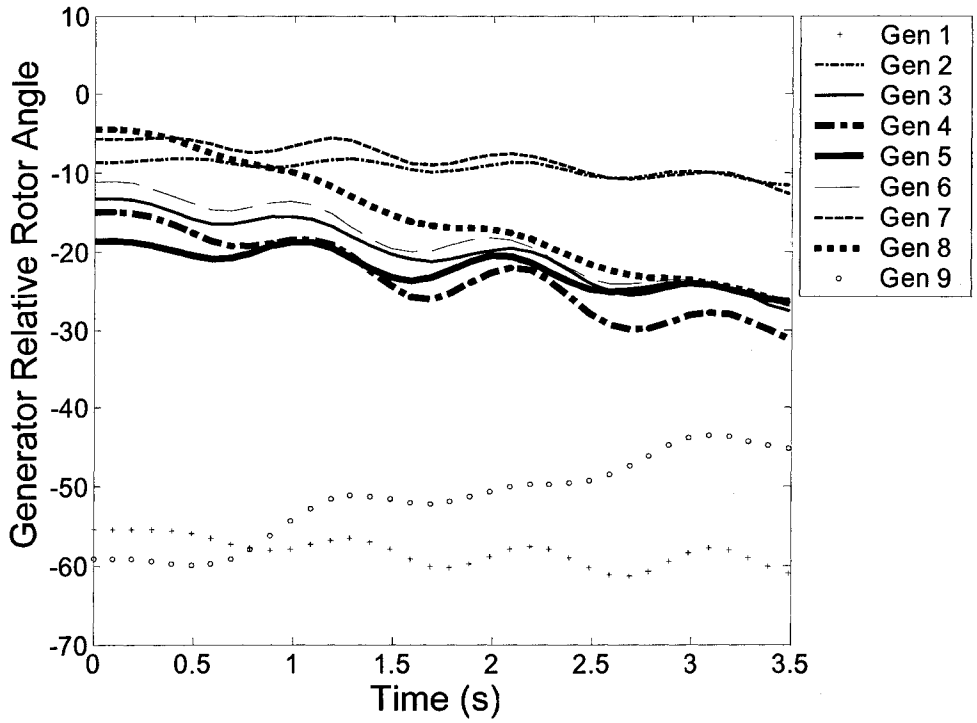


Figure 3-6 Time domain generator relative rotor angle under small disturbance for Scenario I at base case

In order to examine the effect of increasing load we reconsider Scenario I. The load at all buses is varied uniformly from the base case using a range $-0.3 \leq \alpha \leq 0.2$. The coherency indices are traced using the continuation method. Figure 3-7 shows the loci of the coherency indices between generators in this range. The vertical line A indicates the critical point (α =0.15) at which the system becomes steady-state unstable. It is of interest to note, from Figure 3-7, that almost all the coherency indices loci change significantly in behavior when they cross the critical point. The coherency indices at α =0.12 are shown in TABLE 3-5. It is observed that there is still three groups Group 2 (Generator 1, 7, and 8), Group 1 (Generator 2, 3, 4, 5, and 6), and Group 3 (Generator 9). Generator 9 still does not show any coherency with other generators. To verify the result, the generator grouping is also calculated using DYNRED. The groups are consistent with those obtained by the continuation method. The continuation method provides not only the generator coherency indices at each operating point but also shows the transition of coherency indices from one operating point to another as shown in Figure 3-7. The relative rotor angles following the same disturbance as described above are shown in Figure 3-8 at α =0.12. It is observed that the behavior is consistent with that predicted by the continuation method.

TABLE 3-5 SCENARIO I – COHERENCY INDICES BETWEEN PAIRS OF GENERATORS AT HEAVY LOAD ($\alpha = 0.12$)

Index	G1	G2	G3	G4	G5	G6	G7	G8	G9
G1	X	0.86	0.87	0.78	0.87	0.88	**0.98**	**0.95**	0.88
G2	0.86	X	**0.96**	**0.94**	**0.98**	**0.96**	0.93	0.78	0.54
G3	0.87	0.96	X	**1.00**	**1.00**	**1.00**	0.87	0.64	0.74
G4	0.78	0.94	1.00	X	**0.99**	**0.99**	0.77	0.45	0.66
G5	0.87	0.98	1.00	0.99	X	**1.00**	0.89	0.72	0.70
G6	0.88	0.96	1.00	0.99	1.00	X	0.90	0.74	0.63
G7	0.98	0.93	0.87	0.77	0.89	0.90	X	**0.97**	0.79
G8	0.95	0.78	0.64	0.45	0.72	0.74	0.97	X	0.72
G9	0.88	0.54	0.74	0.66	0.70	0.63	0.79	0.72	X

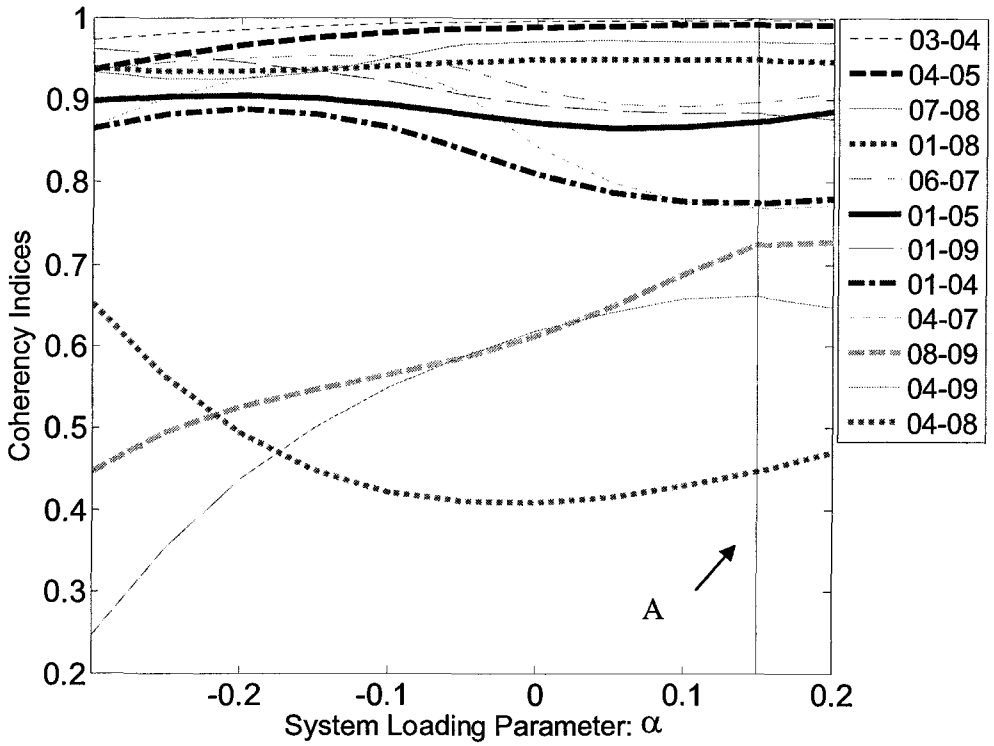


Figure 3-7 Generator Coherency Indices under large operating range for Scenario I

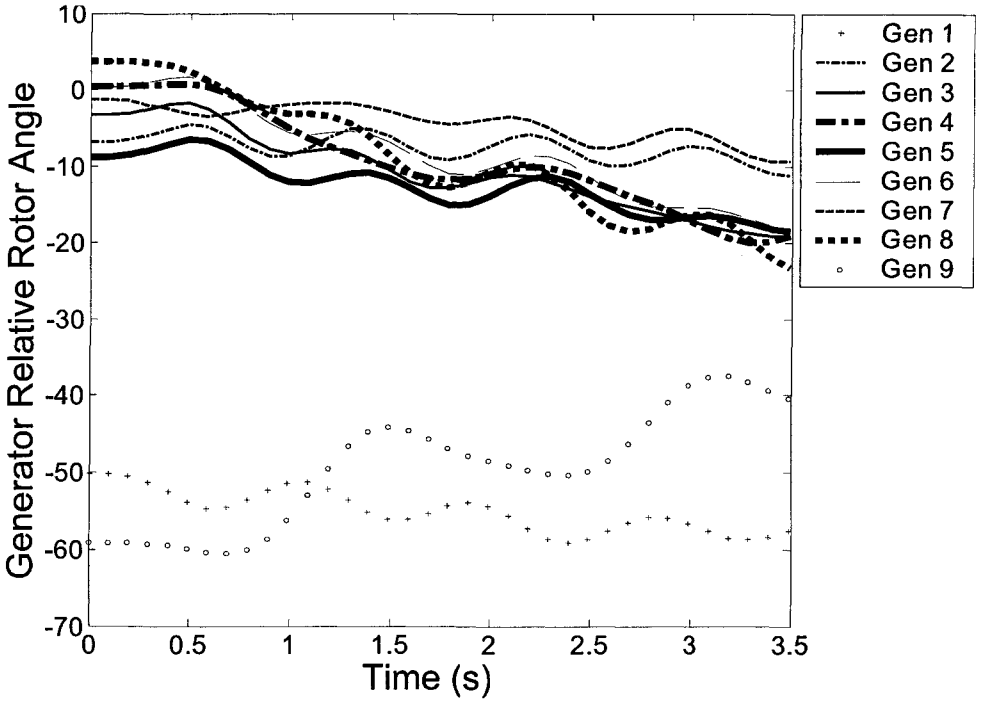


Figure 3-8 Transient rotor angle under small disturbance for Scenario I at α=0.12

Scenario II: This load scenario has been chosen such that the loads at specific load buses are decreased by load factor α. This analysis examines the effect of not increasing the load uniformly at all load buses. The load buses chosen are 15, 16, 18, and 34 and located around the center of the system. To compensate for the change in load, the generation at each generator bus is increased uniformly by a factor α.

The traces of the loci of the generator coherency indices as load factor α varies are shown in Figure 3-9. It may be observed from Figure 3-9 that the traces of coherency indices are different from Scenario I. All the loosely-coherent generators are more tightly coupled but with indices of lower magnitude compared to scenario I. However, none of the generators change their group with the decrease in loading conditions.

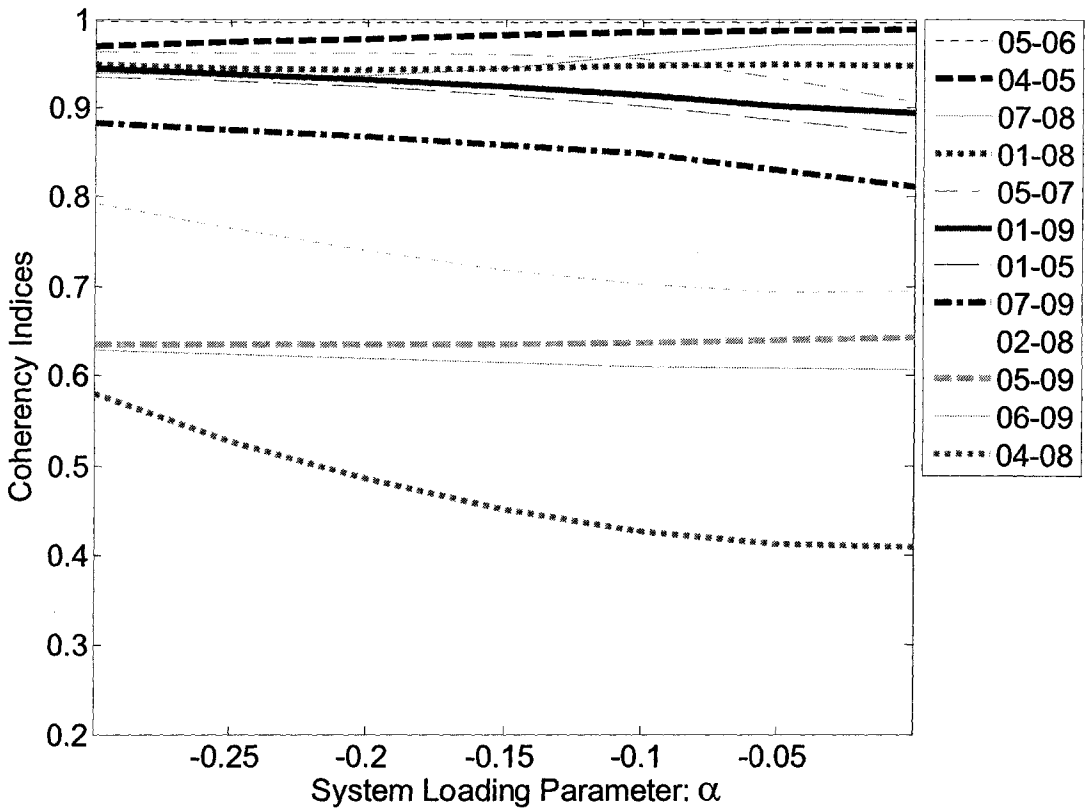


Figure 3-9 Generator Coherency Indices under small operating range (Four Slow Modes) in Scenario II

Scenarios III & IV: Scenarios III and IV are introduced to investigate the impact of the load model on generator coherency. The load model used in Scenario III is ZIP with a combination of constant impedance (40%), constant current (30%), and constant power component (30%). In scenario IV, the load model with a combination of constant impedance (50%), constant current (30%), and constant power component (20%) is used. A frequency component (with a coefficient of 20%) has also been included in both scenarios. The load scenario has been chosen such that the load at each load bus and generation at each generator bus uniformly vary by load factor, α.

The generator coherency indices for Scenarios III and IV are shown in TABLE 3-6 and TABLE 3-7 respectively, which indicate that compared to Scenario I, the values of the coherency indices do not change between generators within a group. However, generators within different groups are loosely coupled.

TABLE 3-6 SCENARIO III - COHERENCY INDICES BETWEEN PAIRS OF GENERATORS AT BASE CASE

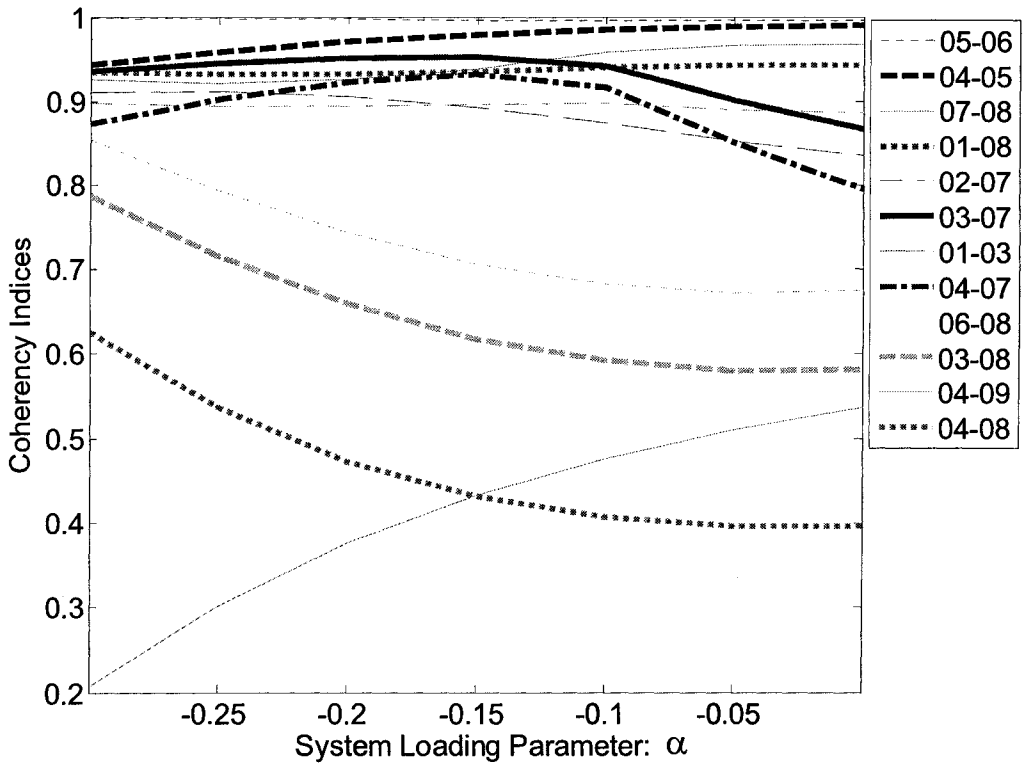
Index	G1	G2	G3	G4	G5	G6	G7	G8	G9
G1	X	0.81	0.84	0.76	0.84	0.85	***0.98***	***0.94***	0.89
G2	0.81	X	**0.96**	**0.94**	**0.98**	**0.96**	0.89	0.70	0.51
G3	0.84	0.96	X	**1.00**	**1.00**	**1.00**	0.87	0.58	0.62
G4	0.76	0.94	1.00	X	**0.99**	**0.99**	0.80	0.40	0.54
G5	0.84	0.98	1.00	0.99	X	**1.00**	0.88	0.66	0.56
G6	0.85	0.96	1.00	0.99	1.00	X	0.89	0.67	0.52
G7	0.98	0.89	0.87	0.80	0.88	0.89	X	***0.97***	0.81
G8	0.94	0.70	0.58	0.40	0.66	0.67	0.97	X	0.58
G9	0.89	0.51	0.62	0.54	0.56	0.52	0.81	0.58	X

As the load factor α varies the loci of the coherency indices are shown in Figure 3-10. Compared with Figure 3-4, it can be observed that, with a change of load model component (the component of constant impedance increases, while the component of constant power decreases) most of the coherency indices between loosely coherent generators diminish.

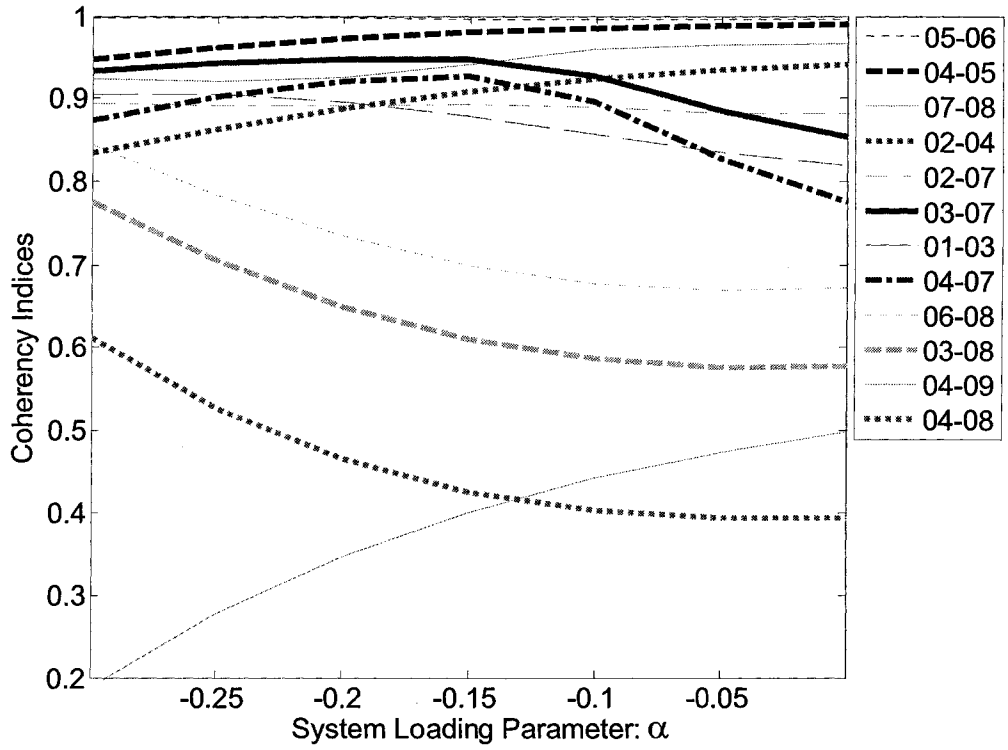
However, it seems that the change in load model components does not affect the tightly-coherent generators significantly. As a result, the grouping does not change.

TABLE 3-7 SCENARIO IV - COHERENCY INDICES BETWEEN PAIRS OF GENERATORS AT BASE CASE

Index	G1	G2	G3	G4	G5	G6	G7	G8	G9
G1	X	0.80	0.82	0.74	0.82	0.83	***0.98***	***0.94***	0.89
G2	0.80	X	**0.96**	**0.94**	**0.98**	**0.96**	0.88	0.70	0.49
G3	0.82	0.96	X	**1.00**	**1.00**	**1.00**	0.85	0.58	0.58
G4	0.74	0.94	1.00	X	**0.99**	**0.99**	0.78	0.39	0.50
G5	0.82	0.98	1.00	0.99	X	**1.00**	0.87	0.66	0.51
G6	0.83	0.96	1.00	0.99	1.00	X	0.87	0.67	0.48
G7	0.98	0.88	0.85	0.78	0.87	0.87	X	***0.97***	0.81
G8	0.94	0.70	0.58	0.39	0.66	0.67	0.97	X	0.56
G9	0.89	0.49	0.58	0.50	0.51	0.48	0.81	0.56	X



a Scenario III (cont'd)



b Scenario IV

Figure 3-10 Generator Coherency Indices under small operating range (Four Slow Modes)

The analysis of different load scenarios suggests that the load component of constant power indeed affects the generator coherency indices. Results from the test system indicate that, with change of constant power, most of the coherency indices between loosely coherent generators diminish, while there is somewhat less effect on tightly-coherent generators. These results have been verified using DYNRED.

3.4.2 WECC 29 Gen - 179 Bus system

TABLE 3-8 shows the selected slow modes, which have the largest participation factors of the generator rotor angle and speed for the base case,.

TABLE 3-8 SYSTEM SELECTED SLOW MODES AT BASE CASE

Mode	Value	Frequency (HZ)	Damping (%)
30	-0.0888+0.2022i	0.0322	40.19
87	0.6117+3.4784i	0.5536	-17.32
94	0.5694+4.5644i	0.7264	-12.38
131	0.5696+6.1331i	0.9761	-9.25
134	0.2172+6.3503i	1.0107	-3.42
141	0.1210+8.3000i	1.3210	-1.46
146	0.3830+9.3363i	1.4859	-4.10
150	0.2826+9.8337i	1.5651	-2.87
152	0.2532+9.9903i	1.5900	-2.53
158	0.0823+11.3521i	1.8067	-0.72
160	-0.0682+11.9837i	1.9073	0.57
162	0.3259+12.0707i	1.9211	-2.70

DYNRED suggests that, at the base case (α=0), generators will be formed into four groups, as shown in TABLE 2-1.

Figures 3.11 – 3.14 illustrate how the selected coherency indices between any pair of generators change with respect to system loading conditions in the 29 Generator 179 Bus WECC by taking twelve slow modes into account. The first conclusion that can be drawn from Figures 3.11 – 3.14 is that the generator coherency indices remain flat within certain range of operating conditions, indicating that the generator grouping information stays fairly constant. Second, the result from our approach shows that Generator 140 is coherent with Generators 43, 149, 47, 40, 144, 13, 148, 138, 103, 15; Generators 11, 159, 45, 6, 18, 36, 4, 9, 162 are within one group; Generator 79 shows coherency with Generators 30, 65, 112, 116 118; Generator 35 is not coherent with any other generators, results which are quite consistent with the results from DYNRED. The details of coherency indices between pairs of generators have been shown in TABLE 3-9. However, Generator 79 shows loose coherency with Generator 70, which is not consistent with DYNRED.

To verify the result using time-domain simulation, a large disturbance was applied to the system. For the base case at 0s, lines 83-168, 83-170 and 83-172 were removed. Figure

3-15 illustrates the generator relative rotor angle dynamics after this large disturbance. The groups shown in the legend are consistent with TABLE 2-1. Since Generator 70 shows a difference in characteristic in comparison with DYNRED, its plot is separated from Group4, as shown in Figure 3-15. The grouping information obtained from continuation method for Group1 and Group3 is identical to the nonlinear time-domain simulation. In Group2, Generator 36 shows a slightly different dynamic behavior from other generators in the same group. The reason is that generator 36 is too close to the disturbance. Therefore, 0.05 second after the disturbance, its relative angle deviates from the others, as shown in Figure 3-15. In Group 4, the relative angles of Generator 112, 116, 118 decelerated more than the other generators 0.15 second after the disturbance, because they are very close to the disturbance. Generator 70 is also close to the disturbance. At the beginning of the disturbance Generator 70 is in Group1, but it shows a tendency toward acceleration.

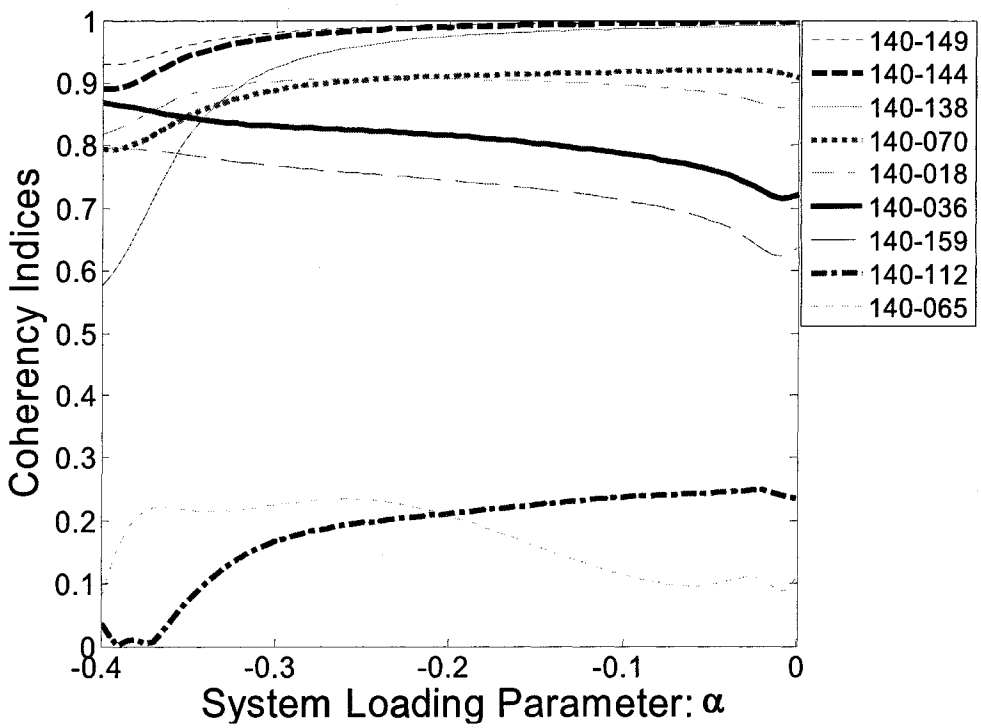


Figure 3-11 Coherency Indices between Generator 140 and other selected generators

TABLE 3-9 GENERATOR COHERENCY INDICES BETWEEN PAIR OF GENERATORS FOR WECC SYSTEM

Index	G140	G11	G35	G79	Index	G140	G11	G35	G79
140	X	0.71	0.02	0.12	6	0.75	0.99	0.07	0.05
43	1.00	0.77	0.08	0.12	18	0.86	0.99	0.07	0.00
149	1.00	0.73	0.04	0.10	36	0.72	0.99	0.23	0.51
47	1.00	0.78	0.13	0.19	4	0.91	0.99	0.09	0.02
40	1.00	0.73	0.06	0.09	9	0.87	0.99	0.06	0.00
144	1.00	0.75	0.04	0.11	162	0.45	0.95	0.30	0.56
13	1.00	0.86	0.04	0.04	35	0.02	0.08	X	0.33
148	0.99	0.80	0.04	0.08	79	0.12	0.07	0.33	X
138	0.99	0.81	0.03	0.02	112	0.24	0.22	0.31	0.96
103	0.99	0.63	0.10	0.33	118	0.80	0.47	0.18	0.96
15	0.98	0.74	0.01	0.05	30	0.19	0.07	0.58	0.95
11	0.71	X	0.08	0.07	116	0.70	0.44	0.17	0.94
159	0.64	1.00	0.19	0.29	65	0.11	0.25	0.10	0.88
45	0.60	0.99	0.20	0.30	70	0.91	0.75	0.13	(0.41)

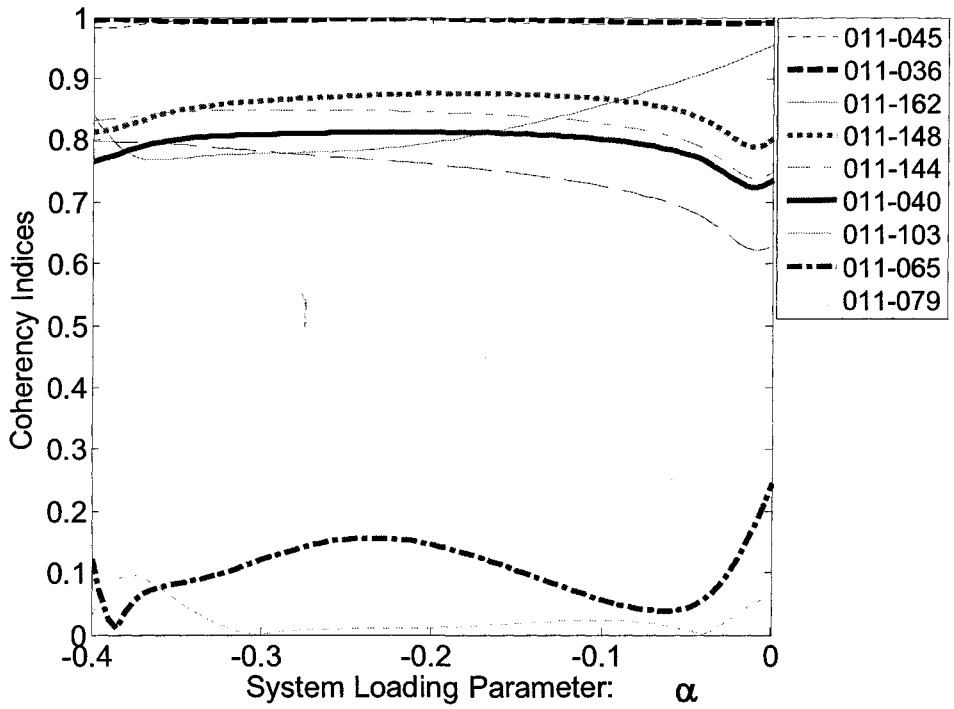


Figure 3-12 Coherency Indices between Generator 11 and other selected generators

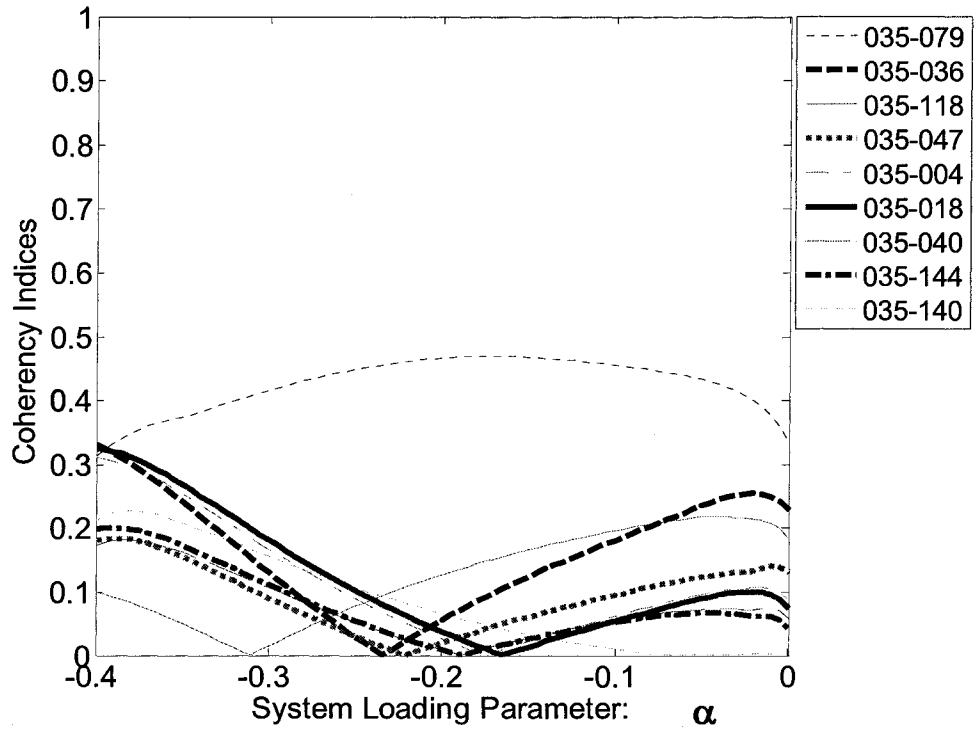


Figure 3-13 Coherency Indices between Generator 35 and other selected generators

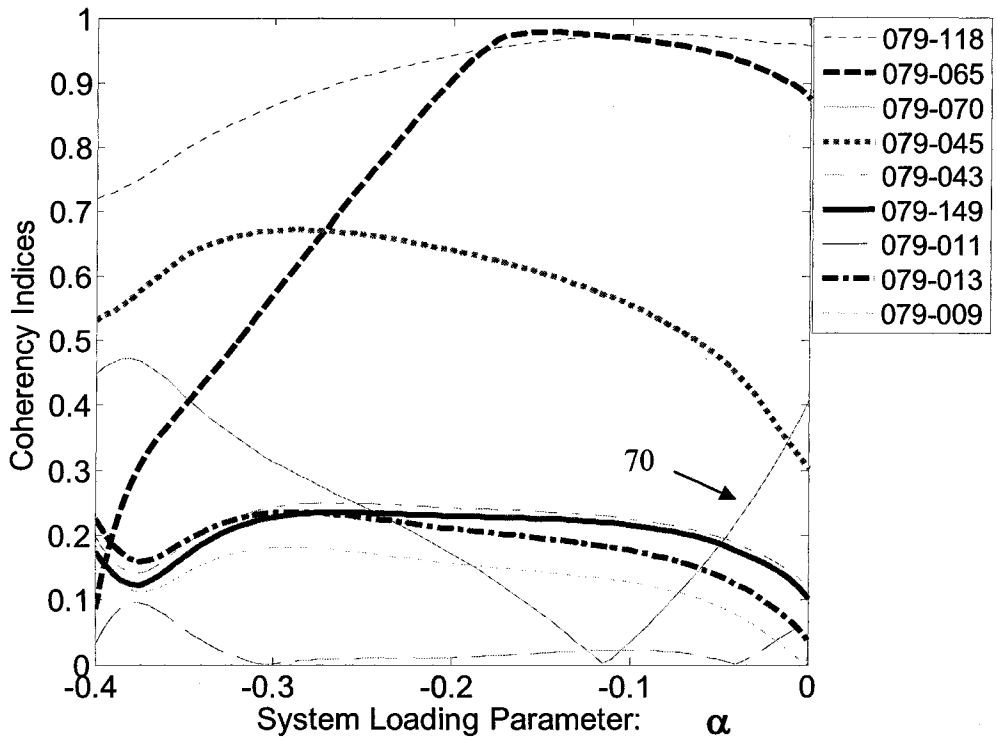


Figure 3-14 Coherency Indices between Generator 79 and other selected generators

DYNRED gives almost the same grouping, except generator 70, which has been assigned to Group4, as shown above in TABLE 2-1. But as seen in Figure 3-15, there is no obvious evidence that generator should belong to Group4.

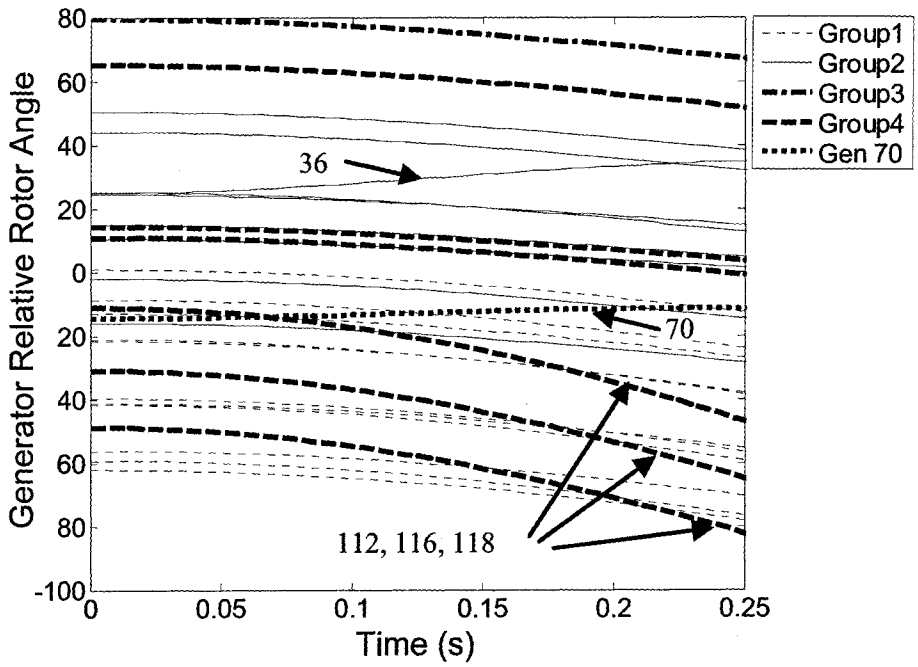


Figure 3-15 Time domain generator relative rotor angles under large disturbance at α =0 (base case)

3.5 Summary

In this dissertation, a novel generator-coherency-indices tracing approach using the continuation method has been presented. It involves modeling of appropriate power system dynamics and network representation to obtain the Jacobian matrix, and use of a globally convergent technique to make the continuation method applicable. Using this approach the trace of the loci of the generator coherency indices over a range of system operating conditions can be obtained. The loci provide information about the grouping associated with the slow modes. This information is critical in techniques which use slow-coherency

grouping to develop corrective control techniques such as adaptive islanding [3] to prevent cascading outages. The grouping information obtained from the continuation-method based loci determines how the loading condition will affect the islands that are formed. This information is essential for developing an islanding strategy and determining islands that have the optimal generation load balance.

The investigation of the impact of load component on generator coherency indices shows that a constant impedance load component can make loosely coherent generators less coherent, while it has little effect on tightly coherent generators. In this approach the technique was demonstrated on a small test system which was highly equivalenced and a moderate sized system representing the WECC. As a result, the coherent grouping does not change much. However, the technique developed will capture changes in grouping in realistic systems.

In this dissertation, we try to present a new approach that could be used to extend the type of generator grouping method that is originally computed at one operating condition. The coding language used was MATLAB, and the code developed has not been optimized since it was only developed to provide a candidate method and to illustrate the potential of the method. From (3.26), it may observed that J_{AUG} is a sparse matrix, which means that with the well-developed sparse matrix technique, and with optimization technique, our approach can be realized with high efficiency.

We believe that one of the advantages of our approach is that the sensitivity of the coherency indices can be obtained from the Predictor. Therefore, a clear picture not only of the generator coherency indices under the current operating condition, but also of sensitivity to changes in load conditions can be obtained without performing power flow and small signal analysis at each operating condition. Furthermore, parameter α in this approach is not limited to load changes; actually we can also investigate the relationship between generator coherency and other parameters such as control variables.

Theoretically, this approach can be applied to any system. However, the disadvantage of

the approach is that it requires a Jacobian matrix, which is difficult to obtain when the system scale becomes large, and in many case analytical derivatives are even unavailable. It gets even worse if multiple slow-modes tracing is conducted because the dimension of the augment matrix increases dramatically. One possible alternative is to undertake Matrix reduction (Guyan Reduction, IRS Reduction, etc.) [42], by which only certain states such as generator rotor angle states will be retained for the computation,. Furthermore, it will be promising for an approach that does not require the computation of the Jacobian matrix. There are quasi-Newton methods that provide inexpensive approximations to the Jacobian matrix for root-finding. Probably the best method among the methods described in the literature is Multidimensional Secant Methods: Broyden's Method [41].

4 CONCLUSION AND FUTURE WORK

4.1 Conclusion

Power system islanding is considered as such a rare or improbable event that it may seem that it does not deserve a great deal more attention. However, the result of unintentional islanding on power systems and electricity customers leads individuals and the public to have great concern. It is critically important to bring to the table the question of how to conduct controlled system islanding as a last resort when large disturbances occur in the system, especially under the circumstance of a deregulated power market in which power systems are being operated close to their limits.

In this research, two technical issues have been discussed regarding automatic power system islanding, taking both system network topology and component dynamic characteristics into consideration, listed as follows:

1) **The first issue is how to find the paths that will propagate cascading events once a large disturbance is initialized in the system.**

   i. Generator Grouping based on Slow Coherency

The slow-coherency technique has mainly been used to conduct power system network reduction. However, it has also shown great applicability in power system generator grouping to investigate the strong connections among coherent generators and the weak connections among general groups of generators.

Based on slow coherency theory, it is the weak connection between the groups of the generators that will most likely have the greatest impact on the system and propagate cascading events. Two assumptions have been made: 1) the coherent groups of generators are almost independent of the size of the disturbance so that the linearized model can be used to determine coherency. 2) the coherent groups are independent of the level of detail used in modeling the generating unit so that a classical model may be used to model the generator.

Once the grouping information is obtained, the power system network may be reduced in scale in such a way that generators in the same group can be represented as only one single bus in the reduced network.

ii. Minimal Cutset Based Islanding

A minimal cutset technique which originates from Graph Theory has been applied to search the system to find the boundary of each island.

In the literature, most islanding schemes have focused on vertices (buses) other than edges (lines), since it is very straightforward to enumerate all the buses to obtain the imbalance of real power within the islands. However, the transition from vertices to lines makes it possible to obtain the same information by computing only the power flowing through the lines connecting to other islands.

The advantage by doing this is that the number of those lines is limited, one of the requirements of islanding. Therefore, the problem has been simplified into searching the minimal cutsets (MCs) to construct the island with the minimal net flow. We can decompose the islanding problem into two stages: a) find Minimal Cutset candidates; b) obtain the Optimal Minimal Cutset by various criteria. An automatic power system islanding program using minimal cutsets and breadth-first searching (BFS) flag-based depth-first searching (DFS) technique of Graph Theory has been developed to automatically determine where to create the island.

From the optimal-cutset and time-domain simulation for the WECC 29 generator and 179 bus system, it has been shown that the controlled islanding approach with an adaptive load-shedding scheme has the advantage of shedding fewer loads than that from islanding based on practical experience. Furthermore, it has also been shown that with the new islanding scheme, the system has experienced less frequency oscillation than with islanding based on practical experience.

2) The second issue is to investigate the relationship between generator coherencies

and different operating conditions.

Slow coherency does not claim consistency under varying load conditions. With the deregulated environment, the need for the electrical power has grown constantly. One may ask how to extend slow coherency at one operating condition to a range of operating conditions.

A novel approach has been presented to update coherency information if needed by using the predictor and corrector type of continuation method. This method combines the power system DAE model together with the eigenspace equations to form a complete set of nonlinear equations to trace the relationship between generator coherency indices and the load condition.

In the literature, the continuation method mainly has been used in voltage stability which only considers power system steady state. In this dissertation, a systematic approach has been presented to utilize all the available information, including system dynamic and steady state information, as well as eigenspace equations to formulate the situation in such a way that it projects the interaction between system dynamics and load condition into the movement of the eigenvalues and eigenvectors in eigenspace. This approach has been applied to a 10 generator 39 bus New England system and a 29 generator 179 bus model of the WECC system. The result shows that it is compatible with DYNRED and time-domain simulation. The impact of load variation on generator coherency indices has also been investigated.

The advantage and disadvantages of the approach have also been addressed. Alternative improvements have been pointed out in the summaries at the end of each chapter.

4.2 Future Work

Future work will focus on how to improve the performance of this approach and apply it to much larger scale systems. There is significant necessary work to be done since academic research work almost always involves small-scale systems with several hundred buses or

less. In order to handle large-scale systems, the program needs to be modified such that it can provide a better mechanism to support large data structures and to manipulate large-volume data efficiently.

In this dissertation, two different aspects of controlled system islanding have been presented. One is a power system islanding scheme using graph theory technique at one operating point. The other is to investigate the relationship between generator coherency and system operating conditions. Studies need to be conducted to combine these two approaches into one comprehensive strategy to adaptively conduct power system islanding according to various system operating conditions.

4.3 Contribution

For modern power systems, catastrophic cascading events can cause huge losses to the economy and society. By using the minimal cutest technique, a controlled system islanding program has been developed in this dissertation, based on slow coherency grouping information. The most significant contributions may be summarized as follows:

1. A comprehensive approach: Different from other approaches in the literature, this approach takes both system dynamic characteristics and power system network topology into consideration.
2. Two tier islanding scheme: this approach makes it possible to decompose the islanding scheme into two stages: 1) consider the system dynamics and find out the weak connection among generators; 2) find out the minimal cutset space based on the system topology information and obtain the optimal cutset by computing the net real flow on each cutset. Another advantage is that, in the second stage, we can also apply any additional criteria to formulate the optimization objective function under different conditions, such as the requirements for system restoration, while the first stage remains unchanged.
3. Both slow-coherency theory and the minimal cutsets method have been widely used

in different applications. However, this is the first time they have been introduced together to solve the system-islanding problem. This may be of great interest to the power industry after the recent blackout in the United States and Canada [5] because the proposed method provides a completely new strategy for corrective action following large disturbances in power system.

4. A study has been conducted to investigate the relationship between slow coherency and system load conditions. It has been shown that slow coherency may not be consistent with system load conditions, which has led us to adopt the slow-coherency group-tracing method to adjust the islanding scheme with respect to system load conditions. In the literature, the continuation method has mainly been used in voltage stability studies which have only considered power system steady-state. The approach proposed in this dissertation combines the power system DAE model with the eigenspace equations to form a complete set of nonlinear equations to trace the relationship between generator coherency indices and the load condition.

APPENDIX

A.1 New governor model in PSAPAC format

```
RGEN
   45    1  100  100
   99   99   99   99   50   99   99   99   99   99   99   99
    0
    0
        B1        LL       1.0       0.0       1.0
        B2        IN     0.015       1.0      -1.0
        B3        LL      20.0       0.0       0.1
        B4        GN       5.0       1.0      -1.0
        B5        IN       1.0       1.0       0.0
        B6        T4      0.10      8.72     2.616       0.4
                          0.95      0.22      0.22      0.30       0.0       0.0

    1   -2    2    4    3   -4    4    5    5    6
    1   PT         1    2 CONT     17.80    2 CONT        -1    3   DW         1
    4 CONT         1    4   VG         1    1 VREF         1
EDATA
```

A.2 MATLAB Code

A.2.1 Jacobian matrix function

```
function [Fx, Fy, Gx, Gy,
Fa,Ga]=getjacobian(x,Sys,Machines,Governors,Excitors,Loads,Ybus_abs,Ybus_angle,alpha)

M=Sys(1);
N=Sys(2);
NS=Sys(3);
% note: all the vectors are row vectors.
Epq=x(1:NS:NS*(M-1)+1); Epd=x(2:NS:NS*(M-1)+ 2); Efd=x(3:NS:NS*(M-1)+3);
Vr=x(4:NS:NS*(M-1)+ 4); Rf=x(5:NS:NS*(M-1)+ 5); Pm=x(6:NS:NS*(M-1)+6);
Miu=x(7:NS:NS*(M-1)+7); omega=x(NS-1:NS:NS*(M-1)+ NS-1); delta=[x(NS:NS:NS*(M-1)) 0];
v=x(NS*M+2*[0:N-1]);
theta=x(NS*M+1+2*[0:N-1]);
%%%%%%%%%%machine%%%%%%%%%%%%%%%%%%
R=Machines(1,:); Xd=Machines(2,:); Xq=Machines(3,:); Xpd=Machines(4,:);
Xpq=Machines(5,:);Tpd0=Machines(6,:); Tpq0=Machines(7,:); Mg=Machines(8,:);
D=Machines(9,:); Pg0=Machines(10,:); Km=Machines(11,:);
%%%%%%%%%%governor%%%%%%%%%%%%%%%%%%
```

```
K1=Governors(1,:); Tg=Governors(2,:); Tch=Governors(3,:);
%%%%%%%%%%%excitor%%%%%%%%%%%%%%%%%
Ka=Excitors(1,:); Ta=Excitors(2,:); Ke=Excitors(3,:); Te=Excitors(4,:);
Se=Excitors(5,:); Kf=Excitors(6,:); Tf=Excitors(7,:); Vref=Excitors(8,:);
%%%%%%%%%%%loads%%%%%%%%%%%%%%%%%%%%
v0=Loads(1,:); Pl0=Loads(2,:); Kpp=Loads(3,:); Kip=Loads(4,:);
Kzp=Loads(5,:); Kpl=Loads(6,:);

Ql0=Loads(7,:); Kpq=Loads(8,:); Kiq=Loads(9,:); Kzq=Loads(10,:);
Kql=Loads(11,:);

Kw=Loads(12,:);

%%%%%%%%%%%%%% getFx %%%%%%%%%%%%%%%%%%%%
Fx=sparse(NS*M-1,NS*M-1);

fx=sparse(NS,NS);
temp=[];

for i=1:M-1
    A=R(i)^2+Xpd(i)*Xpq(i);
    fx(1)=-(R(i)^2+Xpq(i)*Xd(i))/A/Tpd0(i);
    fx(2)=-(-Xq(i)+Xpq(i))*R(i)/A/Tpq0(i);
     fx(8)=(-2*R(i)^3*Epq(i)+2*R(i)^2*Xpd(i)*Epd(i)+R(i)^3*v(i)*cos(delta(i)-theta(i))-2*R(i)^
        2*Xpd(i)*v(i)*sin(delta(i)-theta(i))-R(i)*Xpd(i)*Xpq(i)*v(i)*cos(delta(i)-theta(i))-2*X
        pq(i)*R(i)^2*Epd(i)-2*Xpq(i)^2*R(i)*Epq(i)+Xpq(i)*R(i)^2*v(i)*sin(delta(i)-theta(i))+2*
        Xpq(i)^2*R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)^2*Xpd(i)*v(i)*sin(delta(i)-theta(i)))/
        A^2/Mg(i);
    fx(10)=(-Xd(i)+Xpd(i))*R(i)/A/Tpd0(i);
    fx(11)=-(R(i)^2+Xpd(i)*Xq(i))/A/Tpq0(i);
     fx(17)=-(-2*Xpd(i)*R(i)^2*Epq(i)+2*Xpd(i)^2*R(i)*Epd(i)+Xpd(i)*R(i)^2*v(i)*cos(delta(i)-t
        heta(i))-2*Xpd(i)^2*R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)^2*Xpq(i)*v(i)*cos(delta(i)-
        theta(i))+2*R(i)^3*Epd(i)+2*R(i)^2*Xpq(i)*Epq(i)-R(i)^3*v(i)*sin(delta(i)-theta(i))-2*R
        (i)^2*Xpq(i)*v(i)*cos(delta(i)-theta(i))+R(i)*Xpq(i)*Xpd(i)*v(i)*sin(delta(i)-theta(i))
        )/A^2/Mg(i);
    fx(19)=1/Tpd0(i);
    fx(21)=-(Se(i)+Ke(i))/Te(i);
    fx(23)=-(Se(i)+Ke(i))*Kf(i)/Te(i)/Tf(i);
    fx(30)=1/Te(i);
    fx(31)=-1/Ta(i);
    fx(32)=Kf(i)/Te(i)/Tf(i);
    fx(40)=-Ka(i)/Ta(i);
    fx(41)=-1/Tf(i);
    fx(51)=-1/Tch(i);
    fx(53)=1/Mg(i);
    fx(60)=1/Tch(i);
    fx(61)=-1/Tg(i);
    fx(70)=-K1(i)/Tg(i);
    fx(71)=-D(i)/Mg(i);
    fx(72)=120*pi;
     fx(73)=-(-Xd(i)+Xpd(i))*v(i)*(R(i)*cos(delta(i)-theta(i))-Xpq(i)*sin(delta(i)-theta(i)))/
         A/Tpd0(i);
```

```
    fx(74)=-(-Xq(i)+Xpq(i))*v(i)*(R(i)*sin(delta(i)-theta(i))+Xpd(i)*cos(delta(i)-theta(i)))/
        A/Tpq0(i);
    fx(80)=v(i)*(-Xpd(i)*R(i)^2*v(i)+Xpq(i)*Xpd(i)^2*v(i)-Xpq(i)^2*Xpd(i)*v(i)+Xpq(i)*R(i)^2*
        v(i)+Epd(i)*R(i)^3*cos(delta(i)-theta(i))-Epq(i)*R(i)^3*sin(delta(i)-theta(i))+2*R(i)*
        cos(delta(i)-theta(i))*Xpd(i)^2*Epd(i)-Xpq(i)*sin(delta(i)-theta(i))*Xpd(i)^2*Epd(i)-2
        *Xpd(i)*R(i)^2*cos(delta(i)-theta(i))*Epq(i)+2*Xpd(i)*R(i)^2*cos(delta(i)-theta(i))^2*
        v(i)+Xpd(i)*R(i)^2*Epd(i)*sin(delta(i)-theta(i))-2*Xpd(i)^2*Xpq(i)*v(i)*cos(delta(i)-t
        heta(i))^2-2*Xpq(i)^2*sin(delta(i)-theta(i))*R(i)*Epq(i)-Xpd(i)*cos(delta(i)-theta(i))
        *Xpq(i)^2*Epq(i)+2*Xpd(i)*cos(delta(i)-theta(i))^2*Xpq(i)^2*v(i)-2*Xpq(i)*R(i)^2*sin(d
        elta(i)-theta(i))*Epd(i)+Xpq(i)*R(i)^2*Epq(i)*cos(delta(i)-theta(i))-2*Xpq(i)*R(i)^2*v
        (i)*cos(delta(i)-theta(i))^2-2*R(i)*cos(delta(i)-theta(i))*Xpd(i)^2*v(i)*sin(delta(i)-
        theta(i))+Xpd(i)*Xpq(i)*sin(delta(i)-theta(i))*R(i)*Epq(i)+2*Xpq(i)^2*sin(delta(i)-the
        ta(i))*R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*R(i)*cos(delta(i)-theta(i))*Xpd(i)*Epd(
        i))/A^2/Mg(i);
    temp=sparse(blockdiag(temp,fx));
end


% for machine M
fx=sparse(zeros(NS-1,NS-1));

for i=M:M
    A=R(i)^2+Xpd(i)*Xpq(i);
    fx(1)=-(R(i)^2+Xpq(i)*Xd(i))/A/Tpd0(i);
    fx(2)=(Xq(i)-Xpq(i))*R(i)/A/Tpq0(i);
    fx(8)=(-2*R(i)^3*Epq(i)+2*R(i)^2*Xpd(i)*Epd(i)+R(i)^3*v(i)*cos(theta(i))+2*R(i)^2*Xpd(i)*
        v(i)*sin(theta(i))-R(i)*Xpd(i)*Xpq(i)*v(i)*cos(theta(i))-2*Xpq(i)*R(i)^2*Epd(i)-2*Xpq(
        i)^2*R(i)*Epq(i)-Xpq(i)*R(i)^2*v(i)*sin(theta(i))+2*Xpq(i)^2*R(i)*v(i)*cos(theta(i))+X
        pq(i)^2*Xpd(i)*v(i)*sin(theta(i)))/A^2/Mg(i);
    fx(9)=-(Xd(i)-Xpd(i))*R(i)/A/Tpd0(i);
    fx(10)=-(R(i)^2+Xpd(i)*Xq(i))/A/Tpq0(i);
    fx(16)=-(-2*Xpd(i)*R(i)^2*Epq(i)+2*Xpd(i)^2*R(i)*Epd(i)+Xpd(i)*R(i)^2*v(i)*cos(theta(i))+
        2*Xpd(i)^2*R(i)*v(i)*sin(theta(i))-Xpd(i)^2*Xpq(i)*v(i)*cos(theta(i))+2*R(i)^3*Epd(i)+
        2*R(i)^2*Xpq(i)*Epq(i)+R(i)^3*v(i)*sin(theta(i))-2*R(i)^2*Xpq(i)*v(i)*cos(theta(i))-R(
        i)*Xpq(i)*Xpd(i)*v(i)*sin(theta(i)))/A^2/Mg(i);
    fx(17)=1/Tpd0(i);
    fx(19)=-(Se(i)+Ke(i))/Te(i);
    fx(21)=-(Se(i)+Ke(i))*Kf(i)/Te(i)/Tf(i);
    fx(27)=1/Te(i);
    fx(28)=-1/Ta(i);
    fx(29)=Kf(i)/Te(i)/Tf(i);
    fx(36)=-Ka(i)/Ta(i);
    fx(37)=-1/Tf(i);
    fx(46)=-1/Tch(i);
    fx(48)=1/Mg(i);
    fx(54)=1/Tch(i);
    fx(55)=-1/Tg(i);

    fx(63)=-K1(i)/Tg(i);
end

Fx=blockdiag(temp,fx);
```

```
Fx(((M*NS-1)*(M*NS-2))+NS*(1:M-1))=-120*pi;
Fx(((M*NS-1)*(M*NS-2))-1+NS*(1:M-1))=D(1:M-1)./Mg(1:M-1);

%%%%%%%%%%%%%%%%%%%%%%%%getFy %%%%%%%%%%%%%%%%%%%
Fy=sparse(NS*M-1,2*N);

fy=zeros(NS,2);
temp=[];

for i=1:M-1
    A=R(i)^2+Xpd(i)*Xpq(i);

     fy(1)=-(-Xd(i)+Xpd(i))*(R(i)*sin(delta(i)-theta(i))+Xpq(i)*cos(delta(i)
        -theta(i)))/A/Tpd0(i);
     fy(2)=-(-Xq(i)+Xpq(i))*(-R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-theta(i)))/A/Tpq
        0(i);
     fy(4)=-Ka(i)/Ta(i);
     fy(8)=-(-Epd(i)*R(i)^3*sin(delta(i)-theta(i))-2*Xpd(i)*R(i)^2*sin(delta(i)-theta(i))*v(i)
        *cos(delta(i)-theta(i))+Xpd(i)*Xpq(i)*cos(delta(i)-theta(i))*R(i)*Epq(i)-2*Xpq(i)^2*co
        s(delta(i)-theta(i))*Xpd(i)*v(i)*sin(delta(i)-theta(i))+2*Xpq(i)*R(i)^2*sin(delta(i)-t
        heta(i))*v(i)*cos(delta(i)-theta(i))+Xpq(i)*R(i)*sin(delta(i)-theta(i))*Xpd(i)*Epd(i)-
        2*Xpq(i)*R(i)*Xpd(i)*v(i)-Xpq(i)*R(i)^2*Epq(i)*sin(delta(i)-theta(i))-2*Xpq(i)^2*cos(d
        elta(i)-theta(i))*R(i)*Epq(i)+2*Xpq(i)^2*cos(delta(i)-theta(i))^2*R(i)*v(i)+Xpd(i)*sin
        (delta(i)-theta(i))*Xpq(i)^2*Epq(i)-2*Xpq(i)*R(i)^2*cos(delta(i)-theta(i))*Epd(i)+2*Xp
        q(i)*cos(delta(i)-theta(i))*Xpd(i)^2*v(i)*sin(delta(i)-theta(i))+2*R(i)*Xpd(i)^2*v(i)-
        2*R(i)*Xpd(i)^2*v(i)*cos(delta(i)-theta(i))^2-Xpq(i)*cos(delta(i)-theta(i))*Xpd(i)^2*E
        pd(i)+2*Xpd(i)*R(i)^2*sin(delta(i)-theta(i))*Epq(i)+Xpd(i)*R(i)^2*Epd(i)*cos(delta(i)-
        theta(i))-Epq(i)*R(i)^3*cos(delta(i)-theta(i))-2*R(i)*sin(delta(i)-theta(i))*Xpd(i)^2*
        Epd(i))/A^2/Mg(i);
     fy(10)=-(-Xd(i)+Xpd(i))*v(i)*(-R(i)*cos(delta(i)-theta(i))+Xpq(i)*sin(delta(i)-theta(i)))
        /A/Tpd0(i);
     fy(11)=(-Xq(i)+Xpq(i))*v(i)*(R(i)*sin(delta(i)-theta(i))+Xpd(i)*cos(delta(i)-theta(i)))/A
        /Tpq0(i);
     fy(17)=v(i)*(Xpd(i)*R(i)^2*v(i)-Xpq(i)*Xpd(i)^2*v(i)+Xpq(i)^2*Xpd(i)*v(i)-Xpq(i)*R(i)^2*v
        (i)-2*R(i)*cos(delta(i)-theta(i))*Xpd(i)^2*Epd(i)+Xpq(i)*sin(delta(i)-theta(i))*Xpd(i)
        ^2*Epd(i)+2*Xpd(i)*R(i)^2*cos(delta(i)-theta(i))*Epq(i)-2*Xpd(i)*R(i)^2*cos(delta(i)-t
        heta(i))^2*v(i)-Xpd(i)*R(i)^2*Epd(i)*sin(delta(i)-theta(i))+2*Xpd(i)^2*Xpq(i)*v(i)*cos
        (delta(i)-theta(i))^2+2*Xpq(i)^2*sin(delta(i)-theta(i))*R(i)*Epq(i)+Xpd(i)*cos(delta(i
        )-theta(i))*Xpq(i)^2*Epq(i)-2*Xpd(i)*cos(delta(i)-theta(i))^2*Xpq(i)^2*v(i)+2*Xpq(i)*R
        (i)^2*sin(delta(i)-theta(i))*Epd(i)-Xpq(i)*R(i)^2*Epq(i)*cos(delta(i)-theta(i))+2*Xpq(
        i)*R(i)^2*v(i)*cos(delta(i)-theta(i))^2-Epd(i)*R(i)^3*cos(delta(i)-theta(i))+Epq(i)*R(
        i)^3*sin(delta(i)-theta(i))+2*R(i)*cos(delta(i)-theta(i))*Xpd(i)^2*v(i)*sin(delta(i)-t
        heta(i))-Xpd(i)*Xpq(i)*sin(delta(i)-theta(i))*R(i)*Epq(i)-2*Xpq(i)^2*sin(delta(i)-thet
        a(i))*R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*R(i)*cos(delta(i)-theta(i))*Xpd(i)*Epd(i
        ))/A^2/Mg(i);
    temp=sparse(blockdiag(temp,fy));
end

fy=zeros(NS-1,2);
for i=M:M
    A=R(i)^2+Xpd(i)*Xpq(i);
```

```
    fy(1)=-(Xd(i)-Xpd(i))*(R(i)*sin(theta(i))-Xpq(i)*cos(theta(i)))/A/Tpd0(i);
    fy(2)=-(Xq(i)-Xpq(i))*(R(i)*cos(theta(i))+Xpd(i)*sin(theta(i)))/A/Tpq0(i);
    fy(4)=-Ka(i)/Ta(i);
    fy(8)=-(2*R(i)*sin(theta(i))*Xpd(i)^2*Epd(i)-2*Xpq(i)*cos(theta(i))*Xpd(i)^2*v(i)*sin(the
        ta(i))+2*Xpd(i)*R(i)^2*sin(theta(i))*v(i)*cos(theta(i))+Xpd(i)*Xpq(i)*cos(theta(i))*R(
        i)*Epq(i)+2*R(i)*Xpd(i)^2*v(i)-2*R(i)*Xpd(i)^2*v(i)*cos(theta(i))^2-Xpq(i)*cos(theta(i
        ))*Xpd(i)^2*Epd(i)-2*Xpd(i)*R(i)^2*sin(theta(i))*Epq(i)-Epq(i)*R(i)^3*cos(theta(i))+Xp
        d(i)*R(i)^2*Epd(i)*cos(theta(i))-Xpd(i)*sin(theta(i))*Xpq(i)^2*Epq(i)+Epd(i)*R(i)^3*si
        n(theta(i))-2*Xpq(i)^2*cos(theta(i))*R(i)*Epq(i)+2*Xpq(i)^2*cos(theta(i))^2*R(i)*v(i)+
        2*Xpq(i)^2*cos(theta(i))*Xpd(i)*v(i)*sin(theta(i))-2*Xpq(i)*R(i)^2*cos(theta(i))*Epd(i
        )+Xpq(i)*R(i)^2*Epq(i)*sin(theta(i))-2*Xpq(i)*R(i)^2*sin(theta(i))*v(i)*cos(theta(i))-
        Xpq(i)*R(i)*sin(theta(i))*Xpd(i)*Epd(i)-2*Xpq(i)*R(i)*Xpd(i)*v(i))/A^2/Mg(i);
    fy(9)=-(Xd(i)-Xpd(i))*v(i)*(R(i)*cos(theta(i))+Xpq(i)*sin(theta(i)))/A/Tpd0(i);
    fy(10)=(Xq(i)-Xpq(i))*v(i)*(R(i)*sin(theta(i))-Xpd(i)*cos(theta(i)))/A/Tpq0(i);
    fy(16)=-v(i)*(Epd(i)*R(i)^3*cos(theta(i))+Epq(i)*R(i)^3*sin(theta(i))+2*R(i)*cos(theta(i)
        )*Xpd(i)^2*Epd(i)+Xpq(i)*sin(theta(i))*Xpd(i)^2*Epd(i)-2*Xpd(i)*R(i)^2*cos(theta(i))*E
        pq(i)+2*Xpd(i)*R(i)^2*cos(theta(i))^2*v(i)-Xpd(i)*R(i)^2*Epd(i)*sin(theta(i))+2*R(i)*c
        os(theta(i))*Xpd(i)^2*v(i)*sin(theta(i))-Xpd(i)*Xpq(i)*sin(theta(i))*R(i)*Epq(i)-2*Xpd
        (i)^2*Xpq(i)*v(i)*cos(theta(i))^2+2*Xpq(i)^2*sin(theta(i))*R(i)*Epq(i)-Xpd(i)*cos(thet
        a(i))*Xpq(i)^2*Epq(i)+2*Xpd(i)*cos(theta(i))^2*Xpq(i)^2*v(i)+2*Xpq(i)*R(i)^2*sin(theta
        (i))*Epd(i)+Xpq(i)*R(i)^2*Epq(i)*cos(theta(i))-2*Xpq(i)*R(i)^2*v(i)*cos(theta(i))^2-2*
        Xpq(i)^2*sin(theta(i))*R(i)*v(i)*cos(theta(i))-Xpq(i)*R(i)*cos(theta(i))*Xpd(i)*Epd(i)
        +Xpq(i)*Xpd(i)^2*v(i)-Xpd(i)*R(i)^2*v(i)+Xpq(i)*R(i)^2*v(i)-Xpq(i)^2*Xpd(i)*v(i))/A^2/
        Mg(i);
    temp=sparse(blockdiag(temp,fy));
end

Fy=sparse([temp zeros(NS*M-1,2*(N-M))]);

%%%%%%%%%%%%%%%%%%getFa%%%%%%%%%%%%%%%%%%%%%%%%
Fa=sparse(zeros(NS*M-1,1));
Fa(NS*[0:M-1]+7)=Pg0.*Km./Tg;

%%%%%%%%%%%%%%%%%%getGx%%%%%%%%%%%%%%%%%%%%%%%%
Gx=sparse(2*N,NS*M-1);
gx=zeros(2,NS);
temp=[];

for i=1:M-1
   A=R(i)^2+Xpd(i)*Xpq(i);

    gx(1)=v(i)*(Xpq(i)*sin(delta(i)-theta(i))+R(i)*cos(delta(i)-theta(i)))/A;
    gx(2)=v(i)*(Xpq(i)*cos(delta(i)-theta(i))-R(i)*sin(delta(i)-theta(i)))/A;
    gx(3)=-v(i)*(-R(i)*sin(delta(i)-theta(i))+Xpd(i)*cos(delta(i)-theta(i)))/A;
    gx(4)=v(i)*(R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-theta(i)))/A;
    gx(17)=v(i)*(v(i)*Xpq(i)-2*Xpq(i)*v(i)*cos(delta(i)-theta(i))^2+cos(delta(i)-theta(i))*R(
        i)*Epd(i)+cos(delta(i)-theta(i))*Xpq(i)*Epq(i)+2*v(i)*cos(delta(i)-theta(i))^2*Xpd(i)-
        sin(delta(i)-theta(i))*R(i)*Epq(i)+sin(delta(i)-theta(i))*Xpd(i)*Epd(i)-Xpd(i)*v(i))/A
        ;
    gx(18)=v(i)*(2*v(i)*cos(delta(i)-theta(i))*Xpq(i)*sin(delta(i)-theta(i))-sin(delta(i)-the
        ta(i))*R(i)*Epd(i)-sin(delta(i)-theta(i))*Xpq(i)*Epq(i)-2*v(i)*sin(delta(i)-theta(i))*
        Xpd(i)*cos(delta(i)-theta(i))-cos(delta(i)-theta(i))*R(i)*Epq(i)+cos(delta(i)-theta(i)
```

```
        )*Xpd(i)*Epd(i))/A;
    temp=sparse(blockdiag(temp,gx));
end

% machine M
gx=zeros(2,NS-1);
for i=M:M
    A=R(i)^2+Xpd(i)*Xpq(i);

    gx(1)=-v(i)*(Xpq(i)*sin(theta(i))-R(i)*cos(theta(i)))/A;
    gx(2)=v(i)*(Xpq(i)*cos(theta(i))+R(i)*sin(theta(i)))/A;
    gx(3)=-v(i)*(R(i)*sin(theta(i))+Xpd(i)*cos(theta(i)))/A;
    gx(4)=-v(i)*(-R(i)*cos(theta(i))+Xpd(i)*sin(theta(i)))/A;
    temp=sparse(blockdiag(temp,gx));
end

Gx=[temp;zeros(2*(N-M),NS*M-1)];
Gx(2*N*(NS*M-2)+(1:2:2*N))= -Pl0.*(Kpp.*v0.^2+v.*Kip.*v0+v.^2.*Kzp).*Kw.*(1+alpha*Kpl)./v0.^2;
Gx(2*N*(NS*M-2)+1+(1:2:2*N))=-Ql0.*(Kpq.*v0.^2+v.*Kiq.*v0+v.^2.*Kzq).*Kw.*(1+alpha*Kql)./v0.^
    2;

%%%%%%%%%%%%%%%%%%%%%%%%getGy%%%%%%%%%%%%%%%%%%%%
Gy=sparse(2*N,2*N);
for i=1:N
    for k=1:N
        if k==i
            if i<=M
                A=R(i)^2+Xpd(i)*Xpq(i);

                    gy1(k,i)=-sum(v.*Ybus_abs(i,:).*cos(theta(i)-Ybus_angle(i,:)-theta),2)-
                        v(i)*Ybus_abs(i,k)*cos(-Ybus_angle(i,k)) + ...
                    R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(d
                        elta(i)-theta(i)))/A*cos(delta(i)-theta(i)) + ...
                    (R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(
                        delta(i)-theta(i)))/A*sin(delta(i)-theta(i)) + ...
                    (-R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-theta(i)))/A*v(i)*cos(delt
                        a(i)-theta(i)) + ...
                    (-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-theta(i)))/A*v(i)*sin(delt
                        a(i)-theta(i)) + ...
                    (-Pl0(i)*(1/v0(i)*Kip(i)+2*v(i)/v0(i)^2*Kzp(i))*(1+Kw(i)*(omega(M)-1))*(1+al
                        pha*Kpl(i)));

                    gy2(k,i)=-sum(v.*Ybus_abs(i,:).*sin(theta(i)-Ybus_angle(i,:)-theta),2)-
                        v(i)*Ybus_abs(i,k)*sin(-Ybus_angle(i,k)) + ...
                    (-(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*si
                        n(delta(i)-theta(i)))/A*sin(delta(i)-theta(i))) + ...
                    (R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(
                        delta(i)-theta(i)))/A*cos(delta(i)-theta(i)) + ...
                    (-(-R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-theta(i)))/A*v(i)*sin(de
                        lta(i)-theta(i))) + ...
                    (-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-theta(i)))/A*v(i)*cos(delt
                        a(i)-theta(i)) + ...
```

```
            (-Ql0(i)*(1/v0(i)*Kiq(i)+2*v(i)/v0(i)^2*Kzq(i))*(1+Kw(i)*(omega(M)-1))*(1+al
               pha*Kql(i))) ;

            gy3(k,i)=v(i)*( sum(v.*Ybus_abs(i,:).*sin(theta(i)- Ybus_angle(i,:)-theta),2)
               - v(i)*Ybus_abs(i,k)*sin(-Ybus_angle(i,k))) + ...
            (R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(
               delta(i)-theta(i)))/A*v(i)*sin(delta(i)-theta(i)) + ...
            (-(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*co
               s(delta(i)-theta(i)))/A*v(i)*cos(delta(i)-theta(i))) + ...
            (R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin(delta(i)-theta(i)))/A*v(i)
               *sin(delta(i)-theta(i)) + ...
            (-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos(delta(i)-theta(i)))/A*v(i
               )*cos(delta(i)-theta(i)) ;

            gy4(k,i)=-v(i)*( sum(v.*Ybus_abs(i,:).*cos(theta(i)- Ybus_angle(i,:)-theta),2)
               - v(i)*Ybus_abs(i,k)*cos(-Ybus_angle(i,k))) + ...
            (R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(
               delta(i)-theta(i)))/A*v(i)*sin(delta(i)-theta(i)) + ...
            (R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(
               delta(i)-theta(i)))/A*v(i)*cos(delta(i)-theta(i)) + ...
            (R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin(delta(i)-theta(i)))/A*v(i)
               *cos(delta(i)-theta(i)) + ...
            (-(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos(delta(i)-theta(i)))/A*v
               (i)*sin(delta(i)-theta(i))) ;
    else

            gy1(k,i)= -sum(v.*Ybus_abs(i,:).*cos(theta(i)-Ybus_angle(i,:)-theta),2)-
               v(i)*Ybus_abs(i,k)*cos(-Ybus_angle(i,k)) + ...
            (-Pl0(i)*(1/v0(i)*Kip(i)+2*v(i)/v0(i)^2*Kzp(i))*(1+Kw(i)*(omega(M)-1))*(1+al
               pha*Kpl(i)));

            gy2(k,i)=-sum(v.*Ybus_abs(i,:).*sin(theta(i)-Ybus_angle(i,:)-theta),2)-
               v(i)*Ybus_abs(i,k)*sin(-Ybus_angle(i,k)) + ...
            (-Ql0(i)*(1/v0(i)*Kiq(i)+2*v(i)/v0(i)^2*Kzq(i))*(1+Kw(i)*(omega(M)-1))*(1+al
               pha*Kql(i))) ;

            gy3(k,i)=v(i)*( sum(v.*Ybus_abs(i,:).*sin(theta(i)- Ybus_angle(i,:)-theta),2)
               - v(i)*Ybus_abs(i,k)*sin(-Ybus_angle(i,k)));

            gy4(k,i)=-v(i)*( sum(v.*Ybus_abs(i,:).*cos(theta(i)- Ybus_angle(i,:)-theta),2)
               - v(i)*Ybus_abs(i,k)*cos(-Ybus_angle(i,k)));
    end
else
      gy1(k,i)=-v(k)*Ybus_abs(k,i)*cos(-theta(k)+Ybus_angle(k,i)+theta(i));
      gy2(k,i)=v(k)*Ybus_abs(k,i)*sin(-theta(k)+Ybus_angle(k,i)+theta(i));
      gy3(k,i)=v(k)*v(i)*Ybus_abs(k,i)*sin(-theta(k)+Ybus_angle(k,i)+theta(i));
      gy4(k,i)=v(k)*v(i)*Ybus_abs(k,i)*cos(-theta(k)+Ybus_angle(k,i)+theta(i));
end

Gy(2*(k-1)+1,2*(i-1)+1)=gy1(k,i);
Gy(2*k,2*(i-1)+1)=gy2(k,i);
Gy(2*(k-1)+1,2*i)=gy3(k,i);
```

```
        Gy(2*k,2*i)=gy4(k,i);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%getGa%%%%%%%%%%%%%%%%%%%%%%%%%
Ga=zeros(2*N,1);
Ga(1+2*[0:N-1]) =-Pl0.*(Kpp+v./v0.*Kip+v.^2./v0.^2.*Kzp).*(1+Kw.*(omega(M)-1)).*Kpl;
Ga(2*[1:N])     =-Ql0.*(Kpq+v./v0.*Kiq+v.^2./v0.^2.*Kzq).*(1+Kw.*(omega(M)-1)).*Kql;
```

A.2.2 Second derivatives - Fxx

```
function Fxx=getFxx(x,Sys,Machines,Governors,Excitors)
M=Sys(1); N=Sys(2); NS=Sys(3);
% note: all the vectors are row vectors.
Epq=x(1:NS:NS*(M-1)+1); Epd=x(2:NS:NS*(M-1)+ 2); Efd=x(3:NS:NS*(M-1)+3);
Vr=x(4:NS:NS*(M-1)+ 4); Rf=x(5:NS:NS*(M-1)+ 5); Pm=x(6:NS:NS*(M-1)+6);
Miu=x(7:NS:NS*(M-1)+7); omega=x(NS-1:NS:NS*(M-1)+ NS-1);
delta=[x(NS:NS:NS*(M-1)) 0]; v=x(NS*M+2*[0:N-1]); theta=x(NS*M+1+2*[0:N-1]);
%%%%%%%%%machine%%%%%%%%%%%%%%%%
R=Machines(1,:); Xd=Machines(2,:); Xq=Machines(3,:); Xpd=Machines(4,:);
Xpq=Machines(5,:); Tpd0=Machines(6,:); Tpq0=Machines(7,:);
Mg=Machines(8,:); D=Machines(9,:); Pg0=Machines(10,:); Km=Machines(11,:);
%%%%%%%%%governor%%%%%%%%%%%%%%%%
K1=Governors(1,:); Tg=Governors(2,:); Tch=Governors(3,:);
%%%%%%%%%%excitor%%%%%%%%%%%%%%%%
Ka=Excitors(1,:); Ta=Excitors(2,:); Ke=Excitors(3,:); Te=Excitors(4,:);
Se=Excitors(5,:); Kf=Excitors(6,:); Tf=Excitors(7,:); Vref=Excitors(8,:);

% for machine 1:M-1
kk=[8    8    8    8    8    8    8    8    1    2    8];
mm=[1    2    9    1    2    9    1    2    9    9    9];
nn=[1    1    1    2    2    2    9    9    9    9    9];
pos=kk+(mm-1)*(NS*M-1)+NS*(NS*M-1)*(nn-1);

temp=[];
for i=1:M-1
    fxx=sparse((NS*M-1)*NS,NS);
    A=R(i)^2+Xpd(i)*Xpq(i);
     fxx((i-1)*NS+pos(1))=(-2*(1-Xpd(i)*Xpq(i)/A)*R(i)/A-2*Xpq(i)^2*R(i)/A^2)/Mg(i);
     fxx((i-1)*NS+pos(2))=((1-Xpd(i)*Xpq(i)/A)*Xpd(i)/A+Xpd(i)*R(i)^2/A^2-Xpq(i)*R(i)^2/A^2-(1
         -Xpd(i)*Xpq(i)/A)*Xpq(i)/A)/Mg(i);
     fxx((i-1)*NS+pos(3))=(-(1-Xpd(i)*Xpq(i)/A)*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(i)*v(i)*
         cos(delta(i)-theta(i)))/A+Xpd(i)*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(de
         lta(i)-theta(i)))/A^2*R(i)-Xpq(i)*R(i)/A^2*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v
         (i)*sin(delta(i)-theta(i)))-Xpq(i)^2*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos
         (delta(i)-theta(i)))/A^2)/Mg(i);
     fxx((i-1)*NS+pos(4))=((1-Xpd(i)*Xpq(i)/A)*Xpd(i)/A+Xpd(i)*R(i)^2/A^2-Xpq(i)*R(i)^2/A^2-(1
         -Xpd(i)*Xpq(i)/A)*Xpq(i)/A)/Mg(i);
     fxx((i-1)*NS+pos(5))=(-2*Xpd(i)^2*R(i)/A^2-2*(1-Xpd(i)*Xpq(i)/A)*R(i)/A)/Mg(i);
     fxx((i-1)*NS+pos(6))=(Xpd(i)*R(i)/A^2*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos(d
         elta(i)-theta(i)))-Xpd(i)^2*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(delta(i
```

```
        )-theta(i)))/A^2-(1-Xpd(i)*Xpq(i)/A)*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*si
        n(delta(i)-theta(i)))/A-Xpq(i)*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos(delta
        (i)-theta(i)))/A^2*R(i))/Mg(i);
     fxx((i-1)*NS+pos(7))=(-(1-Xpd(i)*Xpq(i)/A)*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(i)*v(i)*
        cos(delta(i)-theta(i)))/A+Xpd(i)*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(de
        lta(i)-theta(i)))/A^2*R(i)-Xpq(i)*R(i)/A^2*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v
        (i)*sin(delta(i)-theta(i)))-Xpq(i)^2*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos
        (delta(i)-theta(i)))/A^2)/Mg(i);
     fxx((i-1)*NS+pos(8))=(Xpd(i)*R(i)/A^2*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos(d
        elta(i)-theta(i)))-Xpd(i)^2*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(delta(i
        )-theta(i)))/A^2-(1-Xpd(i)*Xpq(i)/A)*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*si
        n(delta(i)-theta(i)))/A-Xpq(i)*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos(delta
        (i)-theta(i)))/A^2*R(i))/Mg(i);
     fxx((i-1)*NS+pos(9))=-(Xd(i)-Xpd(i))*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpq(i)*v(i)*cos(de
        lta(i)-theta(i)))/A/Tpd0(i);
     fxx((i-1)*NS+pos(10))=(Xq(i)-Xpq(i))*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpd(i)*v(i)*sin(de
        lta(i)-theta(i)))/A/Tpq0(i);
     fxx((i-1)*NS+pos(11))=(Xpd(i)*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpq(i)*v(i)*cos(delta(i)-
        theta(i)))/A^2*(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)
        *sin(delta(i)-theta(i)))+2*Xpd(i)*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(d
        elta(i)-theta(i)))/A^2*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos(delta(i)-thet
        a(i)))-(Epq(i)-Xpd(i)*(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(
        i)*v(i)*cos(delta(i)-theta(i)))/A)*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpd(i)*v(i)*sin(d
        elta(i)-theta(i)))/A-Xpq(i)*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpd(i)*v(i)*sin(delta(i)
        -theta(i)))/A^2*(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i
        )*cos(delta(i)-theta(i)))-2*Xpq(i)*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos(d
        elta(i)-theta(i)))/A^2*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(delta(i)-the
        ta(i)))-(Epd(i)+Xpq(i)*(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd
        (i)*v(i)*sin(delta(i)-theta(i)))/A)*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpq(i)*v(i)*cos(
        delta(i)-theta(i)))/A)/Mg(i);
   temp=blockdiag(temp,fxx);
end

% for Machine M
kk=[8    8    8    8 ]; mm=[1    2    1    2 ]; nn=[1    1    2    2 ];
pos=kk+(mm-1)*(NS*M-1)+(NS-1)*(NS*M-1)*(nn-1);

for i=M:M
   fxx=sparse( (NS*M-1)*(NS-1),NS-1);
   A=R(i)^2+Xpd(i)*Xpq(i);

    fxx((i-1)*NS+pos(1))=(-2*(1-Xpd(i)*Xpq(i)/A)*R(i)/A-2*Xpq(i)^2*R(i)/A^2)/Mg(i);
    fxx((i-1)*NS+pos(2))=((1-Xpd(i)*Xpq(i)/A)*Xpd(i)/A+Xpd(i)*R(i)^2/A^2-Xpq(i)*R(i)^2/A^2-(1
       -Xpd(i)*Xpq(i)/A)*Xpq(i)/A)/Mg(i);
    fxx((i-1)*NS+pos(3))=((1-Xpd(i)*Xpq(i)/A)*Xpd(i)/A+Xpd(i)*R(i)^2/A^2-Xpq(i)*R(i)^2/A^2-(1
       -Xpd(i)*Xpq(i)/A)*Xpq(i)/A)/Mg(i);
    fxx((i-1)*NS+pos(4))=(-2*Xpd(i)^2*R(i)/A^2-2*(1-Xpd(i)*Xpq(i)/A)*R(i)/A)/Mg(i);
   temp=blockdiag(temp,fxx);
end
Fxx=sparse(temp);
```

A.2.3 Second derivatives - Fxy

```
function Fxy=getFxy(x,Sys,Machines,Governors,Excitors)
M=Sys(1); N=Sys(2); NS=Sys(3);
% note: all the vectors are row vectors.
Epq=x(1:NS:NS*(M-1)+1); Epd=x(2:NS:NS*(M-1)+ 2); Efd=x(3:NS:NS*(M-1)+3);
Vr=x(4:NS:NS*(M-1)+ 4); Rf=x(5:NS:NS*(M-1)+ 5); Pm=x(6:NS:NS*(M-1)+6);
Miu=x(7:NS:NS*(M-1)+7); omega=x(NS-1:NS:NS*(M-1)+ NS-1);
delta=[x(NS:NS:NS*(M-1)) 0];

v=x(NS*M+2*[0:N-1]); theta=x(NS*M+1+2*[0:N-1]);
%%%%%%%%%%machine%%%%%%%%%%%%%%%%%%
R=Machines(1,:); Xd=Machines(2,:); Xq=Machines(3,:); Xpd=Machines(4,:);
Xpq=Machines(5,:); Tpd0=Machines(6,:); Tpq0=Machines(7,:); Mg=Machines(8,:);
D=Machines(9,:); Pg0=Machines(10,:); Km=Machines(11,:);
%%%%%%%%%%governor%%%%%%%%%%%%%%%%%%
K1=Governors(1,:); Tg=Governors(2,:); Tch=Governors(3,:);
%%%%%%%%%%%%excitor%%%%%%%%%%%%%%%%%%
Ka=Excitors(1,:); Ta=Excitors(2,:); Ke=Excitors(3,:); Te=Excitors(4,:);
Se=Excitors(5,:); Kf=Excitors(6,:); Tf=Excitors(7,:); Vref=Excitors(8,:);

% for machine 1 to M-1
kk=[8    8    1    2    8    8    8    1    2    8];
mm=[1    2    9    9    9    1    2    9    9    9];
nn=[1    1    1    1    1    2    2    2    2    2];
pos=kk+(mm-1)*(NS*M-1)+NS*(NS*M-1)*(nn-1);

temp=[];
for i=1:M-1
    fxy=sparse((NS*M-1)*NS,2);
    A=R(i)^2+Xpd(i)*Xpq(i);
     fxy((i-1)*NS+pos(1))=(-(1-Xpd(i)*Xpq(i)/A)*(-R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta
         (i)-theta(i)))/A+Xpd(i)*(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-theta(i)))/A
         ^2*R(i)-Xpq(i)*R(i)/A^2*(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-theta(i)))-X
         pq(i)^2*(-R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-theta(i)))/A^2)/Mg(i);
     fxy((i-1)*NS+pos(2))=(Xpd(i)*R(i)/A^2*(-R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-t
         heta(i)))-Xpd(i)^2*(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-theta(i)))/A^2-(1
         -Xpd(i)*Xpq(i)/A)*(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-theta(i)))/A-Xpq(i
         )*(-R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-theta(i)))/A^2*R(i))/Mg(i);
     fxy((i-1)*NS+pos(3))=-(Xd(i)-Xpd(i))*(-R(i)*cos(delta(i)-theta(i))+Xpq(i)*sin(delta(i)-th
         eta(i)))/A/Tpd0(i);
     fxy((i-1)*NS+pos(4))=(Xq(i)-Xpq(i))*(R(i)*sin(delta(i)-theta(i))+Xpd(i)*cos(delta(i)-thet
         a(i)))/A/Tpq0(i);
     fxy((i-1)*NS+pos(5))=(Xpd(i)*(-R(i)*cos(delta(i)-theta(i))+Xpq(i)*sin(delta(i)-theta(i)))
         /A^2*(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta
         (i)-theta(i)))+Xpd(i)*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(delta(i)-thet
         a(i)))/A^2*(-R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-theta(i)))+Xpd(i)*(-R(i)*
         sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-theta(i)))/A^2*(R(i)*v(i)*sin(delta(i)-thet
         a(i))+Xpd(i)*v(i)*cos(delta(i)-theta(i)))-(Epq(i)-Xpd(i)*(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(
         i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)))/A)*(R(i)*sin(delta(
         i)-theta(i))+Xpd(i)*cos(delta(i)-theta(i)))/A-Xpq(i)*(R(i)*sin(delta(i)-theta(i))+Xpd(
```

```
        i)*cos(delta(i)-theta(i)))/A^2*(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)-theta
        (i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)))-Xpq(i)*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(
        i)*v(i)*cos(delta(i)-theta(i)))/A^2*(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-
        theta(i)))-Xpq(i)*(-R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-theta(i)))/A^2*(-R
        (i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(delta(i)-theta(i)))-(Epd(i)+Xpq(i)*(R(
        i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(i)-thet
        a(i)))/A)*(-R(i)*cos(delta(i)-theta(i))+Xpq(i)*sin(delta(i)-theta(i)))/A)/Mg(i);
    fxy((i-1)*NS+pos(6))=(-(1-Xpd(i)*Xpq(i)/A)*(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)
        *cos(delta(i)-theta(i)))/A+Xpd(i)*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin(de
        lta(i)-theta(i)))/A^2*R(i)-Xpq(i)*R(i)/A^2*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(
        i)*sin(delta(i)-theta(i)))-Xpq(i)^2*(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos
        (delta(i)-theta(i)))/A^2)/Mg(i);
    fxy((i-1)*NS+pos(7))=(Xpd(i)*R(i)/A^2*(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos(
        delta(i)-theta(i)))-Xpd(i)^2*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin(delta(i
        )-theta(i)))/A^2-(1-Xpd(i)*Xpq(i)/A)*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin
        (delta(i)-theta(i)))/A-Xpq(i)*(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos(delta
        (i)-theta(i)))/A^2*R(i))/Mg(i);
    fxy((i-1)*NS+pos(8))=-(Xd(i)-Xpd(i))*(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(d
        elta(i)-theta(i)))/A/Tpd0(i);
    fxy((i-1)*NS+pos(9))=(Xq(i)-Xpq(i))*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(de
        lta(i)-theta(i)))/A/Tpq0(i);
    fxy((i-1)*NS+pos(10))=(Xpd(i)*(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)
        -theta(i)))/A^2*(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i
        )*sin(delta(i)-theta(i)))+Xpd(i)*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(de
        lta(i)-theta(i)))/A^2*(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos(delta(i)-thet
        a(i)))+Xpd(i)*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin(delta(i)-theta(i)))/A^
        2*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos(delta(i)-theta(i)))-(Epq(i)-Xpd(i)
        *(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)-
        theta(i)))/A)*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(i)-theta(i)))/A
        -Xpq(i)*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(i)-theta(i)))/A^2*(R(
        i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)-thet
        a(i)))-Xpq(i)*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos(delta(i)-theta(i)))/A^
        2*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin(delta(i)-theta(i)))-Xpq(i)*(-R(i)*
        v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos(delta(i)-theta(i)))/A^2*(-R(i)*v(i)*cos(de
        lta(i)-theta(i))+Xpq(i)*v(i)*sin(delta(i)-theta(i)))-(Epd(i)+Xpq(i)*(R(i)*Epq(i)-Xpd(i
        )*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(i)-theta(i)))/A)*(-R(i
        )*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)))/A)/Mg(i);
  temp=blockdiag(temp,fxy);
end

% for Machine M
kk=[8    8    8    8 ]; mm=[1    2    1    2 ]; nn=[1    1    2    2 ];
pos=kk+(mm-1)*(NS*M-1)+(NS-1)*(NS*M-1)*(nn-1);

for i=M:M
  fxy=sparse( (NS*M-1)*(NS-1),2);
  A=R(i)^2+Xpd(i)*Xpq(i);

   fxy((i-1)*NS+pos(1))=(-(1-Xpd(i)*Xpq(i)/A)*(-R(i)*cos(theta(i))-Xpd(i)*sin(theta(i)))/A+X
      pd(i)*(R(i)*sin(theta(i))-Xpq(i)*cos(theta(i)))/A^2*R(i)-Xpq(i)*R(i)/A^2*(R(i)*sin(the
      ta(i))-Xpq(i)*cos(theta(i)))-Xpq(i)^2*(-R(i)*cos(theta(i))-Xpd(i)*sin(theta(i)))/A^2)/
      Mg(i);
```

```
    fxy((i-1)*NS+pos(2))=(Xpd(i)*R(i)/A^2*(-R(i)*cos(theta(i))-Xpd(i)*sin(theta(i)))-Xpd(i)^2
        *(R(i)*sin(theta(i))-Xpq(i)*cos(theta(i)))/A^2-(1-Xpd(i)*Xpq(i)/A)*(R(i)*sin(theta(i))
        -Xpq(i)*cos(theta(i)))/A-Xpq(i)*(-R(i)*cos(theta(i))-Xpd(i)*sin(theta(i)))/A^2*R(i))/M
        g(i);
    fxy((i-1)*NS+pos(3))=(-(1-Xpd(i)*Xpq(i)/A)*(R(i)*v(i)*sin(theta(i))-Xpd(i)*v(i)*cos(theta
        (i)))/A+Xpd(i)*(R(i)*v(i)*cos(theta(i))+Xpq(i)*v(i)*sin(theta(i)))/A^2*R(i)-Xpq(i)*R(i
        )/A^2*(R(i)*v(i)*cos(theta(i))+Xpq(i)*v(i)*sin(theta(i)))-Xpq(i)^2*(R(i)*v(i)*sin(thet
        a(i))-Xpd(i)*v(i)*cos(theta(i)))/A^2)/Mg(i);
    fxy((i-1)*NS+pos(4))=(Xpd(i)*R(i)/A^2*(R(i)*v(i)*sin(theta(i))-Xpd(i)*v(i)*cos(theta(i)))
        -Xpd(i)^2*(R(i)*v(i)*cos(theta(i))+Xpq(i)*v(i)*sin(theta(i)))/A^2-(1-Xpd(i)*Xpq(i)/A)*
        (R(i)*v(i)*cos(theta(i))+Xpq(i)*v(i)*sin(theta(i)))/A-Xpq(i)*(R(i)*v(i)*sin(theta(i))-
        Xpd(i)*v(i)*cos(theta(i)))/A^2*R(i))/Mg(i);
  temp=blockdiag(temp,fxy);
end

Fxy=sparse(zeros( (NS*M-1)^2,2*N));
Fxy( 1:(NS*M-1)^2,1:2*M)=temp;
```

A.2.4 Second derivatives - Fxa

```
function Fxa=getFxa(x,Sys)
M=Sys(1); N=Sys(2); NS=Sys(3);
Fxa=sparse(zeros((NS*M-1)^2,1));
```

A.2.5 Second derivatives - Fyx

```
function Fyx=getFyx(x,Sys,Machines,Governors,Excitors)

M=Sys(1); N=Sys(2); NS=Sys(3);
% note: all the vectors are row vectors.
Epq=x(1:NS:NS*(M-1)+1); Epd=x(2:NS:NS*(M-1)+ 2); Efd=x(3:NS:NS*(M-1)+3);
Vr=x(4:NS:NS*(M-1)+ 4); Rf=x(5:NS:NS*(M-1)+ 5); Pm=x(6:NS:NS*(M-1)+6);
Miu=x(7:NS:NS*(M-1)+7); omega=x(NS-1:NS:NS*(M-1)+ NS-1);
delta=[x(NS:NS:NS*(M-1)) 0];

v=x(NS*M+2*[0:N-1]); theta=x(NS*M+1+2*[0:N-1]);
%%%%%%%%%%machine%%%%%%%%%%%%%%%%%%%
R=Machines(1,:); Xd=Machines(2,:); Xq=Machines(3,:); Xpd=Machines(4,:);
Xpq=Machines(5,:); Tpd0=Machines(6,:); Tpq0=Machines(7,:); Mg=Machines(8,:);
D=Machines(9,:); Pg0=Machines(10,:); Km=Machines(11,:);
%%%%%%%%%%governor%%%%%%%%%%%%%%%%%%%
K1=Governors(1,:); Tg=Governors(2,:); Tch=Governors(3,:);
%%%%%%%%%%%%excitor%%%%%%%%%%%%%%%%%%
Ka=Excitors(1,:); Ta=Excitors(2,:);  Ke=Excitors(3,:); Te=Excitors(4,:);
Se=Excitors(5,:); Kf=Excitors(6,:); Tf=Excitors(7,:); Vref=Excitors(8,:);

% for machine 1 to M-1
kk=[8    8    8    1    2    8    1    2    8];
mm=[1    2    1    2    1    1    1    2    2    2];
nn=[1    1    2    2    9    9    9    9    9    9];
```

```
pos=kk+(mm-1)*(NS*M-1)+2*(NS*M-1)*(nn-1);

temp=[];
for i=1:M-1
    fxy=sparse((NS*M-1)*2,NS);
    A=R(i)^2+Xpd(i)*Xpq(i);
    fxy((i-1)*NS+pos(1))=(-(1-Xpd(i)*Xpq(i)/A)*(-R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta
        (i)-theta(i)))/A+Xpd(i)*(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-theta(i)))/A
        ^2*R(i)-Xpq(i)*R(i)/A^2*(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-theta(i)))-X
        pq(i)^2*(-R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-theta(i)))/A^2)/Mg(i);
    fxy((i-1)*NS+pos(2))=(-(1-Xpd(i)*Xpq(i)/A)*(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)
        *cos(delta(i)-theta(i)))/A+Xpd(i)*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin(de
        lta(i)-theta(i)))/A^2*R(i)-Xpq(i)*R(i)/A^2*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(
        i)*sin(delta(i)-theta(i)))-Xpq(i)^2*(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos
        (delta(i)-theta(i)))/A^2)/Mg(i);
    fxy((i-1)*NS+pos(3))=(Xpd(i)*R(i)/A^2*(-R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-t
        heta(i)))-Xpd(i)^2*(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-theta(i)))/A^2-(1
        -Xpd(i)*Xpq(i)/A)*(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-theta(i)))/A-Xpq(i
        )*(-R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-theta(i)))/A^2*R(i))/Mg(i);
    fxy((i-1)*NS+pos(4))=(Xpd(i)*R(i)/A^2*(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos(
        delta(i)-theta(i)))-Xpd(i)^2*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin(delta(i
        )-theta(i)))/A^2-(1-Xpd(i)*Xpq(i)/A)*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin
        (delta(i)-theta(i)))/A-Xpq(i)*(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos(delta
        (i)-theta(i)))/A^2*R(i))/Mg(i);
    fxy((i-1)*NS+pos(5))=-(Xd(i)-Xpd(i))*(-R(i)*cos(delta(i)-theta(i))+Xpq(i)*sin(delta(i)-th
        eta(i)))/A/Tpd0(i);
    fxy((i-1)*NS+pos(6))=(Xq(i)-Xpq(i))*(R(i)*sin(delta(i)-theta(i))+Xpd(i)*cos(delta(i)-thet
        a(i)))/A/Tpq0(i);
    fxy((i-1)*NS+pos(7))=(Xpd(i)*(-R(i)*cos(delta(i)-theta(i))+Xpq(i)*sin(delta(i)-theta(i)))
        /A^2*(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta
        (i)-theta(i)))+Xpd(i)*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(delta(i)-thet
        a(i)))/A^2*(-R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-theta(i)))+Xpd(i)*(-R(i)*
        sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-theta(i)))/A^2*(R(i)*v(i)*sin(delta(i)-thet
        a(i))+Xpd(i)*v(i)*cos(delta(i)-theta(i)))-(Epq(i)-Xpd(i)*(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(
        i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)))/A)*(R(i)*sin(delta(
        i)-theta(i))+Xpd(i)*cos(delta(i)-theta(i)))/A-Xpq(i)*(R(i)*sin(delta(i)-theta(i))+Xpd(
        i)*cos(delta(i)-theta(i)))/A^2*(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)-theta
        (i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)))-Xpq(i)*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(
        i)*v(i)*cos(delta(i)-theta(i)))/A^2*(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-
        theta(i)))-Xpq(i)*(-R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-theta(i)))/A^2*(-R
        (i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(delta(i)-theta(i)))-(Epd(i)+Xpq(i)*(R(
        i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(i)-thet
        a(i)))/A)*(-R(i)*cos(delta(i)-theta(i))+Xpq(i)*sin(delta(i)-theta(i)))/A)/Mg(i);
    fxy((i-1)*NS+pos(8))=-(Xd(i)-Xpd(i))*(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(d
        elta(i)-theta(i)))/A/Tpd0(i);
    fxy((i-1)*NS+pos(9))=(Xq(i)-Xpq(i))*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(de
        lta(i)-theta(i)))/A/Tpq0(i);
    fxy((i-1)*NS+pos(10))=(Xpd(i)*(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)
        -theta(i)))/A^2*(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i
        )*sin(delta(i)-theta(i)))+Xpd(i)*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(de
        lta(i)-theta(i)))/A^2*(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos(delta(i)-thet
        a(i)))+Xpd(i)*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin(delta(i)-theta(i)))/A^
```

```
        2*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos(delta(i)-theta(i)))-(Epq(i)-Xpd(i)
        *(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)-
        theta(i)))/A)*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(i)-theta(i)))/A
        -Xpq(i)*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(i)-theta(i)))/A^2*(R(
        i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)-thet
        a(i)))-Xpq(i)*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos(delta(i)-theta(i)))/A^
        2*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin(delta(i)-theta(i)))-Xpq(i)*(-R(i)*
        v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos(delta(i)-theta(i)))/A^2*(-R(i)*v(i)*cos(de
        lta(i)-theta(i))+Xpq(i)*v(i)*sin(delta(i)-theta(i)))-(Epd(i)+Xpq(i)*(R(i)*Epq(i)-Xpd(i
        )*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(i)-theta(i)))/A)*(-R(i
        )*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)))/A)/Mg(i);
    temp=blockdiag(temp,fxy);
end
% for Machine M
kk=[8    8    8    8]; mm=[1    2    1    2]; nn=[1    1    2    2];
pos=kk+(mm-1)*(NS*M-1)+2*(NS*M-1)*(nn-1);

for i=M:M
    fxy=sparse( (NS*M-1)*2,NS-1);
    A=R(i)^2+Xpd(i)*Xpq(i);

    fxy((i-1)*NS+pos(1))=(-(1-Xpd(i)*Xpq(i)/A)*(-R(i)*cos(theta(i))-Xpd(i)*sin(theta(i)))/A+X
        pd(i)*(R(i)*sin(theta(i))-Xpq(i)*cos(theta(i)))/A^2*R(i)-Xpq(i)*R(i)/A^2*(R(i)*sin(the
        ta(i))-Xpq(i)*cos(theta(i)))-Xpq(i)^2*(-R(i)*cos(theta(i))-Xpd(i)*sin(theta(i)))/A^2)/
        Mg(i);
    fxy((i-1)*NS+pos(2))=(-(1-Xpd(i)*Xpq(i)/A)*(R(i)*v(i)*sin(theta(i))-Xpd(i)*v(i)*cos(theta
        (i)))/A+Xpd(i)*(R(i)*v(i)*cos(theta(i))+Xpq(i)*v(i)*sin(theta(i)))/A^2*R(i)-Xpq(i)*R(i
        )/A^2*(R(i)*v(i)*cos(theta(i))+Xpq(i)*v(i)*sin(theta(i)))-Xpq(i)^2*(R(i)*v(i)*sin(thet
        a(i))-Xpd(i)*v(i)*cos(theta(i)))/A^2)/Mg(i);
    fxy((i-1)*NS+pos(3))=(Xpd(i)*R(i)/A^2*(-R(i)*cos(theta(i))-Xpd(i)*sin(theta(i)))-Xpd(i)^2
        *(R(i)*sin(theta(i))-Xpq(i)*cos(theta(i)))/A^2-(1-Xpd(i)*Xpq(i)/A)*(R(i)*sin(theta(i))
        -Xpq(i)*cos(theta(i)))/A-Xpq(i)*(-R(i)*cos(theta(i))-Xpd(i)*sin(theta(i)))/A^2*R(i))/M
        g(i);
    fxy((i-1)*NS+pos(4))=(Xpd(i)*R(i)/A^2*(R(i)*v(i)*sin(theta(i))-Xpd(i)*v(i)*cos(theta(i)))
        -Xpd(i)^2*(R(i)*v(i)*cos(theta(i))+Xpq(i)*v(i)*sin(theta(i)))/A^2-(1-Xpd(i)*Xpq(i)/A)*
        (R(i)*v(i)*cos(theta(i))+Xpq(i)*v(i)*sin(theta(i)))/A-Xpq(i)*(R(i)*v(i)*sin(theta(i))-
        Xpd(i)*v(i)*cos(theta(i)))/A^2*R(i))/Mg(i);
    temp=blockdiag(temp,fxy);
end

Fyx=sparse([temp; zeros( (NS*M-1)*2*(N-M),(NS*M-1))]);
```

A.2.6 Second derivatives - Fyy

```
function Fyy=getFyy(x,Sys,Machines,Governors,Excitors)

M=Sys(1); N=Sys(2); NS=Sys(3);
% note: all the vectors are row vectors.
Epq=x(1:NS:NS*(M-1)+1); Epd=x(2:NS:NS*(M-1)+ 2); Efd=x(3:NS:NS*(M-1)+3);
Vr=x(4:NS:NS*(M-1)+ 4); Rf=x(5:NS:NS*(M-1)+ 5); Pm=x(6:NS:NS*(M-1)+6);
Miu=x(7:NS:NS*(M-1)+7); omega=x(NS-1:NS:NS*(M-1)+ NS-1);
```

```
delta=[x(NS:NS:NS*(M-1)) 0];

v=x(NS*M+2*[0:N-1]); theta=x(NS*M+1+2*[0:N-1]);
%%%%%%%%%%machine%%%%%%%%%%%%%%%%%%
R=Machines(1,:); Xd=Machines(2,:); Xq=Machines(3,:); Xpd=Machines(4,:);
Xpq=Machines(5,:); Tpd0=Machines(6,:); Tpq0=Machines(7,:);
Mg=Machines(8,:); D=Machines(9,:); Pg0=Machines(10,:);
Km=Machines(11,:);
%%%%%%%%%%governor%%%%%%%%%%%%%%%%%%
K1=Governors(1,:); Tg=Governors(2,:); Tch=Governors(3,:);
%%%%%%%%%%%%excitor%%%%%%%%%%%%%%%%%%
Ka=Excitors(1,:); Ta=Excitors(2,:); Ke=Excitors(3,:); Te=Excitors(4,:);
Se=Excitors(5,:); Kf=Excitors(6,:); Tf=Excitors(7,:); Vref=Excitors(8,:);

% for machine 1 to M
kk=[    8    1    2    8    1    2    8    1    2    8];
mm=[    1    2    2    2    1    1    1    2    2    2];
nn=[    1    1    1    1    2    2    2    2    2    2];
pos=kk+(mm-1)*(NS*M-1)+2*(NS*M-1)*(nn-1);

temp=[];
for i=1:M
    fyy=sparse((NS*M-1)*2,2);
    A=R(i)^2+Xpd(i)*Xpq(i);
    fyy((i-1)*NS+pos(1))=(2*Xpd(i)*(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-theta(i)
        ))/A^2*(-R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-theta(i)))-2*Xpq(i)*(-R(i)*co
        s(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-theta(i)))/A^2*(-R(i)*sin(delta(i)-theta(i))-
        Xpq(i)*cos(delta(i)-theta(i))))/Mg(i);
    fyy((i-1)*NS+pos(2))=-(Xd(i)-Xpd(i))*(R(i)*cos(delta(i)-theta(i))-Xpq(i)*sin(delta(i)-the
        ta(i)))/A/Tpd0(i);
    fyy((i-1)*NS+pos(3))=(Xq(i)-Xpq(i))*(-R(i)*sin(delta(i)-theta(i))-Xpd(i)*cos(delta(i)-the
        ta(i)))/A/Tpq0(i);
    fyy((i-1)*NS+pos(4))=(Xpd(i)*(R(i)*cos(delta(i)-theta(i))-Xpq(i)*sin(delta(i)-theta(i)))/
        A^2*(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(
        i)-theta(i)))+Xpd(i)*(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-theta(i)))/A^2*
        (-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos(delta(i)-theta(i)))+Xpd(i)*(R(i)*v(
        i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin(delta(i)-theta(i)))/A^2*(-R(i)*cos(delta(i)-
        theta(i))+Xpd(i)*sin(delta(i)-theta(i)))-(Epq(i)-Xpd(i)*(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i
        )*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)))/A)*(-R(i)*sin(delta(
        i)-theta(i))-Xpd(i)*cos(delta(i)-theta(i)))/A-Xpq(i)*(-R(i)*sin(delta(i)-theta(i))-Xpd
        (i)*cos(delta(i)-theta(i)))/A^2*(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)-thet
        a(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)))-Xpq(i)*(-R(i)*cos(delta(i)-theta(i))+Xpd(i)*
        sin(delta(i)-theta(i)))/A^2*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin(delta(i)
        -theta(i)))-Xpq(i)*(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos(delta(i)-theta(i
        )))/A^2*(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-theta(i)))-(Epd(i)+Xpq(i)*(R
        (i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(i)-the
        ta(i)))/A)*(R(i)*cos(delta(i)-theta(i))-Xpq(i)*sin(delta(i)-theta(i)))/A)/Mg(i);
    fyy((i-1)*NS+pos(5))=-(Xd(i)-Xpd(i))*(R(i)*cos(delta(i)-theta(i))-Xpq(i)*sin(delta(i)-the
        ta(i)))/A/Tpd0(i);
    fyy((i-1)*NS+pos(6))=(Xq(i)-Xpq(i))*(-R(i)*sin(delta(i)-theta(i))-Xpd(i)*cos(delta(i)-the
        ta(i)))/A/Tpq0(i);
    fyy((i-1)*NS+pos(7))=(Xpd(i)*(R(i)*cos(delta(i)-theta(i))-Xpq(i)*sin(delta(i)-theta(i)))/
```

```
        A^2*(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(
        i)-theta(i)))+Xpd(i)*(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-theta(i)))/A^2*
        (-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos(delta(i)-theta(i)))+Xpd(i)*(R(i)*v(
        i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin(delta(i)-theta(i)))/A^2*(-R(i)*cos(delta(i)-
        theta(i))+Xpd(i)*sin(delta(i)-theta(i)))-(Epq(i)-Xpd(i)*(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i
        )*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)))/A)*(-R(i)*sin(delta(
        i)-theta(i))-Xpd(i)*cos(delta(i)-theta(i)))/A-Xpq(i)*(-R(i)*sin(delta(i)-theta(i))-Xpd
        (i)*cos(delta(i)-theta(i)))/A^2*(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)-thet
        a(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)))-Xpq(i)*(-R(i)*cos(delta(i)-theta(i))+Xpd(i)*
        sin(delta(i)-theta(i)))/A^2*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin(delta(i)
        -theta(i)))-Xpq(i)*(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos(delta(i)-theta(i
        )))/A^2*(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-theta(i)))-(Epd(i)+Xpq(i)*(R
        (i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(i)-the
        ta(i)))/A)*(R(i)*cos(delta(i)-theta(i))-Xpq(i)*sin(delta(i)-theta(i)))/A)/Mg(i);
    fyy((i-1)*NS+pos(8))=-(Xd(i)-Xpd(i))*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpq(i)*v(i)*cos(de
        lta(i)-theta(i)))/A/Tpd0(i);
    fyy((i-1)*NS+pos(9))=(Xq(i)-Xpq(i))*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpd(i)*v(i)*sin(del
        ta(i)-theta(i)))/A/Tpq0(i);
    fyy((i-1)*NS+pos(10))=(Xpd(i)*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpq(i)*v(i)*cos(delta(i)-
        theta(i)))/A^2*(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)
        *sin(delta(i)-theta(i)))+2*Xpd(i)*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin(de
        lta(i)-theta(i)))/A^2*(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos(delta(i)-thet
        a(i)))-(Epq(i)-Xpd(i)*(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(
        i)*v(i)*cos(delta(i)-theta(i)))/A)*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpd(i)*v(i)*sin(d
        elta(i)-theta(i)))/A-Xpq(i)*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpd(i)*v(i)*sin(delta(i)
        -theta(i)))/A^2*(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i
        )*cos(delta(i)-theta(i)))-2*Xpq(i)*(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos(
        delta(i)-theta(i)))/A^2*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin(delta(i)-the
        ta(i)))-(Epd(i)+Xpq(i)*(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd
        (i)*v(i)*sin(delta(i)-theta(i)))/A)*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpq(i)*v(i)*cos(
        delta(i)-theta(i)))/A)/Mg(i);
    temp=blockdiag(temp,fyy);
end

Fyy=sparse(zeros((NS*M-1)*2*N,2*N));
Fyy(1:(NS*M-1)*2*M,1:2*M)=temp;
```

A.2.7 Second drivatives - Fya

```
function Fya=getFya(x,Sys)
M=Sys(1); N=Sys(2); NS=Sys(3);
Fya=sparse(zeros((NS*M-1)*2*N,1));
```

A.2.8 Second drivatives - Gxx

```
function Gxx=getGxx(x,Sys,Machines,Governors,Excitors)

M=Sys(1); N=Sys(2); NS=Sys(3);
% note: all the vectors are row vectors.
Epq=x(1:NS:NS*(M-1)+1); Epd=x(2:NS:NS*(M-1)+ 2); Efd=x(3:NS:NS*(M-1)+3);
Vr=x(4:NS:NS*(M-1)+ 4); Rf=x(5:NS:NS*(M-1)+ 5); Pm=x(6:NS:NS*(M-1)+6);
```

```
Miu=x(7:NS:NS*(M-1)+7); omega=x(NS-1:NS:NS*(M-1)+ NS-1);
delta=[x(NS:NS:NS*(M-1)) 0];

v=x(NS*M+2*[0:N-1]); theta=x(NS*M+1+2*[0:N-1]);
%%%%%%%%%%machine%%%%%%%%%%%%%%%%%%
R=Machines(1,:); Xd=Machines(2,:); Xq=Machines(3,:); Xpd=Machines(4,:);
Xpq=Machines(5,:); Tpd0=Machines(6,:); Tpq0=Machines(7,:);
Mg=Machines(8,:); D=Machines(9,:); Pg0=Machines(10,:);
Km=Machines(11,:);
%%%%%%%%%%governor%%%%%%%%%%%%%%%%%%
K1=Governors(1,:); Tg=Governors(2,:); Tch=Governors(3,:);
%%%%%%%%%%%%excitor%%%%%%%%%%%%%%%%%
Ka=Excitors(1,:); Ta=Excitors(2,:); Ke=Excitors(3,:); Te=Excitors(4,:);
Se=Excitors(5,:); Kf=Excitors(6,:); Tf=Excitors(7,:); Vref=Excitors(8,:);

% for machine 1 to M-1
kk=[    1    2    1    2    1    2    1    2    1    2];
mm=[    9    9    9    9    1    1    2    2    9    9];
nn=[    1    1    2    2    9    9    9    9    9    9];
pos=kk+(mm-1)*(2*N)+ NS*(2*N)*(nn-1);

temp=[];
for i=1:M-1
    fxx=sparse(2*N*NS,NS);
    A=R(i)^2+Xpd(i)*Xpq(i);
    fxx((i-1)*2+pos(1))=Xpq(i)/A*v(i)*cos(delta(i)-theta(i))-R(i)/A*v(i)*sin(delta(i)-theta(i
        ));
    fxx((i-1)*2+pos(2))=-Xpq(i)/A*v(i)*sin(delta(i)-theta(i))-R(i)/A*v(i)*cos(delta(i)-theta(
        i));
    fxx((i-1)*2+pos(3))=R(i)/A*v(i)*cos(delta(i)-theta(i))+Xpd(i)/A*v(i)*sin(delta(i)-theta(i
        ));
    fxx((i-1)*2+pos(4))=-R(i)/A*v(i)*sin(delta(i)-theta(i))+Xpd(i)/A*v(i)*cos(delta(i)-theta(
        i));
    fxx((i-1)*2+pos(5))=Xpq(i)/A*v(i)*cos(delta(i)-theta(i))-R(i)/A*v(i)*sin(delta(i)-theta(i
        ));
    fxx((i-1)*2+pos(6))=-Xpq(i)/A*v(i)*sin(delta(i)-theta(i))-R(i)/A*v(i)*cos(delta(i)-theta(
        i));
    fxx((i-1)*2+pos(7))=R(i)/A*v(i)*cos(delta(i)-theta(i))+Xpd(i)/A*v(i)*sin(delta(i)-theta(i
        ));
    fxx((i-1)*2+pos(8))=-R(i)/A*v(i)*sin(delta(i)-theta(i))+Xpd(i)/A*v(i)*cos(delta(i)-theta(
        i));
    fxx((i-1)*2+pos(9))=(R(i)*v(i)*sin(delta(i)-theta(i))+Xpq(i)*v(i)*cos(delta(i)-theta(i)))
        /A*v(i)*sin(delta(i)-theta(i))+2*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(de
        lta(i)-theta(i)))/A*v(i)*cos(delta(i)-theta(i))-(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*s
        in(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)))/A*v(i)*sin(delta(i)-theta(i)
        )+(R(i)*v(i)*cos(delta(i)-theta(i))-Xpd(i)*v(i)*sin(delta(i)-theta(i)))/A*v(i)*cos(del
        ta(i)-theta(i))-2*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos(delta(i)-theta(i))
        )/A*v(i)*sin(delta(i)-theta(i))-(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-thet
        a(i))+Xpd(i)*v(i)*sin(delta(i)-theta(i)))/A*v(i)*cos(delta(i)-theta(i));
    fxx((i-1)*2+pos(10))=(R(i)*v(i)*sin(delta(i)-theta(i))+Xpq(i)*v(i)*cos(delta(i)-theta(i))
        )/A*v(i)*cos(delta(i)-theta(i))-2*(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(d
        elta(i)-theta(i)))/A*v(i)*sin(delta(i)-theta(i))-(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*
```

```
        sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)))/A*v(i)*cos(delta(i)-theta(i
        ))-(R(i)*v(i)*cos(delta(i)-theta(i))-Xpd(i)*v(i)*sin(delta(i)-theta(i)))/A*v(i)*sin(de
        lta(i)-theta(i))-2*(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos(delta(i)-theta(i)
        ))/A*v(i)*cos(delta(i)-theta(i))+(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-the
        ta(i))+Xpd(i)*v(i)*sin(delta(i)-theta(i)))/A*v(i)*sin(delta(i)-theta(i));
    temp=blockdiag(temp,fxx);
end

Gxx=sparse(zeros(2*N*(NS*M-1),NS*M-1));
Gxx(1:2*N*NS*(M-1), 1:NS*(M-1))=temp;
```

A.2.9 Second derivatives - Gxy

```
function Gxy=getGxy(x,Sys,Machines,Governors,Excitors,Loads,alpha)

M=Sys(1); N=Sys(2); NS=Sys(3);
% note: all the vectors are row vectors.
Epq=x(1:NS:NS*(M-1)+1); Epd=x(2:NS:NS*(M-1)+ 2); Efd=x(3:NS:NS*(M-1)+3);
Vr=x(4:NS:NS*(M-1)+ 4); Rf=x(5:NS:NS*(M-1)+ 5); Pm=x(6:NS:NS*(M-1)+6);
Miu=x(7:NS:NS*(M-1)+7); omega=x(NS-1:NS:NS*(M-1)+ NS-1);
delta=[x(NS:NS:NS*(M-1)) 0];

v=x(NS*M+2*[0:N-1]); theta=x(NS*M+1+2*[0:N-1]);
%%%%%%%%%%machine%%%%%%%%%%%%%%%%%%
R=Machines(1,:); Xd=Machines(2,:);  Xq=Machines(3,:); Xpd=Machines(4,:);
Xpq=Machines(5,:); Tpd0=Machines(6,:); Tpq0=Machines(7,:);
Mg=Machines(8,:); D=Machines(9,:); Pg0=Machines(10,:); Km=Machines(11,:);
%%%%%%%%%%governor%%%%%%%%%%%%%%%%%%
K1=Governors(1,:); Tg=Governors(2,:); Tch=Governors(3,:);
%%%%%%%%%%%%excitor%%%%%%%%%%%%%%%%%%
Ka=Excitors(1,:); Ta=Excitors(2,:); Ke=Excitors(3,:); Te=Excitors(4,:);
Se=Excitors(5,:); Kf=Excitors(6,:); Tf=Excitors(7,:); Vref=Excitors(8,:);
%%%%%%%%%%%%load%%%%%%%%%%%%%%%%%%%%
v0=Loads(1,:); Pl0=Loads(2,:); Kpp=Loads(3,:); Kip=Loads(4,:);
Kzp=Loads(5,:); Kpl=Loads(6,:); Ql0=Loads(7,:); Kpq=Loads(8,:);
Kiq=Loads(9,:); Kzq=Loads(10,:); Kql=Loads(11,:);

Kw=Loads(12,:);

% for machine 1:M-1
kk=[    1    2    1    2    1    2    1    2    1    2    1    2];
mm=[    1    1    2    2    9    9    1    1    2    2    9    9];
nn=[    1    1    1    1    1    1    2    2    2    2    2    2];
pos=kk+(mm-1)*(2*N)+ NS*(2*N)*(nn-1);

temp=[];
for i=1:M-1
    gxy=sparse(2*N*NS,2);
    A=R(i)^2+Xpd(i)*Xpq(i);
    gxy((i-1)*2+pos(1))=Xpq(i)/A*sin(delta(i)-theta(i))+R(i)/A*cos(delta(i)-theta(i));
    gxy((i-1)*2+pos(2))=Xpq(i)/A*cos(delta(i)-theta(i))-R(i)/A*sin(delta(i)-theta(i));
```

```
gxy((i-1)*2+pos(3))=R(i)/A*sin(delta(i)-theta(i))-Xpd(i)/A*cos(delta(i)-theta(i));
gxy((i-1)*2+pos(4))=R(i)/A*cos(delta(i)-theta(i))+Xpd(i)/A*sin(delta(i)-theta(i));
gxy((i-1)*2+pos(5))=(-R(i)*cos(delta(i)-theta(i))+Xpq(i)*sin(delta(i)-theta(i)))/A*v(i)*s
    in(delta(i)-theta(i))+(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(delta(i)-thet
    a(i)))/A*sin(delta(i)-theta(i))+(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-thet
    a(i)))/A*v(i)*cos(delta(i)-theta(i))+(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)
    -theta(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)))/A*cos(delta(i)-theta(i))+(R(i)*sin(delt
    a(i)-theta(i))+Xpd(i)*cos(delta(i)-theta(i)))/A*v(i)*cos(delta(i)-theta(i))+(R(i)*v(i)
    *sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos(delta(i)-theta(i)))/A*cos(delta(i)-theta(i))-(
    -R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-theta(i)))/A*v(i)*sin(delta(i)-theta(
    i))-(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(
    i)-theta(i)))/A*sin(delta(i)-theta(i));
gxy((i-1)*2+pos(6))=(-R(i)*cos(delta(i)-theta(i))+Xpq(i)*sin(delta(i)-theta(i)))/A*v(i)*c
    os(delta(i)-theta(i))+(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(delta(i)-thet
    a(i)))/A*cos(delta(i)-theta(i))-(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-thet
    a(i)))/A*v(i)*sin(delta(i)-theta(i))-(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)
    -theta(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)))/A*sin(delta(i)-theta(i))-(R(i)*sin(delt
    a(i)-theta(i))+Xpd(i)*cos(delta(i)-theta(i)))/A*v(i)*sin(delta(i)-theta(i))-(R(i)*v(i)
    *sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos(delta(i)-theta(i)))/A*sin(delta(i)-theta(i))-(
    -R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-theta(i)))/A*v(i)*cos(delta(i)-theta(
    i))-(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(
    i)-theta(i)))/A*cos(delta(i)-theta(i));
gxy((i-1)*2+pos(7))=-Xpq(i)/A*v(i)*cos(delta(i)-theta(i))+R(i)/A*v(i)*sin(delta(i)-theta(
    i));
gxy((i-1)*2+pos(8))=Xpq(i)/A*v(i)*sin(delta(i)-theta(i))+R(i)/A*v(i)*cos(delta(i)-theta(i
    ));
gxy((i-1)*2+pos(9))=-R(i)/A*v(i)*cos(delta(i)-theta(i))-Xpd(i)/A*v(i)*sin(delta(i)-theta(
    i));
gxy((i-1)*2+pos(10))=R(i)/A*v(i)*sin(delta(i)-theta(i))-Xpd(i)/A*v(i)*cos(delta(i)-theta(
    i));
gxy((i-1)*2+pos(11))=(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)
    ))/A*v(i)*sin(delta(i)-theta(i))-(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(de
    lta(i)-theta(i)))/A*v(i)*cos(delta(i)-theta(i))+(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(
    i)*v(i)*sin(delta(i)-theta(i)))/A*v(i)*cos(delta(i)-theta(i))+(R(i)*Epd(i)+Xpq(i)*Epq(
    i)-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)))/A*v(i)*sin(del
    ta(i)-theta(i))+(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(i)-theta(i)))
    /A*v(i)*cos(delta(i)-theta(i))+(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos(delta
    (i)-theta(i)))/A*v(i)*sin(delta(i)-theta(i))-(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)
    *v(i)*cos(delta(i)-theta(i)))/A*v(i)*sin(delta(i)-theta(i))+(R(i)*Epq(i)-Xpd(i)*Epd(i)
    -R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(i)-theta(i)))/A*v(i)*cos(delta
    (i)-theta(i));
gxy((i-1)*2+pos(12))=(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)
    ))/A*v(i)*cos(delta(i)-theta(i))+(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(de
    lta(i)-theta(i)))/A*v(i)*sin(delta(i)-theta(i))-(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(
    i)*v(i)*sin(delta(i)-theta(i)))/A*v(i)*sin(delta(i)-theta(i))+(R(i)*Epd(i)+Xpq(i)*Epq(
    i)-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)))/A*v(i)*cos(del
    ta(i)-theta(i))-(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(i)-theta(i)))
    /A*v(i)*sin(delta(i)-theta(i))+(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos(delta
    (i)-theta(i)))/A*v(i)*cos(delta(i)-theta(i))-(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)
    *v(i)*cos(delta(i)-theta(i)))/A*v(i)*cos(delta(i)-theta(i))-(R(i)*Epq(i)-Xpd(i)*Epd(i)
    -R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(i)-theta(i)))/A*v(i)*sin(delta
    (i)-theta(i));
```

```
    temp=blockdiag(temp,gxy);
end

%for machine M
kk=[    1    2    1    2    1    2    1    2];
mm=[    1    1    2    2    1    1    2    2];
nn=[    1    1    1    1    2    2    2    2];
pos=kk+(mm-1)*(2*N)+ (NS-1)*(2*N)*(nn-1);

for i=M:M
    gxy=sparse(2*N*(NS-1),2);
    A=R(i)^2+Xpd(i)*Xpq(i);

    gxy((i-1)*2+pos(1))=Xpq(i)/A*sin(delta(i)-theta(i))+R(i)/A*cos(delta(i)-theta(i));
    gxy((i-1)*2+pos(2))=Xpq(i)/A*cos(delta(i)-theta(i))-R(i)/A*sin(delta(i)-theta(i));
    gxy((i-1)*2+pos(3))=R(i)/A*sin(delta(i)-theta(i))-Xpd(i)/A*cos(delta(i)-theta(i));
    gxy((i-1)*2+pos(4))=R(i)/A*cos(delta(i)-theta(i))+Xpd(i)/A*sin(delta(i)-theta(i));
    gxy((i-1)*2+pos(5))=-Xpq(i)/A*v(i)*cos(delta(i)-theta(i))+R(i)/A*v(i)*sin(delta(i)-theta(
        i));
    gxy((i-1)*2+pos(6))=Xpq(i)/A*v(i)*sin(delta(i)-theta(i))+R(i)/A*v(i)*cos(delta(i)-theta(i
        ));
    gxy((i-1)*2+pos(7))=-R(i)/A*v(i)*cos(delta(i)-theta(i))-Xpd(i)/A*v(i)*sin(delta(i)-theta(
        i));
    gxy((i-1)*2+pos(8))=R(i)/A*v(i)*sin(delta(i)-theta(i))-Xpd(i)/A*v(i)*cos(delta(i)-theta(i
        ));

    temp=blockdiag(temp,gxy);
end

Gxy=sparse(zeros( (2*N)*(NS*M-1),2*N));
Gxy( 1:(2*N)*(NS*M-1),1:2*M)=temp;

temp=sparse(zeros(2*N,2*N));
for i=1:N
    temp((i-1)*2+2*2*N*(i-1)+1)=-Pl0(i)*(1/v0(i)*Kip(i)+2*v(i)/v0(i)^2*Kzp(i))*Kw(i)*(1+alpha
        *Kpl(i));
    temp((i-1)*2+2*2*N*(i-1)+2)=-Ql0(i)*(1/v0(i)*Kiq(i)+2*v(i)/v0(i)^2*Kzq(i))*Kw(i)*(1+alpha
        *Kql(i));
end

Gxy(2*N*(NS*M-2)+[1:2*N],1:2*N)=temp;
```

A.2.10 Second derivatives - Gxa

```
function Gxa=getGxa(x,Sys,Loads)

M=Sys(1);
N=Sys(2);
NS=Sys(3);
v=x(NS*M+2*[0:N-1]);
```

```
%%%%%%%%%%%%loads%%%%%%%%%%%%%%%%%%
v0=Loads(1,:);
Pl0=Loads(2,:);
Kpp=Loads(3,:);
Kip=Loads(4,:);
Kzp=Loads(5,:);
Kpl=Loads(6,:);

Ql0=Loads(7,:);
Kpq=Loads(8,:);
Kiq=Loads(9,:);
Kzq=Loads(10,:);
Kql=Loads(11,:);

Kw=Loads(12,:);

gxa=zeros(2*N,1);
gxa(1+2*[0:N-1])=-Pl0.*(Kpp+v./v0.*Kip+v.^2./v0.^2.*Kzp).*Kw.*Kpl;
gxa(2*[1:N])    =-Ql0.*(Kpq+v./v0.*Kiq+v.^2./v0.^2.*Kzq).*Kw.*Kql;

Gxa=sparse([zeros(2*N*(NS*M-2),1);gxa]);
```

A.2.11 Second derivatives - Gyx

```
function Gyx=getGyx(x,Sys,Machines,Governors,Excitors,Loads,alpha)

M=Sys(1); N=Sys(2); NS=Sys(3);
% note: all the vectors are row vectors.
Epq=x(1:NS:NS*(M-1)+1); Epd=x(2:NS:NS*(M-1)+ 2); Efd=x(3:NS:NS*(M-1)+3);
Vr=x(4:NS:NS*(M-1)+ 4); Rf=x(5:NS:NS*(M-1)+ 5); Pm=x(6:NS:NS*(M-1)+6);
Miu=x(7:NS:NS*(M-1)+7); omega=x(NS-1:NS:NS*(M-1)+ NS-1);
delta=[x(NS:NS:NS*(M-1)) 0];

v=x(NS*M+2*[0:N-1]); theta=x(NS*M+1+2*[0:N-1]);
%%%%%%%%%%machine%%%%%%%%%%%%%%%%%%
R=Machines(1,:); Xd=Machines(2,:); Xq=Machines(3,:);
Xpd=Machines(4,:); Xpq=Machines(5,:);
Tpd0=Machines(6,:); Tpq0=Machines(7,:); Mg=Machines(8,:);
D=Machines(9,:); Pg0=Machines(10,:); Km=Machines(11,:);
%%%%%%%%%%governor%%%%%%%%%%%%%%%%%%
K1=Governors(1,:); Tg=Governors(2,:); Tch=Governors(3,:);
%%%%%%%%%%%%excitor%%%%%%%%%%%%%%%%%%
Ka=Excitors(1,:); Ta=Excitors(2,:); Ke=Excitors(3,:);
Te=Excitors(4,:); Se=Excitors(5,:); Kf=Excitors(6,:);
Tf=Excitors(7,:); Vref=Excitors(8,:);
%%%%%%%%%%%%load%%%%%%%%%%%%%%%%%%%%
v0=Loads(1,:); Pl0=Loads(2,:); Kpp=Loads(3,:);
Kip=Loads(4,:); Kzp=Loads(5,:); Kpl=Loads(6,:);

Ql0=Loads(7,:); Kpq=Loads(8,:); Kiq=Loads(9,:);
```

```
Kzq=Loads(10,:); Kql=Loads(11,:);

Kw=Loads(12,:);

% for machine 1 to M-1
kk=[    1    2    1    2    1    2    1    2    1    2    1    2];
mm=[    1    1    2    2    1    1    2    2    1    1    2    2];
nn=[    1    1    1    1    2    2    2    2    9    9    9    9];
pos=kk+(mm-1)*(2*N)+ 2*(2*N)*(nn-1);

temp=[];
for i=1:M-1
    gyx=sparse(2*N*2,NS);
    A=R(i)^2+Xpd(i)*Xpq(i);

    gyx((i-1)*2+pos(1))=Xpq(i)/A*sin(delta(i)-theta(i))+R(i)/A*cos(delta(i)-theta(i));
    gyx((i-1)*2+pos(2))=Xpq(i)/A*cos(delta(i)-theta(i))-R(i)/A*sin(delta(i)-theta(i));
    gyx((i-1)*2+pos(3))=-Xpq(i)/A*v(i)*cos(delta(i)-theta(i))+R(i)/A*v(i)*sin(delta(i)-theta(
        i));
    gyx((i-1)*2+pos(4))=Xpq(i)/A*v(i)*sin(delta(i)-theta(i))+R(i)/A*v(i)*cos(delta(i)-theta(i
        ));
    gyx((i-1)*2+pos(5))=R(i)/A*sin(delta(i)-theta(i))-Xpd(i)/A*cos(delta(i)-theta(i));
    gyx((i-1)*2+pos(6))=R(i)/A*cos(delta(i)-theta(i))+Xpd(i)/A*sin(delta(i)-theta(i));
    gyx((i-1)*2+pos(7))=-R(i)/A*v(i)*cos(delta(i)-theta(i))-Xpd(i)/A*v(i)*sin(delta(i)-theta(
        i));
    gyx((i-1)*2+pos(8))=R(i)/A*v(i)*sin(delta(i)-theta(i))-Xpd(i)/A*v(i)*cos(delta(i)-theta(i
        ));
    gyx((i-1)*2+pos(9))=(-R(i)*cos(delta(i)-theta(i))+Xpq(i)*sin(delta(i)-theta(i)))/A*v(i)*s
        in(delta(i)-theta(i))+(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(delta(i)-thet
        a(i)))/A*sin(delta(i)-theta(i))+(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-thet
        a(i)))/A*v(i)*cos(delta(i)-theta(i))+(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)
        -theta(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)))/A*cos(delta(i)-theta(i))+(R(i)*sin(delt
        a(i)-theta(i))+Xpd(i)*cos(delta(i)-theta(i)))/A*v(i)*cos(delta(i)-theta(i))+(R(i)*v(i)
        *sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos(delta(i)-theta(i)))/A*cos(delta(i)-theta(i))-(
        -R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-theta(i)))/A*v(i)*sin(delta(i)-theta(
        i))-(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(
        i)-theta(i)))/A*sin(delta(i)-theta(i));
    gyx((i-1)*2+pos(10))=(-R(i)*cos(delta(i)-theta(i))+Xpq(i)*sin(delta(i)-theta(i)))/A*v(i)*
        cos(delta(i)-theta(i))+(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(delta(i)-the
        ta(i)))/A*cos(delta(i)-theta(i))-(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-the
        ta(i)))/A*v(i)*sin(delta(i)-theta(i))-(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i
        )-theta(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)))/A*sin(delta(i)-theta(i))-(R(i)*sin(del
        ta(i)-theta(i))+Xpd(i)*cos(delta(i)-theta(i)))/A*v(i)*sin(delta(i)-theta(i))-(R(i)*v(i
        )*sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos(delta(i)-theta(i)))/A*sin(delta(i)-theta(i))-
        (-R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-theta(i)))/A*v(i)*cos(delta(i)-theta
        (i))-(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta
        (i)-theta(i)))/A*cos(delta(i)-theta(i));
    gyx((i-1)*2+pos(11))=(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)
        ))/A*v(i)*sin(delta(i)-theta(i))-(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(de
        lta(i)-theta(i)))/A*v(i)*cos(delta(i)-theta(i))+(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(
        i)*v(i)*sin(delta(i)-theta(i)))/A*v(i)*cos(delta(i)-theta(i))+(R(i)*Epd(i)+Xpq(i)*Epq(
        i)-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)))/A*v(i)*sin(del
```

```
        ta(i)-theta(i))+(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(i)-theta(i)))
        /A*v(i)*cos(delta(i)-theta(i))+(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos(delta
        (i)-theta(i)))/A*v(i)*sin(delta(i)-theta(i))-(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)
        *v(i)*cos(delta(i)-theta(i)))/A*v(i)*sin(delta(i)-theta(i))+(R(i)*Epq(i)-Xpd(i)*Epd(i)
        -R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(i)-theta(i)))/A*v(i)*cos(delta
        (i)-theta(i));
    gyx((i-1)*2+pos(12))=(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)
        ))/A*v(i)*cos(delta(i)-theta(i))+(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpq(i)*v(i)*sin(de
        lta(i)-theta(i)))/A*v(i)*sin(delta(i)-theta(i))-(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(
        i)*v(i)*sin(delta(i)-theta(i)))/A*v(i)*sin(delta(i)-theta(i))+(R(i)*Epd(i)+Xpq(i)*Epq(
        i)-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos(delta(i)-theta(i)))/A*v(i)*cos(del
        ta(i)-theta(i))-(-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(i)-theta(i)))
        /A*v(i)*sin(delta(i)-theta(i))+(R(i)*v(i)*sin(delta(i)-theta(i))+Xpd(i)*v(i)*cos(delta
        (i)-theta(i)))/A*v(i)*cos(delta(i)-theta(i))-(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)
        *v(i)*cos(delta(i)-theta(i)))/A*v(i)*cos(delta(i)-theta(i))-(R(i)*Epq(i)-Xpd(i)*Epd(i)
        -R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin(delta(i)-theta(i)))/A*v(i)*sin(delta
        (i)-theta(i));

    temp=blockdiag(temp,gyx);
end

% for machine M
kk=[    1    2    1    2    1    2    1    2];
mm=[    1    1    2    2    1    1    2    2];
nn=[    1    1    1    1    2    2    2    2];
pos=kk+(mm-1)*(2*N)+ 2*(2*N)*(nn-1);

for i=M:M
    gyx=sparse(2*N*2,(NS-1));
    A=R(i)^2+Xpd(i)*Xpq(i);

    gyx((i-1)*2+pos(1))=-Xpq(i)/A*sin(theta(i))+R(i)/A*cos(theta(i));
    gyx((i-1)*2+pos(2))=Xpq(i)/A*cos(theta(i))+R(i)/A*sin(theta(i));
    gyx((i-1)*2+pos(3))=-Xpq(i)/A*v(i)*cos(theta(i))-R(i)/A*v(i)*sin(theta(i));
    gyx((i-1)*2+pos(4))=-Xpq(i)/A*v(i)*sin(theta(i))+R(i)/A*v(i)*cos(theta(i));
    gyx((i-1)*2+pos(5))=-R(i)/A*sin(theta(i))-Xpd(i)/A*cos(theta(i));
    gyx((i-1)*2+pos(6))=R(i)/A*cos(theta(i))-Xpd(i)/A*sin(theta(i));
    gyx((i-1)*2+pos(7))=-R(i)/A*v(i)*cos(theta(i))+Xpd(i)/A*v(i)*sin(theta(i));
    gyx((i-1)*2+pos(8))=-R(i)/A*v(i)*sin(theta(i))-Xpd(i)/A*v(i)*cos(theta(i));

    temp=blockdiag(temp,gyx);
end

Gyx=sparse(zeros( (2*N)^2,NS*M-1));
Gyx( 1:(2*N)*2*M,1:(NS*M-1))=temp;

%% for X(end)
temp=sparse(zeros( (2*N)^2,1));
for i=1:N
    temp((i-1)*2+2*2*N*(i-1)+1)=-Pl0(i)*(1/v0(i)*Kip(i)+2*v(i)/v0(i)^2*Kzp(i))*Kw(i)*(1+alpha
        *Kpl(i));
    temp((i-1)*2+2*2*N*(i-1)+2)=-Ql0(i)*(1/v0(i)*Kiq(i)+2*v(i)/v0(i)^2*Kzq(i))*Kw(i)*(1+alpha
```

```
        *Kql(i));
end

Gyx( 1:end,end)=temp;
```

A.2.12 Second derivatives - Gyy

```
function Gyy=getGyy(x,Sys,Machines,Governors,Excitors,Loads,Ybus_abs,Ybus_angle,alpha)

M=Sys(1); N=Sys(2); NS=Sys(3);
% note: all the vectors are row vectors.
Epq=x(1:NS:NS*(M-1)+1); Epd=x(2:NS:NS*(M-1)+ 2); Efd=x(3:NS:NS*(M-1)+3);
Vr=x(4:NS:NS*(M-1)+ 4); Rf=x(5:NS:NS*(M-1)+ 5); Pm=x(6:NS:NS*(M-1)+6);
Miu=x(7:NS:NS*(M-1)+7); omega=x(NS-1:NS:NS*(M-1)+ NS-1);
delta=[x(NS:NS:NS*(M-1)) 0];

v=x(NS*M+2*[0:N-1]); theta=x(NS*M+1+2*[0:N-1]);
%%%%%%%%%%machine%%%%%%%%%%%%%%%%%%
R=Machines(1,:); Xd=Machines(2,:); Xq=Machines(3,:); Xpd=Machines(4,:);
Xpq=Machines(5,:); Tpd0=Machines(6,:); Tpq0=Machines(7,:);
Mg=Machines(8,:); D=Machines(9,:); Pg0=Machines(10,:); Km=Machines(11,:);
%%%%%%%%%%governor%%%%%%%%%%%%%%%%%%
K1=Governors(1,:); Tg=Governors(2,:); Tch=Governors(3,:);
%%%%%%%%%%%%excitor%%%%%%%%%%%%%%%%%%
Ka=Excitors(1,:); Ta=Excitors(2,:); Ke=Excitors(3,:); Te=Excitors(4,:);
Se=Excitors(5,:); Kf=Excitors(6,:); Tf=Excitors(7,:); Vref=Excitors(8,:);
%%%%%%%%%%%%loads%%%%%%%%%%%%%%%%%%%%%%
v0=Loads(1,:); Pl0=Loads(2,:); Kpp=Loads(3,:); Kip=Loads(4,:);
Kzp=Loads(5,:); Kpl=Loads(6,:);

Ql0=Loads(7,:); Kpq=Loads(8,:); Kiq=Loads(9,:); Kzq=Loads(10,:);
Kql=Loads(11,:);

Kw=Loads(12,:);

Gyy=sparse(zeros( (2*N)^2,2*N));
for i=1:N
    for k=i:N
        gyy1=sparse(zeros(2*N,1));
        gyy2=sparse(zeros(2*N,1));
        gyy3=sparse(zeros(2*N,1));
        gyy4=sparse(zeros(2*N,1));

        if k==i
            if i<=M
                A=R(i)^2+Xpd(i)*Xpq(i);

                gyy1( (i-1)*2+1)
                    =-2*Ybus_abs(i,k)*cos(Ybus_angle(i,k))-2*Pl0(i)/v0(i)^2*Kzp(i)*(1+Kw(i)*(
                    omega(M)-1))*(1+alpha*Kpl(i))+2*(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(
                    delta(i)-theta(i)))/A*sin(delta(i)-theta(i))+2*(-R(i)*cos(delta(i)-theta(
```

```
    i))+Xpd(i)*sin(delta(i)-theta(i)))/A*cos(delta(i)-theta(i));
gyy1( (i-1)*2+2) =
    2*(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-theta(i)))/A*cos(delt
    a(i)-theta(i))-2*(-R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-theta(
    i)))/A*sin(delta(i)-theta(i))-2*Ql0(i)/v0(i)^2*Kzq(i)*(1+Kw(i)*(omega(M)-
    1))*(1+alpha*Kql(i))+2*Ybus_abs(i,k)*sin(Ybus_angle(i,k));

gyy2(1+2*[0:N-1])=
    v.*Ybus_abs(:,i).'.*sin(-theta+Ybus_angle(:,i).'+theta(i));
gyy2(2*[1:N])    =
    v.*Ybus_abs(:,i).'.*cos(-theta+Ybus_angle(:,i).'+theta(i));
gyy2( (i-1)*2+1)=
    sum(v.*Ybus_abs(i,:).*sin(theta(i)-Ybus_angle(i,:)-theta),2)-v(i)*Ybus_ab
    s(i,i).*sin(theta(i)-Ybus_angle(i,i)-theta(i)) ...
-(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos
    (delta(i)-theta(i)))/A*cos(delta(i)-theta(i)) ...
+(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin
    (delta(i)-theta(i)))/A*sin(delta(i)-theta(i)) ...
-(-R(i)*sin(delta(i)-theta(i))-Xpq(i)*cos(delta(i)-theta(i)))/A*v(i)*cos(del
    ta(i)-theta(i)) ...
+(-R(i)*sin(delta(i)-theta(i))-Xpd(i)*cos(delta(i)-theta(i)))/A*v(i)*cos(del
    ta(i)-theta(i)) ...
+(-R(i)*cos(delta(i)-theta(i))+Xpd(i)*sin(delta(i)-theta(i)))/A*v(i)*sin(del
    ta(i)-theta(i)) ...
+(R(i)*cos(delta(i)-theta(i))-Xpq(i)*sin(delta(i)-theta(i)))/A*v(i)*sin(delt
    a(i)-theta(i)) ...
+(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos(delta(i)-theta(i)))/A*co
    s(delta(i)-theta(i)) ...
+(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin(delta(i)-theta(i)))/A*sin
    (delta(i)-theta(i)) ;

gyy2( (i-1)*2+2)=-sum(v.*Ybus_abs(i,:).*cos(theta(i)-Ybus_angle(i,:)-theta),
    2)+v(i)*Ybus_abs(i,i).*cos(theta(i)-Ybus_angle(i,i)-theta(i)) ...
+(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos
    (delta(i)-theta(i)))/A*sin(delta(i)-theta(i)) ...
+(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin
    (delta(i)-theta(i)))/A*cos(delta(i)-theta(i)) ...
+
    v(i)*(R(i)+3*sin(delta(i)-theta(i))*(Xpd(i)-Xpq(i))*cos(delta(i)-theta(i)
    ))/A ;

gyy4(1+2*[0:N-1])=
    v*v(i).*Ybus_abs(:,i).'.*cos(-theta+Ybus_angle(:,i).'+theta(i));
gyy4(2*[1:N])
    =-v*v(i).*Ybus_abs(:,i).'.*sin(-theta+Ybus_angle(:,i).'+theta(i));

gyy4( (i-1)*2+1) =
    v(i)*sum(v.*Ybus_abs(i,:).*cos(theta(i)-Ybus_angle(i,:)-theta),2)-v(i)*v(
    i)*Ybus_abs(i,i).*cos(theta(i)-Ybus_angle(i,i)-theta(i)) ...
+(R(i)*v(i)*sin(delta(i)-theta(i))+Xpq(i)*v(i)*cos(delta(i)-theta(i)))/A*v(i
    )*sin(delta(i)-theta(i)) ...
-2*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin(delta(i)-theta(i)))/A*v
```

```
                    i)*Ybus_abs(i,i).*cos(theta(i)-Ybus_angle(i,i)-theta(i)) ;
                gyy4( (i-1)*2+2) =
                    v(i)*sum(v.*Ybus_abs(i,:).*sin(theta(i)-Ybus_angle(i,:)-theta),2)-v(i)*v(
                    i)*Ybus_abs(i,i).*sin(theta(i)-Ybus_angle(i,i)-theta(i)) ;
            end
        else
            gyy1( (i-1)*2+1) =-Ybus_abs(i,k)*cos(-theta(i)+Ybus_angle(i,k)+theta(k));
            gyy1( (i-1)*2+2) = Ybus_abs(i,k)*sin(-theta(i)+Ybus_angle(i,k)+theta(k));
            gyy1( (k-1)*2+1) =-Ybus_abs(k,i)*cos(-theta(k)+Ybus_angle(k,i)+theta(i));
            gyy1( (k-1)*2+2) = Ybus_abs(k,i)*sin(-theta(k)+Ybus_angle(k,i)+theta(i));

            gyy2( (i-1)*2+1) = v(k)*Ybus_abs(i,k)*sin(-theta(i)+Ybus_angle(i,k)+theta(k));
            gyy2( (i-1)*2+2) = v(k)*Ybus_abs(i,k)*cos(-theta(i)+Ybus_angle(i,k)+theta(k));
            gyy2( (k-1)*2+1) =-v(k)*Ybus_abs(k,i)*sin(-theta(k)+Ybus_angle(k,i)+theta(i));
            gyy2( (k-1)*2+2) =-v(k)*Ybus_abs(k,i)*cos(-theta(k)+Ybus_angle(k,i)+theta(i));

            gyy3( (i-1)*2+1) =-v(i)*Ybus_abs(i,k)*sin(-theta(i)+Ybus_angle(i,k)+theta(k));
            gyy3( (i-1)*2+2) =-v(i)*Ybus_abs(i,k)*cos(-theta(i)+Ybus_angle(i,k)+theta(k));
            gyy3( (k-1)*2+1) = v(i)*Ybus_abs(k,i)*sin(-theta(k)+Ybus_angle(k,i)+theta(i));
            gyy3( (k-1)*2+2) = v(i)*Ybus_abs(k,i)*cos(-theta(k)+Ybus_angle(k,i)+theta(i));

            gyy4( (i-1)*2+1)
                =-v(i)*v(k)*Ybus_`abs(i,k)*cos(-theta(i)+Ybus_angle(i,k)+theta(k));
            gyy4( (i-1)*2+2) =
                v(i)*v(k)*Ybus_abs(i,k)*sin(-theta(i)+Ybus_angle(i,k)+theta(k));
            gyy4( (k-1)*2+1)
                =-v(k)*v(i)*Ybus_abs(k,i)*cos(-theta(k)+Ybus_angle(k,i)+theta(i));
            gyy4( (k-1)*2+2) =
                v(k)*v(i)*Ybus_abs(k,i)*sin(-theta(k)+Ybus_angle(k,i)+theta(i));

            Gyy( 2*(k-1)*2*N+[1:2*N],2*i)=gyy3;
        end

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        Gyy( 2*(k-1)*2*N+[1:2*N],2*(i-1)+1)=gyy1;
        Gyy((2*k-1)*2*N+[1:2*N],2*(i-1)+1)=gyy2;
        Gyy((2*k-1)*2*N+[1:2*N],2*i)=gyy4;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%
% till now, we have only done the lower triangle part of Gyy
for k=1:2*N
    Gyy(1:2*N*(k-1),k)=reshape(Gyy(2*N*(k-1)+[1:2*N],1:k-1),2*N*(k-1),1);
end
```

A.2.13 Second derivatives - Gya

```
function Gya=getGya(x,Sys,Loads)

M=Sys(1); N=Sys(2); NS=Sys(3);
```

```
omega=x(NS-1:NS:NS*(M-1)+ NS-1);
v=x(NS*M+2*[0:N-1]);

%%%%%%%%%%%loads%%%%%%%%%%%%%%%%%%%%%
v0=Loads(1,:); Pl0=Loads(2,:); Kpp=Loads(3,:); Kip=Loads(4,:);
Kzp=Loads(5,:); Kpl=Loads(6,:);

Ql0=Loads(7,:); Kpq=Loads(8,:); Kiq=Loads(9,:); Kzq=Loads(10,:);
Kql=Loads(11,:);

Kw=Loads(12,:);

gya=sparse(zeros((2*N)^2,1));
for i=1:N
    gya( 2*2*N*(i-1) + 2*(i-1) +
       1 )=-Pl0(i)*(1/v0(i)*Kip(i)+2*v(i)/v0(i)^2*Kzp(i))*(1+Kw(i)*(omega(M)-1))*Kpl(i);
    gya( 2*2*N*(i-1) + 2*(i-1) +
       2 )=-Ql0(i)*(1/v0(i)*Kiq(i)+2*v(i)/v0(i)^2*Kzq(i))*(1+Kw(i)*(omega(M)-1))*Kql(i);
end
Gya=gya;
```

BIBLIOGRAPHY

[1] Vijay Vittal, "Project Proposal for the 2001 Solicitation from PS_{ERC}", 2001

[2] Prabha Kundur, Power System Stability and Control, McGraw-Hill, Companies, Inc, 1994. ISBN 0-07-030958-X

[3] Xiaoming Wang, Wei Shao, Vijay Vittal, "Adaptive Corrective Control Strategies For Preventing Power System Blackouts", to appear in 15th Power Systems Computation Conference, Liège, Belgium, 2005

[4] M. M. Adibi, Power System Restoration: Methodologies and Implementation Strategies, Wiley-IEEE Press, 2000, ISBN 0780353978

[5] Final Report on the August 14th Blackout in the United States and Canada, https://reports.energy.gov/BlackoutFinal-Web.pdf, March 16, 2005

[6] H. You, V. Vittal, X. Wang, "Slow Coherency Based Islanding", IEEE Transactions on Power Systems, Vol. 19, No. 1, pp. 483 – 491, Feb. 2004

[7] J. H. Chow, R. Date, "A nodal aggregation algorithm for linearized two-time-scale power networks," in IEEE International Symposium on Circuits and Systems, vol. 1, pp. 669–672, 1988

[8] R. A. Date, J. H. Chow, "Aggregation properties of linearized two-time-scale power networks," IEEE Transactions on Circuits and Systems, vol. 38, issue 7, July 1991, pp. 720–730

[9] J. H. Chow, R. Galarza, P. Accari, W. W. Price, "Inertial and slow coherency aggregation algorithms for power system dynamic model reduction," IEEE Transactions on Power Systems, vol. 10, issue 2, May 1995, pp. 680–685

[10] J. H. Chow, R. A. Date, H. Othma, W. W. Price, "Slow coherency aggregation of large power system," in Eigenanalysis and Frequency Domain Methods for System Dynamic Performance, IEEE Publications 90TH0292-3-PWR, 1990, pp. 50-60

[11] S. B. Yusof, G. J. Rogers, R. T. H. Alden, "Slow coherency based network partitioning including load buses," IEEE Transactions on Power Systems, vol. 8, issue 3, August 1993, pp. 1375–1382

[12] S. Tiptipakorn, "A spectral bisection partitioning method for electric power network applications," PSERC 2001 Publications

[13] J. S. Thorp, H. Wang, Computer Simulation of Cascading Disturbances in Electric Power Systems, Impact of Protection Systems on Transmission System Reliability Final Report, PSERC Publication 01-01. May 2001

[14] M. Montagna, G. P. Granelli, "Detecting of Jacobian singularity and network islanding in power flow computions," IEE Proc.-Gener. Transm. Distrib., vol. 142, no. 6, November 1995, pp. 589–594

[15] J. E. Kim, J. S. Hwang, "Islanding detection method of distributed generation units

connected to power distribution system," in International Conference on Power System Technology, 2000. Proceedings of PowerCon 2000, vol. 2, pp. 643–647, 2000

[16] Kai Sun, Da-Zhong Zheng, Qiang Lu, "Splitting Strategies for Islanding Operation of Large-Scale Power Systems Using OBDD-Based Methods", IEEE Transactions Power Systems, Vol. 18, No. 2, pp. 912-923, May 2003

[17] http://sourceforge.net/projects/buddy, February 2, 2005

[18] Antoine Rauzy, "Mathematical Foundations of Minimal Cutsets", IEEE Transact- ions On Reliability, Vol. 50, No. 4, pp. 389-396, Dec. 2001

[19] Heejong Suh, Carl K. Chang, "Algorithms for the Minimal Cutsets Enumeration of Networks by Graph Search and Branch Addition", Local Computer Networks, 2000. LCN 2000. Proceedings. 25th Annual IEEE Conference on , 8-10 Nov. 2000, pp:100 - 107

[20] Huang-Yau Lin, Sy-Yen Kuo, Fu-Min Yeh, "Minimal Cutset Enumeration and Network Reliability Evaluation by Recursive Merge and BDD", Computers and Communication, 2003. (ISCC 2003). Proceedings. Eighth IEEE International Symposium on , pp:1341-1346, 2003

[21] J. George Shanthikumar, "Bounding Netowrk-Reliability Using Consecutive Mi- nimal Cutsets", IEEE Transactions On Reliability, Vol. 37, No. 1, pp:45-49, Apr. 1988

[22] Joe H. Chow, "New algorithms for slow coherency aggregation of large power systems", *System and Control Theory for Power Systems*, Springer-Verlag, 1995

[23] W. Price et al., "Improved dynamic equivalencing software," GE Power Systems Engineering, EPRI TR-105 919 Project 2447-02, Dec. 1995

[24] Joe. H. Chow, "Time-Scale Modeling of Dynamic Networks with Applications to Power Systems," Lectures notes in Control and Information Sciences. 46. Springer-Verlag Berlin. Heidelberg. New York. 1982

[25] Vinod Chachra, Prabhakar M. Ghare, James M. Moore, *Applications of Graph Theory Algorithms*, New York: Elsevier North Holland, 1979. ISBN 0-444-00268-5

[26] Edward M. Reingold, Jurg Nievergelt, Narsingh Deo, *Combinatorial Algorithms: Theory and Practice,* New Jersey: Prentice-Hall, 1977. ISBN 0-13-152447-X

[27] Xiaoming Wang, Vijay Vittal, "Slow Coherency Grouping Based Islanding Using Minimal Cutsets", Session 1P2, 'Coherency and Aggregation', 35th North American Power Symposium, University of Missouri-Rolla, 2003

[28] Les Pereira, "New Thermal Turbine Governor Modeling for the WECC", WECC Modeling and Validation Work Group, October 11, 2002 (Revised)

[29] Governor Modeling Task Force, "Guidelines For Thermal Governor Model Data Selection, Validation, and Submittal to WECC", WECC Modeling & Validation Work Group, October 9, 2002, Revised November 4, 2002

[30] NERC Glossary of terms, ftp://www.nerc.com/pub/sys/all_updl/docs/pubs/glossv 10.pdf, Febrary 2, 2005

[31] Zhong Yang, "A new automatic under-frequency load shedding scheme", Master dissertation, Iowa State University, 2001

[32] Ontario Hydro, "Dynamic Reduction," Version 1.1, vol. 2: User's Manual (Revision 1) EPRI TR-102234-V2R1, May 1994.

[33] Ontario Hydro, "Interactive Power Flow (IPFLOW)" Version 2.1, vol. 2: User's Manual, EPRI TR-103643-V2, May 1994.

[34] V. Ajjarapu, class notes of EE 653 Advanced Power System, 2003, Fall

[35] Y. Zhou, X. Wen, V. Ajjarapu, "Identification and Tracing of Oscillatory Stability Margin Boundaries", Power Engineering Society General Meeting, 2003, IEEE , Volume: 4 , July 13-17, 2003, Pages:2104 - 2110

[36] X. Wen, V. Ajjarapu, "Critical Eigenvalue Trajectory Tracing for Power System Oscillatory Stability Assessment", Proceedings of IEEE Power Engineering Society meeting in Denver , June 2004

[37] Ontario Hydro, "Small Signal Stability Analysis Program (SSSP)," v. 2, EPRI TR-101850-V2R1, May 1994, Version 3.1, User's Manual (Rev. 1).

[38] Z. Feng, V. Ajjarapu, D.J. Maratukulam, "A comprehensive approach for preventive and corrective control to mitigate voltage collapse", Power Systems, IEEE Transactions on, vol. 15, pp. 791-797, May 2000

[39] Stephen L. Richter, Raymond A. Decarlo, "Continuation Methods: Theory and Application", IEEE Transactions On Circuits and Systems, Vol. CAS-30, No. 6, pp. 347-352, Jun. 1983 & IEEE Transactions on Automatic Control, vol. 28 , pp: 660-665, Jun. 1983

[40] V. Ajjarapu, C. Christy, "The Continuation Power Flow: A tool for Steady State Voltage Stability Analysis", Power Industry Computer Application Conference, 1991. Conference Proceedings, 7-10 May 1991

[41] W.H. Press, S.A. Teukolsky, W.T. Vetterling, Brian P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing, 2nd Edition*, Cambridge, England: Cambridge University Press, 2003.

[42] M. I. Friswell, "Model Reduction Using Dynamic and Iterated IRS techniques", Journal of sound and vibration, 186 (2), pp. 311-323, 1995

```
        (i)*cos(delta(i)-theta(i)) ...
    -(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos
        (delta(i)-theta(i)))/A*v(i)*sin(delta(i)-theta(i)) ...
    +(R(i)*v(i)*cos(delta(i)-theta(i))-Xpd(i)*v(i)*sin(delta(i)-theta(i)))/A*v(i
        )*cos(delta(i)-theta(i)) ...
    +2*(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos(delta(i)-theta(i)))/A*
        v(i)*sin(delta(i)-theta(i)) ...
    -(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin
        (delta(i)-theta(i)))/A*v(i)*cos(delta(i)-theta(i)) ;

    gyy4( (i-1)*2+2) =
        v(i)*sum(v.*Ybus_abs(i,:).*sin(theta(i)-Ybus_angle(i,:)-theta),2)-v(i)*v(
        i)*Ybus_abs(i,i).*sin(theta(i)-Ybus_angle(i,i)-theta(i)) ...
    +(R(i)*v(i)*sin(delta(i)-theta(i))+Xpq(i)*v(i)*cos(delta(i)-theta(i)))/A*v(i
        )*cos(delta(i)-theta(i)) ...
    +2*(R(i)*v(i)*cos(delta(i)-theta(i))-Xpq(i)*v(i)*sin(delta(i)-theta(i)))/A*v
        (i)*sin(delta(i)-theta(i)) ...
    -(R(i)*Epd(i)+Xpq(i)*Epq(i)-R(i)*v(i)*sin(delta(i)-theta(i))-Xpq(i)*v(i)*cos
        (delta(i)-theta(i)))/A*v(i)*cos(delta(i)-theta(i)) ...
    -(R(i)*v(i)*cos(delta(i)-theta(i))-Xpd(i)*v(i)*sin(delta(i)-theta(i)))/A*v(i
        )*sin(delta(i)-theta(i)) ...
    +2*(-R(i)*v(i)*sin(delta(i)-theta(i))-Xpd(i)*v(i)*cos(delta(i)-theta(i)))/A*
        v(i)*cos(delta(i)-theta(i)) ...
    +(R(i)*Epq(i)-Xpd(i)*Epd(i)-R(i)*v(i)*cos(delta(i)-theta(i))+Xpd(i)*v(i)*sin
        (delta(i)-theta(i)))/A*v(i)*sin(delta(i)-theta(i)) ;
else
    gyy1( (i-1)*2+1) =
        -2*Ybus_abs(i,k)*cos(Ybus_angle(i,k))-2*Pl0(i)/v0(i)^2*Kzp(i)*(1+Kw(i)*(o
        mega(M)-1))*(1+alpha*Kpl(i));
    gyy1( (i-1)*2+2) =
        2*Ybus_abs(i,k)*sin(Ybus_angle(i,k))-2*Ql0(i)/v0(i)^2*Kzq(i)*(1+Kw(i)*(om
        ega(M)-1))*(1+alpha*Kql(i));

    gyy2(1+2*[0:N-1])=
        v.*Ybus_abs(:,i).'.*sin(-theta+Ybus_angle(:,i).'+theta(i));
    gyy2(2*[1:N])    =
        v.*Ybus_abs(:,i).'.*cos(-theta+Ybus_angle(:,i).'+theta(i));
    gyy2( (i-1)*2+1) =
        sum(v.*Ybus_abs(i,:).*sin(theta(i)-Ybus_angle(i,:)-theta),2)-v(i)*Ybus_ab
        s(i,i).*sin(theta(i)-Ybus_angle(i,i)-theta(i)) ;
    gyy2( (i-1)*2+2)
        =-sum(v.*Ybus_abs(i,:).*cos(theta(i)-Ybus_angle(i,:)-theta),2)+v(i)*Ybus_
        abs(i,i).*cos(theta(i)-Ybus_angle(i,i)-theta(i)) ;

    gyy3=gyy2;
    gyy4(1+2*[0:N-1])=
        v*v(i).*Ybus_abs(:,i).'.*cos(-theta+Ybus_angle(:,i).'+theta(i));
    gyy4(2*[1:N])
        =-v*v(i).*Ybus_abs(:,i).'.*sin(-theta+Ybus_angle(:,i).'+theta(i));

    gyy4( (i-1)*2+1) =
        v(i)*sum(v.*Ybus_abs(i,:).*cos(theta(i)-Ybus_angle(i,:)-theta),2)-v(i)*v(
```