

# **EFFICIENT ALGORITHMS TO EXPEDITE TRANSIENT STABILITY ANALYSIS OF POWER SYSTEMS**

by

Sajjad Zadehkhost

B.Sc. The Khaje Nasir University of Technology, 2007

M.Sc. The Sharif University of Technology, 2009

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate and Postdoctoral Studies

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

April 2015

© Sajjad Zadehkhost, 2015

# Abstract

With rapid increase in complexity of modern power systems, there is a strong need for better computational tools to ensure the reliable operation of electrical grids. These tools need to be accurate, computationally efficient, and capable of using advanced measurement devices. In this context, transient stability assessment (TSA) is an important study that determines system's dynamic security margins following a major disturbance. The TSA consists of a set of differential-algebraic equations (DAEs), which are typically solved using time-domain simulation (TDS) approach. While being very accurate, the TDS requires significant computational resources when applied to practical power systems. This problem becomes more significant in transient stability monitoring (TSM), wherein the computational performance of the TDS is typically the bottleneck. This research is to investigate available challenges in the TSM applications and develop new algorithms to help realizing a practical monitoring tool for transient stability studies. The thesis focuses on three research thrusts: i) dynamic reduction of power system to reduce problem size; ii) advanced computation approaches to expedite the TDS method; iii) integration of PMU measurements into the TSM. Initially, a new adaptive aggregation algorithm for dynamic reduction is proposed, wherein parameters of generators and structure of transmission network are considered to aggregate coherent generators and create a reduced-order system. Also, a new criterion is defined to monitor validity of the constructed reduced system. It is shown that the proposed technique is more accurate than traditional aggregation methods. To expedite the TDS approach, this thesis presents two new integration techniques, which are called Multi-Decomposition Approach (MDA) and

Successive Linearization and Integration Technique (SLIT). In these methods, the nonlinear DAEs are decomposed into a series of linear subsystems, which participate in approximating actual solution. It is demonstrated that sequential and parallel versions of the MDA and SLIT are faster than state-of-the-art integration techniques. Finally, a dynamic state estimator based on Extended Kalman Filter is developed to convert the PMU measurements into a set of state variables suitable for transient stability studies. Computer studies show that the proposed framework provides accurate results in highly disturbed power systems with fairly low PMU sampling rates.

# Preface

Based on the research presented in this thesis, several papers have been published in conference proceedings, published and/or submitted as journal articles. In all publications, I have developed mathematical formulations, implemented models, conducted simulations, analyzed results, and prepared the majority of manuscripts. My supervisors, Dr. Jatskevich and Dr. Vaahedi, have provided me with instructive comments and corrections throughout the process of conducting research studies and preparing manuscripts. The following describes published and submitted papers as well as contributions of other co-authors.

Section 2.1 was presented at a conference. S. Zadkhast, A. Alimardani, J. Jatskevich, and E. Vaahedi, "Identifying coherent areas in transmission system for transient stability studies in future smart grids," Power and Energy Society (PES) General Meeting, 21-25 July 2013. As a colleague graduate student in our group, A. Alimardani was working with me on this manuscript and provided comments and suggestions on the studies and content of this paper.

A part of Section 2.2 has been published as a conference paper. S. Zadkhast, J. Jatskevich, E. Vaahedi, A. Alimardani, "An Adaptive Aggregation Algorithm for Power System Dynamic Equivalencing in Transient Stability Studies for Future Energy Grids," In proceedings of International Conference on Power Systems Transients (IPST 2013), Vancouver, BC, Canada, 18-20 July 2013. This paper was later improved and published as a journal article. S. Zadkhast, J. Jatskevich, E. Vaahedi, A. Alimardani, "A new adaptive dynamic reduction method for power system transient stability problems," Electric Power Systems Research,

vol. 115, pp. 102-110, 2014. A. Alimardani has provided comments and useful suggestions for preparing manuscripts.

A part of Chapter 3 has been published in a journal article. S. Zadkhast, J. Jatskevich, E. Vaahedi, "A Multi-Decomposition Approach for Accelerated Time-Domain Simulation of Transient Stability Problems," IEEE Transaction on Power Systems, 2014. DOI: 10.1109/TPWRS.2014.2361529.

A part of Chapter 4 has been submitted for peer review. S. Zadkhast, J. Jatskevich, E. Vaahedi, "A Fast and Accurate Time Domain Transient Stability Method Using Successive Linearization Technique".

A part of Chapter 5 is under preparation for a journal article. S. Zadkhast, J. Jatskevich, E. Vaahedi, "Using Extended Kalman Filter for Dynamic State Estimation in Transient Stability Studies", to be submitted.

# Table of Contents

<b>Abstract</b> .....	ii
<b>Preface</b> .....	iv
<b>Table of Contents</b> .....	vi
<b>List of Tables</b> .....	x
<b>List of Figures</b> .....	xii
<b>List of Abbreviations</b> .....	xv
<b>Acknowledgements</b> .....	xvi
<b>Dedication</b> .....	xvii
<b>Chapter 1: INTRODUCTION</b> .....	1
1.1 Motivation .....	1
1.2 Background.....	5
1.2.1 Solving the DAEs in transient stability problem.....	6
1.2.1.1 Dynamic equivalencing.....	8
1.2.1.2 Parallel computing .....	11
1.2.2 Using phasor measurement units in the TSM.....	12
1.3 State-of-the-Art Research.....	14
1.3.1 Dynamic equivalencing .....	14
1.3.2 Parallel computing .....	15
1.3.3 Using phasor measurement units in the TSA.....	17
1.4 Research Objectives and Anticipated Impacts .....	19
<b>Chapter 2: AGGREGATION OF GENERATORS AND BUSES FOR DYNAMIC EQUIVALENCING</b> .....	24
2.1 Aggregating Generators .....	24

2.1.1 Formulation of the transient stability problem .....	24
2.1.2 Inertial aggregation algorithm .....	25
2.1.3 Network-dependent aggregation (NDA) technique for finding generators weights.....	29
2.1.4 Aggregating coherent generators.....	32
2.1.5 Errors in coherency identification .....	36
2.1.6 Simulation results.....	42
2.2 Aggregation in Transmission System .....	51
2.2.1 Identifying coherent areas in transmission system .....	51
2.2.2 Reducing redundancy .....	54
2.2.3 Simulation results.....	57
<b>Chapter 3: MULTI-DECOMPOSITION APPROACH .....</b>	<b>63</b>
3.1 Conventional Time-Domain Simulation Approach.....	63
3.2 Proposed Multi-Decomposition Approach .....	65
3.2.1 Taylor series of multivariable functions.....	66
3.2.2 Multi-decomposition approach.....	68
3.3 Applying MDA to Transient Stability Problem.....	71
3.3.1 Application of MDA to single machine system .....	71
3.3.2 Adaptive updating scheme for the MDA .....	74
3.4 Parallel Multi-Decomposition Approach.....	77
3.5 Benchmark System and Simulation Results.....	80
3.5.1 Case study.....	81
3.5.2 Calculation of critical clearing time .....	82
3.5.3 Evaluate performance of the MDA during the fault .....	84
3.5.4 Computational performance .....	86

3.5.5 Large-scale system .....	89
<b>Chapter 4: SUCCESSIVE LINEARIZATION AND INTEGRATION TECHNIQUE .....</b>	<b>91</b>
4.1 Mathematical Formulation .....	91
4.2 Practical Considerations and Implementation.....	95
4.3 Using Multi-Core Processor Architecture with SLIT .....	98
4.4 Handling Saturation Function with Hard Limits .....	100
4.5 Benchmark System and Simulation Results.....	103
4.5.1 Case study.....	104
4.5.2 Verification and accuracy evaluation.....	105
4.5.3 Computational performance using large-scale system.....	107
<b>Chapter 5: USING PHASOR MEASUREMENT UNITS IN REAL-TIME TSA .....</b>	<b>112</b>
5.1 Problem Statement .....	112
5.2 Applying Extended Kalman Filter to DSE Problem .....	116
5.2.1 Review of extended kalman filter.....	116
5.2.2 Prediction stage in the TSA.....	119
5.2.3 Update stage in the TSA - simultaneous approach .....	122
5.2.4 Update stage in the TSA – sequential approach.....	125
5.3 Implementation.....	126
5.3.1 Avoid inverting matrix of sensitivity factors .....	126
5.3.2 Summary .....	127
5.4 Simulation Studies.....	129
5.4.1 Comparing accuracy of local DSE against global DSE.....	129
5.4.2 Effect of discretization method .....	130
5.4.3 Simultaneous vs. sequential processing of measurements .....	133

5.4.4 Effect of updating sensitivity factors.....	134
<b>Chapter 6: SUMMARY OF CONTRIBUTIONS AND FUTURE WORKS .....</b>	<b>137</b>
6.1 Conclusions and Contributions .....	137
6.2 Potential Impacts of Contributions.....	141
6.3 Future Work .....	142
<b>References.....</b>	<b>145</b>
<b>APPENDIX A: VALIDATION OF EQUATION (2-31).....</b>	<b>157</b>

# List of Tables

Table 2-1 The Groups of Tightly Coherent Generators .....	44
Table 2-2 The Groups of Coherent Generators in the Second Study.....	45
Table 2-3 The Groups of Coherent Generators with One Non-Coherent Group.....	48
Table 2-4 Plant Generation Limit for Generator at bus #104.....	50
Table 2-5 Load Models.....	60
Table 2-6 Coherent Buses Identified by TDS and Proposed Method.....	61
Table 2-7 Reduction Achieved After Aggregating Coherent Buses .....	61
Table 3-1 Comparison of CCT Calculated by DSATools and MDA.....	84
Table 3-2 Execution Time for Different Scenarios.....	88
Table 3-3 Total Execution Time for Fourteen Scenarios.....	88
Table 3-4 Execution Time for Different Scenarios in Large-Scale System.....	90
Table 4-1 CPU Execution Time for Different Scenarios.....	109
Table 4-2 Comparing Number of Jacobian Calculations and LU Factorization Performed in Different Methods.....	110
Table 4-3 CPU Execution Time Of Parallel MDA and Parallel SLIT.....	111
Table 5-1 Nomenclature of Machine Parameters.....	113
Table 5-2 Evaluation of Error and CPU Timing for Euler and Runge-Kutta Methods.....	133

Table 5-3 Comparing Performance of Simultaneous and Sequential Approach for Processing Measurements.....	134
Table 5-4 Evaluating Accuracy and Performance of DSE When the Update of Sensitivity Factors is Skipped.....	136

# List of Figures

Figure 1-1 Structure of one computational cycle in transient stability monitoring.....	3
Figure 1-2 Detailed structure of a transient stability monitoring package.....	20
Figure 2-1 The procedure of creating an equivalent generator in the IAA. ....	28
Figure 2-2 Creating an equivalent generator from two coherent generators.....	33
Figure 2-3 Block diagram of the aggregated group of coherent generators when local and inter-area oscillations are neglected within the coherent group.....	36
Figure 2-4 Block diagram of the proposed error identification algorithm. ....	40
Figure 2-5 Rotor angle of generator #60 in the first study. ....	44
Figure 2-6 Plot of rotor angle of generator #60 in the second study. ....	46
Figure 2-7 Rotor angle of generators #104 and #111 in the full-order system.....	48
Figure 2-8 Rotor angle of generator #60 in the presence of an error in coherency identification.....	49
Figure 2-9 Voltage angle of second group consisting of buses #{3,4,5,6,9,10,12,33,34,35,36,37,38,39,40,47,48,49,50,72}.....	62
Figure 2-10 Voltage magnitudes of second group consisting of buses #{3,4,5,6,9,10,12,33,34,35,36,37,38,39,40,47,48,49,50,72}.....	62
Figure 3-1 Block diagram of the VDHN method for solving system of nonlinear equations at each integration step.....	65

Figure 3-2 Decomposing nonlinear system into linear subsystems using MDA.....	66
Figure 3-3 Computing system response $\mathbf{x}(t)$ using interconnected linear subsystems according to the proposed MDA.....	71
Figure 3-4 Transient in the rotor angle as predicted by various orders of MDA for the single-machine example system (3-15).....	73
Figure 3-5 Approximation error in the rotor angle resulting from using MDA.....	73
Figure 3-6 Block diagram of the MDA assuming trapezoidal integration rule.....	75
Figure 3-7 Block diagram of sharing computational tasks among three processors in the inner-loop of MDA.....	79
Figure 3-8 Implementation of the inner loop in parallel MDA.....	80
Figure 3-9 Bus voltage and rotor angle of the generator at bus #90.....	82
Figure 3-10 Voltages of five buses when fault is applied at bus #73.....	86
Figure 3-11 Rotor angle of generator at bus #101 simulated by three methods when fault is applied at bus #73.....	86
Figure 3-12 Duplicating base test system to create a large-scale system.....	90
Figure 4-1 Block diagram representation of implementation of the SLIT assuming trapezoidal integration rule.....	97
Figure 4-2 Sharing computational tasks among $N$ processors in the parallel SLIT.....	100
Figure 4-3 Saturation function with hard limits.....	102
Figure 4-4 Derivative of saturation function with hard limits.....	102

Figure 4-5 Approximating derivative of saturation function using function $D(x)$ . .....	103
Figure 4-6 Rotor angle and field voltage of generator at bus #111. ....	105
Figure 4-7 Percentage of integration error in VDHN and SLIT methods. ....	106
Figure 4-8 Duplicating base test system to create a large-scale system. ....	108
Figure 5-1. Comparing global versus local dynamic state estimation.....	130
Figure 5-2 Dynamic state estimation provided by Euler and RK4 methods for PMU sampling rate at 60 samples/sec. ....	132
Figure 5-3 Dynamic state estimation provided by Euler and RK4 methods for PMU sampling rate at 30 samples/sec. ....	132
Figure 5-4 Dynamic state estimation results derived by simultaneous and sequential processing of measured data.....	134
Figure 5-5 Effect of updating sensitivity factors on accuracy of DSE results. ....	136
Figure 6-1 Integrating the methods developed in this thesis into a transient stability monitoring tool.....	138

# List of Abbreviations

<b>Abbreviation</b>	<b>Meaning</b>
AVR	Automatic Voltage Regulator
CCT	Critical Clearing Time
DAE	Differential-Algebraic Equations
DSE	Dynamic State Estimation
EKF	Extended Kalman Filter
EPF	Extended Particle Filter
IAA	Inertial Aggregation Algorithm
MDA	Multi-Decomposition Approach
MF	Magnification Factor
MPT	Maximum Prediction Time
NDA	Network-Dependent Aggregation
PMU	Phasor Measurement Unit
SCADA	Supervisory Control and Data Acquisition
SLIT	Successive Linearization and Integration Technique
TDS	Time-Domain Simulation
TSA	Transient Stability Assessment
TSM	Transient Stability Monitoring
UKF	Unscented Kalman Filter
VBR	Voltage Behind Reactance
VDHN	Very DisHonest Newton

# Acknowledgements

Foremost, I would like to express my sincere gratitude to my supervisors Prof. Juri Jatskevich and Dr. Ebrahim Vaahedi for their continuous support, patience, great inspiration, immense knowledge and insights, and enthusiasm that they provided me throughout my studies. I appreciate all their contribution of time and ideas to make my doctoral experience productive and stimulating. The financial support for this research was made possible through the UBC's Four-Year-Fellowship program and by the Natural Science and Engineering Research Council (NSERC) of Canada under the Strategic Project Grant "Enabling Solutions for the Future Canadian Smart Grid".

My sincere thanks also go to Prof. Uri Ascher whose lectures I attended. Also, I would like to thank Dr. Luis Linares, Dr. Ebrahim Vaahedi, and Mr. Nathan Ozog for the wonderful teaching experience I gained while assisting them.

I would like to also thank my fellow colleagues graduate students in Electric Power and Energy Systems research group at the University of British Columbia: Arash Taavighi, Francis Therrien, Milad Fekri, Arash Alimardani, Soroush Amini, Mehrdad Chapariha, Yingwei Huang, Matin Rahmatian, and Hamed Ahmadi, for many stimulating discussions and all the fun we have had in the past four years at UBC.

The last but not the least, I would like to thank my amazing and loving wife Faranak Nobakhtian who has been very encouraging and supportive throughout this journey.

# Dedication

*To My Wonderful Wife Faranak*

# CHAPTER 1: INTRODUCTION

## 1.1 Motivation

Due to many technological advancements in energy sector, modern power systems have been pushed to their physical operating limits more than ever before. To increase the utilization of available infrastructures and resources, stability margins have decreased considerably, making the systems potentially more prone to instability following a major disturbance. To continue operating power systems in a stable and reliable manner, electrical utilities and engineers are using sophisticated computer simulations and tools in order to monitor, analyze, and control the system during unforeseen events. Currently, state estimation, voltage stability analysis, and transient stability analysis (TSA) are a part of dynamic security assessment (DSA) tool, which are carried out either offline or online using model of the system and measurements from supervisory control and data acquisition (SCADA) systems.

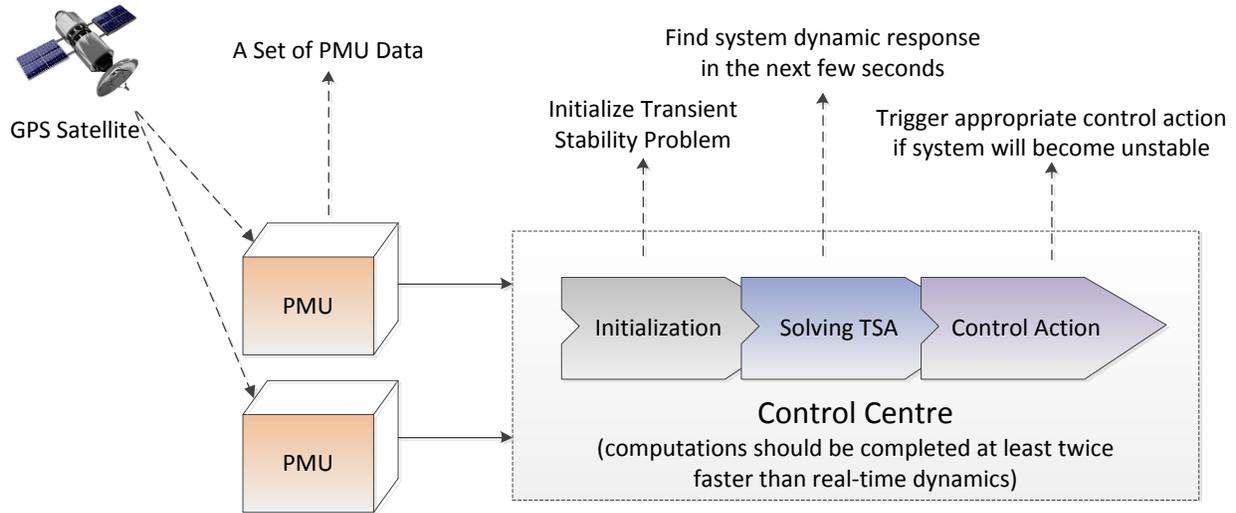
Among different studies performed in the DSA, the TSA is one of the most time-demanding dynamic studies. Time-domain simulation (TDS) is the most reliable approach for solving the TSA problems, wherein numerical integration is employed. The TDS is able to handle a wide variety of dynamic models and provide very accurate results. Several commercial packages, such as DSATools from Powertech Labs Inc. [1], PSS/E from Siemens [2],

PowerWorld Simulator from Power World Corporation [3], and PSLF from General Electric Energy [4] have been developed, wherein the TDS is the primary method for solving transient stability problem. While modern simulation tools are constantly improving, simulation speed and time-consuming nature of the TDS remain to be limiting factors.

Recently, phasor measurement units (PMUs) with modern communication facilities have also become widely available. The additional accurate, high-rate, and synchronized measurements provided by the PMUs are able to capture transients of power systems and therefore, they open the opportunity to perform the TSA following occurrence of fault or disturbance. This scheme is known as transient stability monitoring (TSM) and it aims at using PMU data to monitor and control system's dynamic behavior in real-time. The overall structure of the TSM is shown in Figure 1-1. Initially, PMU measurements are collected across the network and transferred to control centre, wherein the measured quantities are translated into system's state variables to initialize the transient stability problem. Next, the TDS approach is employed to solve the differential-algebraic equations (DAEs) appearing in the transient stability problem for the next few seconds and provide a short-term prediction of system dynamic status. If an instability is detected in this step, appropriate corrective actions are taken to stabilize the system.

The computational cycle depicted in Figure 1-1 encompasses three steps: initializing transient stability problem; solving the DAEs; taking appropriate corrective action. In a practical TSM, this cycle should be completed at least two times faster than real-time dynamics to allow some time for taking appropriate corrective action. However, the state-of-the-art transient stability solvers are not fast enough to achieve this objective. In fact,

with current solution speed of the TDS method, each TSM cycle is almost 4 times slower than real-time system dynamics when applied to very detailed and large power grids with over 15,000 buses, such as Western Electricity Coordinating Council (WECC) system [5]. Therefore, in order to make TSM computationally feasible, there is a strong need for expediting transient stability solution.



**Figure 1-1 Structure of one computational cycle in transient stability monitoring.**

In this regard, a significant effort has been made to expedite the transient stability assessment using theory of nonlinear systems [6]-[10] and the concept of instability in single-machine infinite bus (SMIB) system [11]-[13]. These methods are able to solve the problem without numerical integration, which results in a significant time advantage over the TDS. However, with the rapid development of new devices with sophisticated dynamic models, their inability to handle different models became apparent. Additionally, these methods occasionally may give inaccurate results, which has limited their applications to contingency selection.

System dynamic reduction technique is another approach for reducing size of the system and improving performance of the TDS. In the state-of-the-art solutions, the coherent generators may be aggregated using a weighted averaging approach, wherein the weight of each generator is proportional to its inertia. However, this approach may result in a sub-optimal reduced-order system if a large generator is not able to deliver enough power in the post-disturbance system. More importantly, available dynamic equivalencing techniques fail to provide accurate results when there errors exist in coherency identification step.

Using multi-core processors has been another active area of research for improving computational performance of TDS. The traditional approaches use system partitioning techniques to share computational tasks among several processors. However, this approach, if applied in a straightforward way, suffers from significant overhead communication cost and leads to a reduction in convergence rate of the nonlinear solver used inside.

The TSM helps increasing utilization of available resources while maintaining dynamic security of the system, which ultimately translates into significant economic benefits. In order to realize a TSM tool, there is a considerable need to develop new methods and algorithms that expedite solution speed of the TDS approach. The application of new tools, however, will not limited to the TSM and they can also be used in offline and online transient stability studies to model larger power systems with more contingencies and/or perform the TSA more often. In both cases, having more accurate and fast transient

simulation tools enhances the awareness of system operators and improves reliability of the system.

## **1.2 Background**

In transient stability studies, system dynamics are divided into short-term (fast) and long-term (slow) transients. For the purpose of transient stability studies, transmission network, including transmission lines, transformers, and loads, are considered to belong to the category of fast transients, for which differential equations are substituted by algebraic equations that are obtained assuming equilibrium conditions. Also, the generators' windings, excitation systems, and turbine-governor control system define long-term transients of the system and they are represented by differential equations. Overall, the transient stability problem constitutes a set of differential-algebraic equations (DAEs).

In the online TSA [14], these DAEs are solved using numerical integration for a number of predefined contingencies to ensure that the system is reliable against foreseeable incidents. In this application, a snapshot of the system operating condition is taken using SCADA measurements, and it is used to initialize the DAEs. Due to poor measurement resolution, the SCADA system is not able to capture transient dynamics of system, and therefore, its application has been traditionally limited to steady state conditions. With the development of PMUs providing synchronized measurements at high-rate, it is now possible to monitor transients of the system and perform the TSA using real-time measurements, which is known as the transient stability monitoring (TSM) [15]. As discussed in section 1.1, the solution speed of the transient stability solver is the main bottleneck for the TSM, which is

almost 4 times slower than real-time system dynamics when applied to large-scale systems. In this situation, the TSM cannot identify and secure unstable scenarios in a timely and reliable manner. Another challenge in the TSM is that while the internal states of synchronous machines (and other dynamic devices) are required to initiate the transient stability problem, the PMUs can only measure electrical quantities such as voltage and current. As a result, an intermediate step is required to transform the PMUs' measured quantities into states of dynamic devices. Therefore, in order to achieve an industrial-grade TSM, the above problems should be addressed first. Numerous solutions have been proposed in the literature to tackle these challenges, which are reviewed in the next two sections.

### **1.2.1 Solving the DAEs in transient stability problem**

The TDS [16]-[18] has been proven to be the most accurate and reliable approach for solving DAEs appearing in transient stability problems. In this approach, the DAEs are solved numerically and the system is simulated typically for 3 to 5 seconds for studying the local modes, and for 10 to 20 seconds for studying inter-area modes [19]. This approach can handle various dynamic models of system components.

There are two basic methodologies for solving transient stability DAEs [17] using TDS. In the first approach (also called power-balance form), the equations are solved simultaneously. In this approach, an implicit integration technique is used to discretize the differential equations, which are then solved together with the algebraic equations. In the second approach (also called current-balance form), the DAEs are partitioned, and the discretized differential equations along with algebraic equations are solved separately,

alternating each time step. According to [20], the alternating approach tends to be faster as it allows to use explicit integration rules such as Runge-Kutta [21]. However, the transient stability problem can be stiff with highly nonlinear elements, especially when control systems have large gains, small time constants, saturation limiters, etc. [22]. In this situation, the simultaneous method is preferred as it uses implicit integration and ensures numerical stability.

Since the DAEs are generally nonlinear, the solution of such problems typically requires iterations using either Full Newton method (with complete update of Jacobian matrix at every iteration) [23] or a very dishonest Newton (VDHN) method (with skipping of updating the Jacobian matrix) [24]. Despite recent advancements in computer technology, the TDS requires significant computational resources when applied to practical industrial problems. As a result, considerable effort has been devoted in the literature to find a replacement for the TDS. Energy function method [6]-[10] is one of the alternative approaches where the system stability is assessed by comparing system's energy level against the energy level at certain unstable equilibrium point. Its inability in handling different dynamic models is the main problem associated with the energy function method. Extended equal area criterion (EEAC) [11]-[13] is a hybrid approach, wherein the TDS along with the concept of energy function method are used to detect instability. While being more computationally efficient, the results of EEAC may not always be sufficiently accurate or even conservative [25]. Recently, several heuristic methods based on artificial neural networks [26], support vector machines [27], and decision tree [28] have been proposed as fast tools for the transient stability assessment.

Despite of many innovations, significant economic implications faced by utilities prevent operators from using approximate methods, and many companies still prefer to use the TDS as the most trustable method. For this reason, many research efforts have been focused on expediting the TDS using dynamic equivalencing and parallel computing.

#### **1.2.1.1 Dynamic equivalencing**

Dynamic equivalencing aims at reducing size of the system by aggregating coherent generators. The steps involved in dynamic equivalencing include: (i) identifying groups of coherent generators [29]-[34]; and (ii) aggregating a coherent group to create an equivalent generator [35]-[42]. The first step involves identification of generators with similar angular swing curves. The accuracy of reduced-order system highly depends on the results of stage (i), which is supposed to identify only tightly coherent generators. Linear time simulation (LTS) [29] is one of the earliest methods, which simulates the linearized system using trapezoidal integration with large time step (e.g. 100ms), and the groups of coherent generators are identified using simulation results. Weak couplings approach was proposed in [30], wherein a group of coherent generators is recognized when the coupling between this group and the rest of system is weak. Modal analysis and eigenvalue decomposition approaches are powerful tools proposed in [31], [32], wherein participation factors [33] were used to measure the effect of each machine on system modes. The main challenge associated with the modal-based methods is that when several generators (which are not necessarily coherent) participate in the same mode, these methods cannot separate such generators and may give inaccurate results. Moreover, the modal-based methods typically need the number of coherent groups of generators as the input, which is

not easy to determine before solving the transient stability problem. The groups of generators can also be found using relation factors [34], which are calculated directly from state matrix. In this approach, the coherent groups are determined using relative degree of coupling.

Once the groups of coherent generator are identified, each group is aggregated to create an equivalent machine. The generator terminal bus aggregation algorithm [35] is one of the earliest methods in this area. This method aggregates the generators using an inertial weighting average. The problem associated with this approach is that the infinite admittances are introduced, which can stiffen the problem. The inertial and slow coherency aggregation method [36] is one of the well-established algorithms in this area. In this approach, the aggregation is performed at the generator's internal nodes using weighted average of generators. The weight of each generator is determined by its inertia and accordingly, the generators with large inertia contribute more to the dynamic characteristics of the corresponding equivalent generator. Moreover, the linearized swing equation is used to modify the impedance connecting the terminals of coherent generators. This modification eliminates stiffening problem encountered in [35] and it might slightly improve the accuracy of the reduced-order system. Using terminal bus method for aggregating generators was investigated in [37], wherein the terminals of the coherent generators are connected to an equivalent bus via an ideal transformer with complex ratio. Since the method is based on the power preservation at the terminal bus of the equivalent generator, it preserves the steady state characteristics of the system. However, this method is less accurate than the inertial and slow coherency method [36] since the terminals of the

coherent generators are less coherent compared to the internal nodes. Using participation factors for aggregating generators was investigated in [38], wherein a reference machine is selected for each group of coherent generators and the weight of each generator is determined based on the participation factor between the given generator and the reference machine. The synchrony aggregation approach was presented in [39], where the reference generator is represented in detailed model and the rest of the generators in the group are represented as a current source.

The size of the system can be further reduced by extending the dynamic reduction to transmission level. In transient stability studies, more than half of the equations are attributed to the transmission system. By locating and aggregating coherent load buses, it is possible to reduce the number of corresponding equations and expedite the transient stability programs. Although significant literature is devoted to coherency identification at generation level, much less has been published regarding identifying coherent areas in the transmission systems. One of the earliest methods in this area was proposed in [32], where a small machine is added to each load bus to convert it to a generation bus. Coherent areas are then found using eigenvalue decomposition approach. This method considerably increases the problem size and creates a large number of modes which are too close to each other and therefore difficult to separate. Assigning load buses to generators was presented in [43], where a connection between a load bus and a generator is established via modal analysis. This method tries to find load buses which are coherent with generators. Using network reduction and keeping only high voltage part of the system (230kv and higher)

was investigated in [44]. Therein, the network reduction is applicable as long as all loads are modeled as constant impedance.

#### **1.2.1.2 Parallel computing**

With the recent development of high-performance computing, significant effort in the literature has been devoted to using multi-core processors to expedite the TDS. In the most straightforward approach,  $m$  contingencies are shared among  $m$  processors, which has a potential of reducing the computing time by almost  $m$  factors. While this approach is very efficient in analyzing large number of independent scenarios, similar performance improvement is hardly achieved in a single-contingency TDS problem - which is sequential in nature. As a result, several parallel algorithms have been developed over the past two decades to expedite the TDS in a single-contingency problem. These algorithms can be broadly divided into two categories: parallel-in-time and parallel-in-space methods. In parallel-in-time method [47]-[49], several time-steps are solved concurrently by enlarging the size of the problem solved simultaneously. However, due to sequential dependency between integration time-steps, convergence rate of such relaxed approaches significantly degrades especially when large numbers of processors are used. In parallelism-in-space [50]-[59], several processors are employed to solve a single time-step. In the Multi-Area Thevenin Equivalent (MATE) method presented in [50]-[54], system is decomposed into a number of sub-networks, which are connected by the links. In this approach, the solution is computed in two stages. First, sub-networks are solved independently, assuming that the links are open. In this stage, the computational tasks are shared among several processors to improve performance of the algorithm. In the second stage, the sub-networks are

represented by their Thevenin equivalent to find the current flowing through the links. This process is continued iteratively until the solution converges. In another approach presented in [55]-[59], network equations are decomposed into several groups considering structure of problem as well as network topology. The groups of variables are subsequently shared among processors based on their size and required computational cost. Each processor solves the assigned sub-problem by relaxing variables from other sub-problems and at the end of each iteration, the processors exchange the interfacing variables to update the solution. While parallelism-in-space can efficiently reduce computational time, there are two challenges associated with this approach. Firstly, the Full Newton and VDN methods used for solving system of nonlinear equations are replaced with a parallelizable algorithm such as Gauss-Seidel [49], Successive-Over Relaxation (SOR) [56], or Gauss-Jacobi [58]-[59], which can significantly reduce convergence rate and increase number of iterations. Additionally, in algorithms based on parallel-in-space approach, each processor needs to access memory of processors corresponding to adjacent areas, which can impose significant communication overhead cost. Due to these challenges, applications of parallel computing for a single-contingency TDS problem have been limited to large power systems with few partitions.

### **1.2.2 Using phasor measurement units in the TSM**

In order to use PMUs in the TSM, their measurements need to be translated into a set of states of dynamics components such as machines. In this regard, [60] used extended Kalman filter (EKF) method to estimate internal states of a generator using the PMUs installed at the high voltage buses. In this approach, the field voltage was assumed to be

unknown, and it was estimated along with other states of generator by augmenting state vector. While it was demonstrated that the EKF is able to accurately estimate machine's states, the system simulation time-step and PMU's sampling rate were the same in the studies presented in [60], which is not a valid assumption considering the PMUs' measurement resolution (30~60 samples/sec). Applying the EKF to a wide-area measurement system (WAMS) was proposed in [61], wherein the data collected from all PMUs were used to estimate states of all generators. In this approach, the state estimation was carried out in two stages. In the first stage, a static state estimation based on the EKF was used to estimate voltage of terminal buses of generators using PMU measurements. In the second stage, estimated terminal voltages were used to calculate the internal states of the generators using the EKF method. While it was demonstrated that the EKF accurately estimates the states of machines, [61] oversimplified the problem by considering a classic model for generators and a linear model for the system. In a typical transient stability problem, wherein the nonlinear terms are highly excited and the excitation system has significant effect on the stability limit, this level of simplification might not be acceptable. In order to avoid the limitations imposed by the linearization step in the EKF, [62]-[64] proposed using unscented Kalman filter (UKF) in dynamic state estimation problem. This method was able to eliminate the Jacobian matrix computation step; however, it requires a large number of derivate function evaluations which ultimately makes it less efficient when applied to large-scale power systems. Using extended particle filter (EPF) was investigated in [65], wherein the mean and covariance of states were propagated via Monte Carlo simulation. Similar to the UKF, EKF does not require computation of Jacobian matrix, and moreover, it is able to handle non-Gaussian noise. However, large number of function

evaluations in this method is a significant challenge when applied to large-scale systems. Ensemble kalman filter (EnKF) is presented in [66] wherein, the noise vector is augmented to model inaccuracies in parameters. This vector is then used to estimate the system parameters as well as dynamic states.

## **1.3 State-of-the-Art Research**

### **1.3.1 Dynamic equivalencing**

The dynamic equivalencing approach has been developed to account for the computational challenges associated with the rapid expansion of power systems. This method has been an active area of research for more than 3 decades and various research groups in the world have focused on creating accurate and reliable dynamic equivalents of original system.

Prof. J. H. Chow and his research group at Rensselaer Polytechnic Institute have made considerable contributions to developing coherency identification and aggregation methods. The slow coherency approach was originally proposed for identifying coherency in the system [32] and inertial weighted average [36] for aggregating coherent machines. Both of these methods have received significant acceptance in research community.

DYNRED is a software program that was originally developed by EPRI in 1990 to create dynamically reduced systems. In 2010, Powertech Labs. Inc. completed and commercialized the DYNRED package [67]. This project was sponsored by EPRI and it aimed at adding performance evaluation indices to DYNRED, and making it compatible with present data formats used by industry. The DYNRED is currently a graphical tool that

reduces size of the system for both static and dynamic studies. Moreover, it provides performance benchmarking tools to evaluate accuracy of the reduced system.

At Arizona State University, Prof. V. Vittal and his research group have focused on determining appropriate size of reduced-order system using online SCADA measurements [68], [69]. They have also proposed a hybrid dynamic equivalencing method using artificial neural network approach to improve the accuracy of dynamically reduced system [70].

North Carolina State University has also been active in developing dynamically equivalent models. Dr. A. Chakraborty and his research group have focused on using graph theory [71] and PMUs [72], [73] to aggregate coherent generators and create an equivalent reduced-order model.

The research group at National Technology University of Athens has been researching in area of aggregating control systems of AVRs and turbine-governors. In particular, they have presented a method that is suitable for aggregating prime movers and excitation systems in quasi-steady-state studies [40]-[42].

### **1.3.2 Parallel computing**

Parallel computing has been an active area of research for more than two decades. In the recent years, with a significant increase in the availability of multi-core processing units, parallel computing is now playing an increasingly important role in power systems computational and modeling tools, and more software developers are becoming interested in using parallel computation in their commercial packages.

Universita' degli Studi di Napoli in Italy has been a research centre for developing parallel algorithms for transient stability studies. They have proposed several parallel algorithms including Gauss-Jacobi [48] and Gauss-Seidel [49] to expedite TDS method.

Prof. V. Dinavahi and his research group at University of Alberta have made considerable contributions with developing real-time transient stability simulation tools. Instantaneous-relaxation-based method [58], [59] is a parallel Gauss-Jacobi approach developed by Prof. V. Dinavahi and his research team for solving transient stability problems. This method has been implemented on CPUs as well as graphic processing units (GPUs).

The research group in University of Liege in Belgium has made significant participation in accelerating transient stability simulations on large-scale systems using multi-thread computer systems. Recently, Prof. T. V. Cutsem and his research group have focused on network partitioning techniques to use multi-core processors for expediting time-domain simulation in transient stability studies.

The Electric Power and Energy Systems research group at the University of British Columbia has been active in the area of transient stability since 1972 [18]. In the more recent publications, Prof. J. R. Marti and his research group have presented a Multi-Area Thevenin Equivalent approach [50]-[54] to partition the network equations and solve them in parallel.

For the past two years, Pacific Northwest National Laboratory has collaborated with Powertech Labs. Inc. to use network partitioning algorithms for parallelizing network

solution in DSATools. Using a 64-core processor, they have been able to improve the solution speed of differential equations by 45 times, and network equations by 4 times. It is claimed that the overall simulation speed has been improved by 25 times.

A joint research group between Lawrence Livermore National Laboratory [74] and General Electric Energy has recently investigated the possibility of using parallel computing in GE Concordia PSLF software [4]. The group has used matrix re-ordering techniques to improve computational performance of single-contingency problems. It has been shown that speed improvement becomes saturated when more than 5 processors are used. Also, the maximum speed improvement is slightly more than 3 times, which is achieved using 12 processors [75].

### **1.3.3 Using phasor measurement units in the TSA**

PMUs are one of the most advanced measurement devices available in power systems. They are becoming an increasingly important part of power systems protection, monitoring, and control. There has been a considerable amount of research from various groups in the world focused on the using PMUs in different types of studies such as model validation and calibration, system protection, and dynamic simulation.

Prof. A. Abur and his research group at the Texas A&M University have made significant contributions to developing accurate and robust state estimation algorithms. In the more recent publications, they have used PMUs in static state estimation studies [76]-[80]. In particular, the PMU measurements are employed to identify status of unobservable areas [78] and eliminate bad measurements [79]-[80].

Prof. J. H. Chow and his research group at Rensselaer Polytechnic Institute have been researching in the area of transient stability studies and dynamic equivalencing for more than two decades. In their recent publications [73], [81], they have used PMUs to construct a dynamically equivalent model of major areas in power system. The results are then used to monitor the major power transfer paths of electrical grid.

In the recent years, Georgia Institute of Technology, Georgia, USA, has also been active in the area of using PMUs led by Prof. A. P. S. Meliopoulos. In the recent publications, they have utilized PMU measurements to estimate machine parameters and create a dynamically equivalent system [82], [83]. Also, Prof. A. P. S. Meliopoulos and his research group have been focusing on re-constructing a real-time model of the system using PMU measurements [84].

Prof. I. Kamwa and his research group at Laval University, Quebec, Canada, have used PMUs in dynamic state estimation studies. In the recent publication [60], they have proposed the EKF to estimate unknown inputs of the system. A variation of the UKF has also been presented in [62] to reduce computational cost of dynamic state estimator.

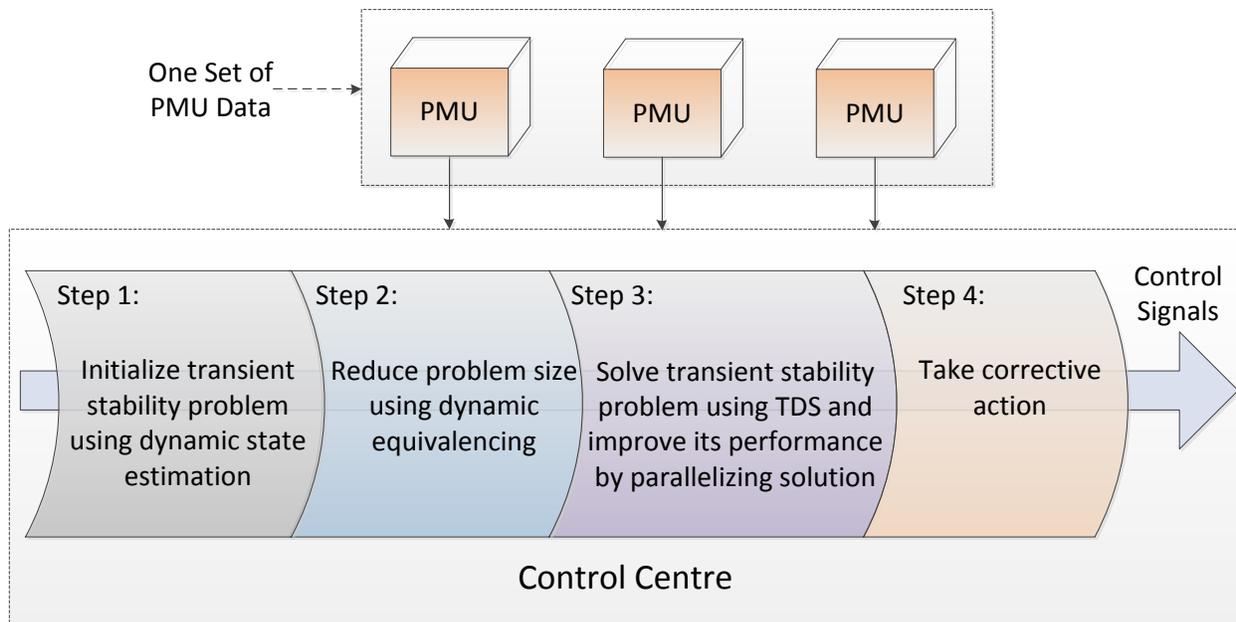
The research group at the University of South Florida Led by Dr. L. Fan have been using PMU measurements in UKF [64], Least Squares [85], and Finite Difference [86] approaches to estimate dynamic states and machine's parameters concurrently.

Pacific Northwest National Lab and its researchers led by Dr. N. Zhou have also made significant contributions with using PMU measurements in transient stability studies.

Initially, they proposed the ensemble Kalman filter [66] to estimate the system parameters as well as dynamic states. In the more recent publication [65], they have presented extended particle filter (EPF) which is claimed to be more robust than the EKF.

#### **1.4 Research Objectives and Anticipated Impacts**

According to the discussion presented in previous sections and for the purpose of this research, detailed structure of a TSM tool is depicted in Figure 1-2. This figure shows how the TSM utilizes one set of PMU data to predict system status in the next few seconds. As discussed, PMUs' measured quantities are initially converted into systems' state variables using a dynamic state estimation technique, and the results serve as initial conditions of the transient stability problem. Also, in order to improve computational performance of transient stability solver, dynamic equivalencing technique is performed at both transmission and generation levels to reduce size of power grid. The reduced-order system is then passed to a TDS package to solve the differential-algebraic equations using numerical integration. At this stage, partitioning techniques discussed in section 1.2.1.2 might be used to decompose original nonlinear DAEs into smaller sub-problems and share them among several processors to expedite solution speed. The TDS package then simulates the system for a short duration (3~5s) and if an instability is detected, an appropriate corrective action is chosen and required signals are sent to protection devices to stabilize the system.



**Figure 1-2 Detailed structure of a transient stability monitoring package.**

Based on the structure shown in Figure 1-2, this thesis considers following research objectives:

**Objective 1: Develop more accurate aggregation methods**

The state-of-the-art methods to aggregate coherent generators tend to keep dynamics of large generators since these generators provide large inertia and potentially a significant amount of power to the system. However, this methodology loses its accuracy if, for example, a large generator is not able to provide enough power in the post-fault state of the system. By considering structure of transmission system as well as machine parameters, this problem will be addressed here to provide a more accurate scheme for aggregating coherent generators.

Also, available methods [36]-[42] fail to create an accurate reduced-order system when the generators in a group are not tightly coherent, which shows that identifying and fixing errors in coherency identification is a key step in creating an accurate and reliable equivalent system. Therefore, this thesis aims at defining a new criterion for evaluating the accuracy of reduced-order system and identifying non-coherent generators.

Moreover, most of the available dynamic equivalencing methods solely rely on coherency identification and aggregation at generation level. However, because of the structural difference between generation and transmission systems, these algorithms may not be as efficient when applied to the transmission system. Therefore, locating coherent areas in the transmission system requires algorithms which are specifically designed for this system. As another direction in the first objective, this research aims at developing a new method for identifying and aggregating coherent areas in transmission system for transient stability studies.

## **Objective 2: Improve computational performance of transient stability solver**

The state-of-the-art parallel algorithms in transient stability area rely on parallelizable nonlinear solvers such as Gauss-Seidel [49], SOR [56], or Block Gauss-Jacobi [58]-[59] to utilize several processors for solving system of nonlinear equations. However, the convergence rate of these methods is much smaller than Full Newton and VDHN methods, which makes them inefficient unless the power network can be clearly partitioned into several areas. Moreover, considerable exchange of information between different processors, which translates into significant communication overhead cost, is a limiting

factor in number of processors employed in these methods. Therefore, it is very desirable to develop a new parallelizable algorithm whose sequential version is comparable to the VDHN method. Developing new schemes for decomposing the original nonlinear DAEs into smaller or linear sub-problems for using multi-core processors is considered in this thesis.

### **Objective 3: Integrate the PMU-measured quantities to initialize the TSA problem**

Using PMU measurements in TSM is still a developing area. The pioneering methods proposed in the literature for estimating internal states of synchronous machines [60]-[65] are suitable only for quasi-steady-state conditions, wherein no significant change occurs in the system variables. However, in transient stability study abrupt changes might occur in electrical quantities, which makes it challenging to estimate internal states using PMUs' measured quantities. This problem becomes more apparent when the PMU sampling rate becomes smaller than frequency of variations in the system. As the third objective, this research focuses on proposing a system-wide dynamic state estimation method to estimate the internal states of generators using PMU data, considering practical sampling rates (30~60 samples/sec). We particularly focus on the TSA problems, wherein the system is largely disturbed and the effects of nonlinear terms are significant.

The outcome of this research will be new methods that improve accuracy and efficiency of available transient stability solvers, and facilitate using advanced measurement technologies in the TSM packages. The results of this research will also help improving computational performance of available TDS methods, which is the main obstacle against realizing a TSM tool. In addition to monitoring applications, the developed tools can be used in offline and online studies. In the offline TSA, larger number of scenarios can be simulated and/or the length of each simulation can be extended. In the online TSA, it will become possible to perform the transient stability assessment more often. In all cases, improving performance of the TSA programs will have significant and beneficial impact on the system reliability.

# **CHAPTER 2: AGGREGATION OF GENERATORS AND BUSES FOR DYNAMIC EQUIVALENCING**

## **2.1 Aggregating Generators**

For the purpose of reducing the system size and aggregating the generators, an equivalent generator is typically created using a weighted average of the coherent generators, wherein the weight of each generator is determined by the inertia [35], [36]. This approach tends to keep dynamic characteristic of the large generators in the reduced-order system. This is a reasonable assumption since large generators are usually more influential on the overall system dynamics. However, this approach may not be accurate when a large generator is not able to deliver enough power in the post-fault system. Moreover, all available aggregation methods assume that the generators in a group are tightly coherent. Thus, if there is an error in coherency identification, the conventional aggregation methods [35]-[40] typically fail to give an accurate reduced-order system. This Chapter deals with challenges associated with creating an accurate and representative reduced-order system.

### **2.1.1 Formulation of the transient stability problem**

In power system, there exists a variety of dynamic sub-systems such as generators, excitation and turbine-governor systems, transmission lines, transformers, etc. The transients associated with these sub-systems are decomposed into fast (short-term) and slow (long-term) transients [87]. While studying long-term dynamics, the differential equations corresponding to short-term transients are replaced by the algebraic equilibrium

conditions [42], which helps to reduce the computational expense without significant loss of accuracy [87]. In transient stability analysis (TSA), the natural transients of network fall into category of fast transients and therefore they are represented by algebraic equations. The dynamics of generators' excitation systems and turbine-governor systems are considered as long-term transients, and these are represented by differential equations. Based on this decomposition, the formulation of TSA is described by set DAEs, which can be written in the following general form:

$$\dot{\mathbf{y}}_d = \mathbf{f}_d(\mathbf{y}_d, \mathbf{y}_a), \tag{2-1}$$

$$0 = \mathbf{f}_a(\mathbf{y}_d, \mathbf{y}_a),$$

where, vectors  $\mathbf{y}_d$  and  $\mathbf{y}_a$  represent dynamic and algebraic variables in the full-order system, respectively; vector function  $\mathbf{f}_d$  relates the time derivative of the dynamic variables to the dynamic and algebraic variables; and vector function  $\mathbf{f}_a$  represents the power flow algebraic constraints in the transmission system.

### 2.1.2 Inertial aggregation algorithm

The choice of aggregation of buses has direct effect on the accuracy of the reduced-order system. Generally, aggregating coherent generators can be performed either at internal nodes or at high voltage terminal buses. Since the coherent generators exhibit similar rotor angular swings, it is also appropriate to perform aggregation at the internal nodes. Moreover, in this section we focus on aggregating terminal buses and determining the weight of individual generators. However, if it is intended to keep the AVR and turbine-governor controls systems, they can be simplified and/or aggregated as well, for example

using the methods described in [40]-[42], [88]. Therefore, in the presence of the generators' control systems, the aggregation should be performed both at internal nodes as well as control systems.

The reduced-order system should preserve the steady state power flow of the original system. To ensure this condition, the full- and reduced-order systems should have the same power injected at high voltage terminals of each individual generator. The inertial aggregation algorithm (IAA) [36] is one of the well-established methods for aggregating classic generators, which addresses the mentioned issues. The steps included in the IAA and the overall procedure can be summarized as follows:

- a) Assume generators 1 to  $N$  constitute a group of coherent generators denoted by  $L$ .
- b) Calculate the initial complex phasor of the voltage behind reactance (VBR) of each generator in the group using the generators' injected power and currents right after fault is cleared.
- c) Compute the initial complex phasor of the internal voltage of the equivalent generator ( $\vec{E}_{qL}'$ ) using an inertial weighted average of the internal voltages of the generators in the group as:

$$\vec{E}_{qL}' = \frac{\sum_{l=1}^N M_l \vec{E}_{ql}'}{\sum_{l=1}^N M_l}, \quad (2-2)$$

where,  $M_l$  and  $\bar{E}_{qL}'$  represents the inertia and the initial VBR of generator  $l$ , respectively.

- d) Connect the internal bus of the equivalent generator to the internal bus of the individual generators using a transformer with the following complex voltage transformation ratio:

$$\bar{\alpha}_l = \frac{\bar{E}_{qL}'}{\bar{E}_{qL}'}, \quad (2-3)$$

This transformer is introduced to ensure that the power flow is the same in full- and reduced-order systems.

These steps are summarized in Figure 2-1. As can be seen in Figure 2-1 (ii), the internal bus of the equivalent generator is connected to more than one bus, which is not consistent with the available model for the classic generators. In order to be consistent with the available models and to avoid introducing a zero impedance link, [36] extends bus  $p$  to bus  $q$  using impedance  $-jx'_{d,eq}$ . Here,  $x'_{d,eq}$  represents the transient reactance of the equivalent generator and it is computed as follows:

$$x'_{d,eq} = \left( \sum_{l=1}^N (x'_{dl})^{-1} \right)^{-1}. \quad (2-4)$$

Then, the bus  $q$  is connected to the internal bus of the equivalent generator.

In the IAA method, the common bus is created using an inertial weighted average of the coherent generators, wherein the weight of each generator is determined by its inertia. In

this approach, the contribution of small generators becomes negligible and the equivalent generator mainly represents the dynamic characteristics of large generators. The IAA gives accurate results in most cases since the small generators have negligible impact on the overall system. However, in addition to the inertia of the generators, structure of transmission network also affects the overall system dynamics. Neglecting the effect of the transmission network in the IAA can lead to a sub-optimal reduced-order system, especially when the effect of small generators is considerable or the generators in a group are not tightly coherent. In this situation, the IAA may fail to give accurate results.

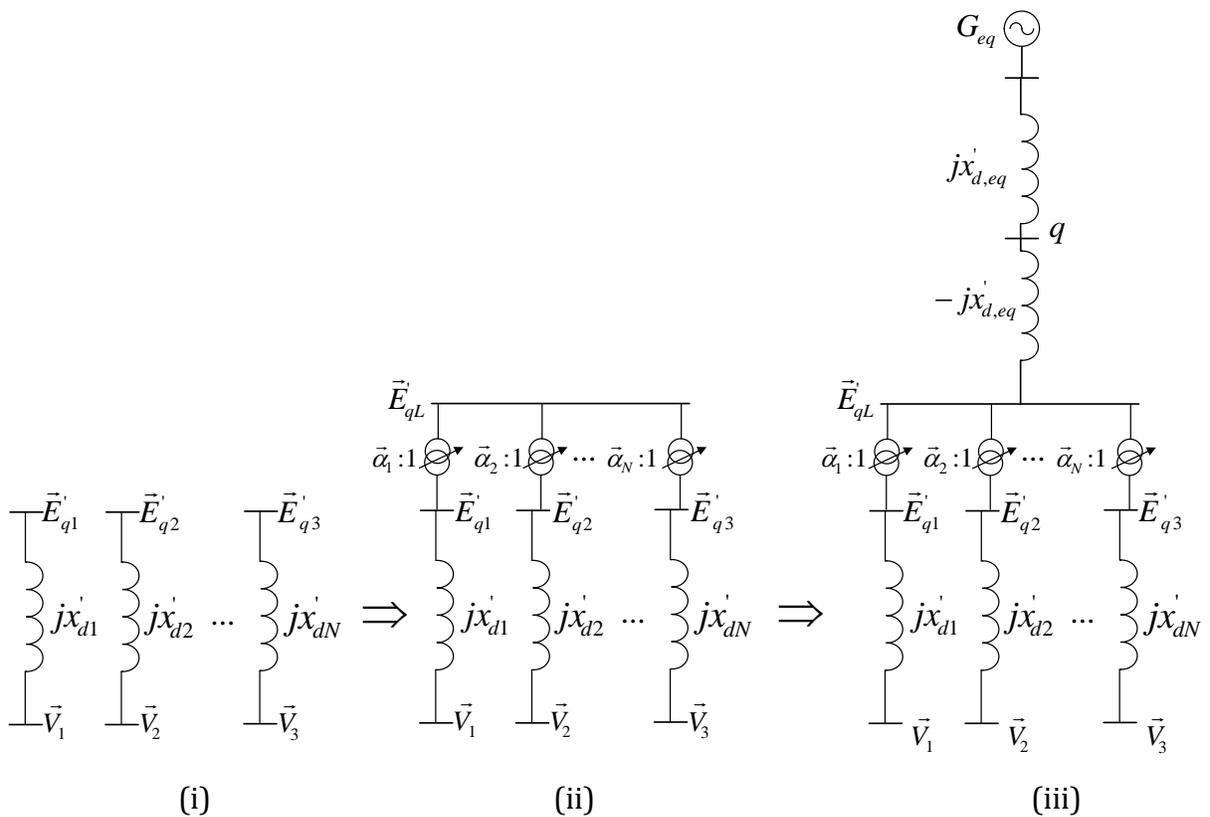


Figure 2-1 The procedure of creating an equivalent generator in the IAA.

### 2.1.3 Network-dependent aggregation (NDA) technique for finding generators weights

Suppose generator  $l$  belongs to group  $L$ . Moreover, suppose that all nonlinear loads (i.e. constant current and constant power loads) are linearized around the initial conditions corresponding to the post-fault condition of the system. The linearized loads as well as constant impedance loads and transient reactances of the generators are then included into the network admittance matrix (Y-Bus), which makes it possible to eliminate the load buses from the system using Kron method [89] to obtain the reduced admittance matrix ( $\mathbf{Y}$ ). Using these steps and assumptions, the electromechanical dynamics of the generator  $l$  is described by:

$$\Delta \dot{\delta}_l = \omega_l, \quad (2-5)$$

$$M_l \dot{\omega}_l = P_{ml} - P_{el},$$

where

$$P_{el} = \Re \left\{ \vec{E}_{ql}' e^{j\Delta\delta_l} \sum_{i=1}^{n_b} \left( \vec{Y}_{li} \vec{E}_{qi}' e^{j\Delta\delta_i} \right)^* \right\}. \quad (2-6)$$

In (2-5) and (2-6),  $M_l$  and  $P_{ml}$  represent the generator's inertia and input mechanical power;  $n_b$  represents number of generators in the system;  $\vec{Y}_{li}$  represents the admittance between generators  $l$  and  $i$  in the reduced Y-Bus;  $\omega_l$  represents rotor angular speed of generator  $l$ ;  $\Delta\delta_l$  represents the change in the rotor angle with respect to the initial angle of the generator; and  $\Re\{\}$  represents the real part of a complex number. Equation (2-6) calculates the power injected at bus  $l$  in the full-order system. In an accurate reduced-order

system, the power injected at bus  $l$  should be as close as possible to (2-6). Using Figure 2-1 (ii), the injected power at bus  $l$  in the reduced-order system is calculated as:

$$P_{el}^R = \Re \left\{ \frac{\bar{E}_{qL}'}{\bar{\alpha}_l} e^{j\Delta\delta_{eq}} \sum_{i=1}^{n_b} \left( \bar{Y}_{li} \bar{E}_{qi}' e^{j\Delta\delta_i} \right)^* \right\}, \quad (2-7)$$

where  $\Delta\delta_{eq}$  represents the rotor angle of the equivalent generator with respect to its initial angle;  $P_{el}^R$  represents the electrical power injected at bus  $l$  in the reduced-order system. By substituting (2-3) into (2-7),  $P_{el}^R$  can be rewritten in terms of the rotor angle of the equivalent generator ( $\Delta\delta_{eq}$ ) and the voltage behind transient reactance of generator  $l$  ( $\bar{E}_{ql}'$ ) as:

$$P_{el}^R = \Re \left\{ \left( \bar{E}_{ql}' e^{j\Delta\delta_{eq}} \sum_{i=1}^{n_b} \left( \bar{Y}_{li} \bar{E}_{qi}' e^{j\Delta\delta_i} \right)^* \right) \right\}. \quad (2-8)$$

Comparing (2-8) with (2-6), it can be seen that  $\Delta\delta_l$  is replaced by  $\Delta\delta_{eq}$ . If all generators in the group are tightly coherent the following condition can be assumed:

$$\Delta\delta_1 \approx \Delta\delta_2 \approx \dots \approx \Delta\delta_N \approx \Delta\delta_{eq}, \quad (2-9)$$

which implies that  $P_{el}^R \approx P_{el}$ . Therefore, if all generators in a group are tightly coherent, the reduced-order system will be accurate. However, if the generators are not tightly coherent, (2-9) does not hold and the accuracy of the reduced-order system degrades. In this situation, inappropriate selection of generators' weights further contributes to degradation of the accuracy of the reduced-order system. In order to see how deviation from (2-9) can reduce the accuracy of the reduced-order system, consider the mismatch  $P_{el} - P_{el}^R$ :

$$\varepsilon_l = P_{el} - P_{el}^R = \Re \left\{ \left( e^{j\Delta\delta_l} - e^{j\Delta\delta_{eq}} \left( \vec{E}_{ql}' \sum_{i=1}^{n_b} \left( \vec{Y}_{li} \vec{E}_{qi}' e^{j\Delta\delta_i} \right)^* \right) \right) \right\}, \quad (2-10)$$

where  $\varepsilon_l$  represents the difference between the active power injected at bus  $l$  in the full- and reduced-order systems. This equation shows that the mismatch  $P_{el} - P_{el}^R$  is caused by the difference between  $e^{j\Delta\delta_l}$  and  $e^{j\Delta\delta_{eq}}$ , and it is magnified by the following term:

$$MF_l = \vec{E}_{ql}' \sum_{i=1}^M \left( \vec{Y}_{li} \vec{E}_{qi}' e^{j\Delta\delta_i} \right)^*. \quad (2-11)$$

In (2-11),  $MF_l$  is called the magnification factor of generator  $l$  and it shows the sensitivity degree of this generator to the mismatch  $e^{j\Delta\delta_l} - e^{j\Delta\delta_{eq}}$ . The generators with large magnification factor are more sensitive to deviation from the solution of the full-order system. These generators may cause larger errors if the reduced-order system is not accurate, and therefore, they need to be modeled more accurately. The magnification factors can also be seen as the capability of generators for injecting power. Typically, large generators have larger VBR and larger elements in the corresponding row in the admittance matrix. This also implies that large generators typically have larger magnification factors.

Based on this observation, the magnification factors can be used to find the weights of generators. However,  $\Delta\delta_i$  is not constant in (2-11), which implies that the magnification factors are time-dependent parameters. In order to find the weights of generators,  $\Delta\delta_i$  can be ignored in (2-11). Using this assumption, an estimate can be derived as:

$$\tilde{M}\tilde{F}_l = \vec{E}'_{ql} \sum_{i=1}^M (\vec{Y}_{li} \vec{E}'_{qi})^* \quad (2-12)$$

In (2-12),  $\tilde{M}\tilde{F}_l$  represents an estimation of  $MF_l$ . Using (2-12), the weight of each generator in group  $L$  can be defined as:

$$w_k = \frac{|\tilde{M}\tilde{F}_k|}{\sum_{l=1}^N |\tilde{M}\tilde{F}_l|}, \quad (2-13)$$

where  $w_k$  is the weight of generator  $k$  in the proposed NDA method. Equation (2-13) defines the weight of generator  $k$  in the corresponding group of coherent generators. Thus, unlike the IAA, the proposed NDA ranks the generators based on their sensitivity to the error in reduced-order system. Considering (2-12), it can be seen that  $\tilde{M}\tilde{F}_l$  also represents the injected power by generator  $l$  right after the fault is cleared. Therefore, the proposed NDA approach assigns large weights to the generators which are capable of injecting more power right after the fault.

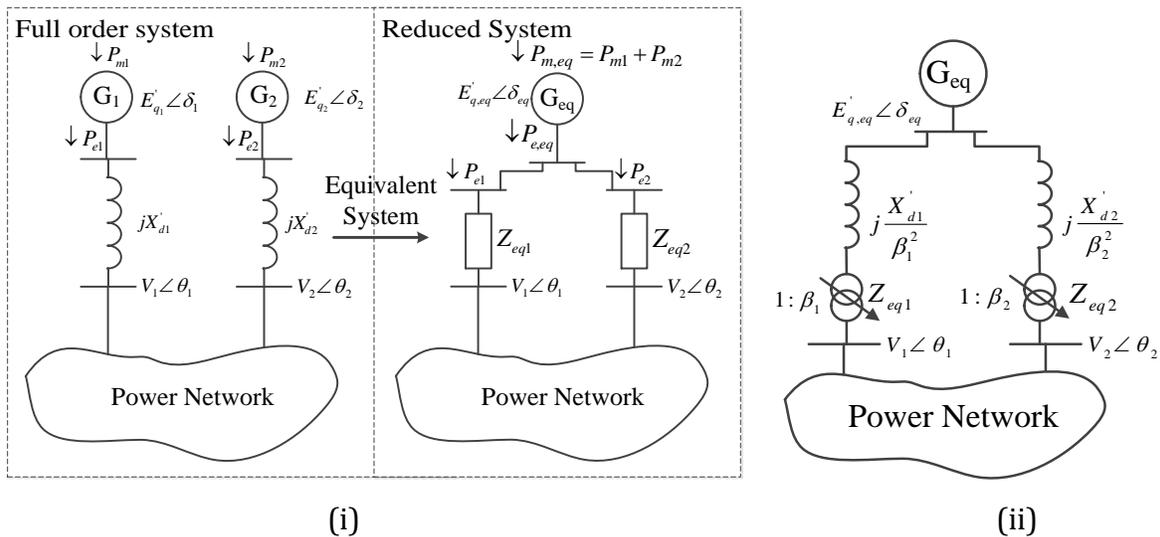
#### 2.1.4 Aggregating coherent generators

In this section, it is demonstrated how the weights calculated in the previous section are used to aggregate the coherent terminal buses of the generators. Suppose that the generators  $G_1$  and  $G_2$  in Figure 2-2 are coherent. The coherency condition between these generators can be expressed in terms of their voltage behind reactance as follows

$$\frac{E'_{q1} e^{j\delta_1}}{E'_{q2} e^{j\delta_2}} = \bar{c}_0 + \varepsilon(t), \quad l_1, l_2 \in L \quad (2-14)$$

where  $\delta_{l_1}$  and  $\delta_{l_2}$  represent the rotor angles of generators  $l_1$  and  $l_2$ , respectively;  $\vec{c}_0$  is a constant complex number; and  $\varepsilon$  represents the error in the coherency between these two generators. If the generators are tightly coherent,  $\varepsilon$  is negligible and the ratio between the VBR of the coherent generators becomes constant for all times. Moreover, it is commonly assumed that within each coherent group, local and inter-area oscillations are neglected, which implies that all the generators within a group have identical rotor speed as:

$$\omega_{eq} = \omega_1 = \omega_2 = \dots = \omega_N, \quad (2-15)$$



**Figure 2-2 Creating an equivalent generator from two coherent generators.**

The equivalent generator can be created using a weighted average of the coherent generators in the group, where the weight of each generator is determined by the method presented in the previous section. Using this methodology, the parameters of the equivalent generator are calculated as follows:

$$\Delta\delta_{eq} = \sum_{l=1}^N w_l \Delta\delta_l, \quad (2-16)$$

$$E'_{q,eq} = \sum_{l=1}^N w_l E'_{ql}. \quad (2-17)$$

Substituting the rotor speed dynamics (2-5) into (2-15), the rotor dynamics of the equivalent generator can be computed as follows:

$$\Delta\dot{\delta}_{eq} = \dot{\omega}_{eq}, \quad (2-18)$$

$$M_{eq}\dot{\omega}_{eq} = M_{eq} \sum_{l=1}^N \frac{w_l}{M_l} (P_{ml} - P_{el}), \quad (2-19)$$

$$M_{eq} = \sum_{l=1}^N M_l.$$

Equation (2-19) can be simplified,

$$M_{eq}\dot{\omega}_{eq} = P_{m,eq} - P_{e,eq}, \quad (2-20)$$

$$P_{m,eq} = \sum_{l=1}^N \frac{w_l M_{eq}}{M_l} P_{ml}, \quad P_{e,eq} = \sum_{l=1}^N \frac{w_l M_{eq}}{M_l} P_{el}.$$

In (2-20),  $P_{m,eq}$  and  $P_{e,eq}$  denote the aggregated input mechanical power and output electrical power, respectively. Equations (2-18) and (2-20) describe the parameters and the dynamics of the equivalent generator.

The concept of aggregation of terminal buses is shown in Figure 2-2, which is based on the terminal buses aggregation shown in Figure 2-1. The equivalent generator receives the aggregated input mechanical power and provides the electrical power  $P_{e,eq}$ , which is sent

to the terminals of the coherent generators in the corresponding group. The total power transferred to the terminal buses in the full-order system is equal to:

$$\bar{S}_{total} = \sum_{l=1}^N \bar{S}_l = \sum_{l=1}^N V_l e^{j\theta_l} \left( \frac{\bar{E}'_{ql} e^{-j\Delta\delta_l} - V_l e^{-j\theta_l}}{-jX'_{dl}} \right). \quad (2-21)$$

If all generators are tightly coherent, the VBR of the individual generators can be written in terms of the VBR of the equivalent generator as:

$$\bar{E}'_{ql} e^{j\Delta\delta_l} = \bar{\beta}_l \bar{E}'_{q,eq} e^{j\Delta\delta_{eq}}, \quad (2-22)$$

$$\bar{\beta}_l = \frac{\bar{E}'_{q,eq}}{\bar{E}'_{ql}}.$$

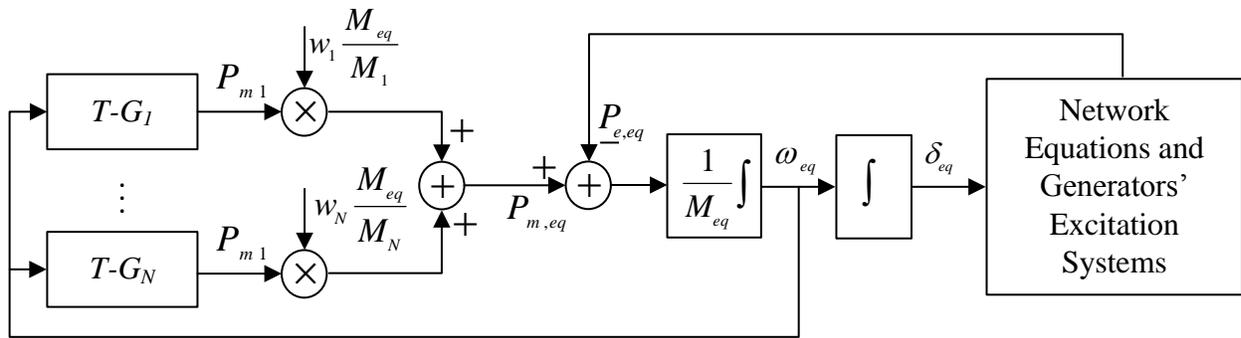
Using (2-9) and (2-22), one can rewrite (2-21) in terms of parameters of the equivalent generator as

$$\bar{S}_{total} = \sum_{l=1}^N \bar{S}_l = \sum_{l=1}^N V_l e^{j\theta_l} \left( \frac{\bar{\beta}_l \bar{E}'_{eq} e^{-j\Delta\delta_{eq}} - V_l e^{-j\theta_l}}{-jX'_{dl}} \right). \quad (2-23)$$

Equation (2-23) represents the total power transferred from the equivalent generator to the individual buses. Presence of  $\bar{\beta}_l$  implies that a transformer/phase-shifter is required to connect the equivalent bus to the internal buses of the coherent generators, as shown in Figure 2-2 (ii).

The block diagram representation of the reduced-order system [described by (2-15)-(2-23)] is shown in Figure 2-3, where the aggregation of one group of coherent generators is shown as an example. In Figure 2-3,  $T - G_i$  represents turbine-governor control system of

the generator  $i$ , which uses the equivalent rotor speed to adjust input mechanical power. If the turbine-governor control system is neglected, the input mechanical power will be constant. While the excitation systems of the generators are not shown in Figure 2-3, they can be maintained and simplified/aggregated using the methods described in [40]-[42], [88]. These dynamics along with network equations are included in the block marked as “Network Equations and Generators’ Excitation Systems” in Figure 2-3.



**Figure 2-3 Block diagram of the aggregated group of coherent generators when local and inter-area oscillations are neglected within the coherent group.**

### 2.1.5 Errors in coherency identification

Equations (2-21) and (2-23) establish connection between the power injected in the terminals of coherent generators in the full- and reduced-order systems. These equations are equivalent as long as the assumption (2-9) holds. This is the underlying assumption in all available aggregation methods and its accuracy highly depends on the coherency between the generators in a group. If the generators are not tightly coherent, (2-9) is not a valid assumption, and consequently, the reduced-order system will not be valid. In this

situation, the reduced-order system might lead to invalid results. For example, the generation limits derived from an inaccurate reduced-order system might be too conservative (which imposes additional economical cost), or be a bit larger than the real limit (which can lead to angular instability in the system in case of a real contingency). Therefore, it is of significant importance to monitor the accuracy of the reduced-order system and ensure that all the generators within a group are tightly coherent. In this section, the problem of coherency identification error is addressed, and an algorithm is proposed to identify the errors in coherency identification.

As discussed before, the full-order system of the transient stability problem consists of DAEs described by (2-1). In order to solve the DAEs, an implicit integration method such as backward Euler or trapezoidal rule is typically used to discretize differential equations, which results in a system of nonlinear equations. For example, discretizing (2-1) using trapezoidal rule yields the following nonlinear equation:

$$\mathbf{h}(\mathbf{y}_d^{n+1}, \mathbf{y}_a^{n+1}, \mathbf{y}_d^n, \mathbf{y}_a^n) = \begin{bmatrix} \mathbf{y}_d^{n+1} - \mathbf{y}_d^n - \frac{T}{2} (\mathbf{g}_d(\mathbf{y}_d^{n+1}, \mathbf{y}_a^{n+1}) + \mathbf{g}_d(\mathbf{y}_d^n, \mathbf{y}_a^n)) \\ \mathbf{g}_a(\mathbf{y}_d^{n+1}, \mathbf{y}_a^{n+1}) \end{bmatrix} = 0, \quad (2-24)$$

where  $\mathbf{h}$  is called the mismatch vector;  $T$  and  $n$  denote the integration time-step and the current integration step, respectively. Equation (2-24) should be solved iteratively to find  $\mathbf{y}_d^{n+1}$  and  $\mathbf{y}_a^{n+1}$ . Using iterative solvers such as Newton-Raphson or SOR [89], it is ensured that the following condition is met in each integration step:

$$\|\mathbf{h}(\mathbf{y}_d^{n+1}, \mathbf{y}_a^{n+1}, \mathbf{y}_d^n, \mathbf{y}_a^n)\|_{\infty} < \varepsilon_T, \quad (2-25)$$

where  $\varepsilon_T$  represents the tolerance; and  $\|\cdot\|_\infty$  represents the infinite norm of a vector.

Since differential equations of coherent generators are aggregated in the reduced-order system,  $\mathbf{y}_d$  is only partially available, and therefore (2-25) cannot be checked in the reduced-order system to verify the solution. However, if the generators are tightly coherent, it can be assumed that (2-9) holds, and neglecting local and inter-area oscillations within a coherent group is a valid assumption. In this situation, the slow dynamics of individual generators in the full-order system can be reconstructed assuming that (2-9) and (2-15) are valid. For example, the rotor angles and rotor speeds of the individual generators in the full-order system can be reconstructed using the following:

$$\tilde{\delta}_l = \delta_{l0} + \Delta\delta_{eq}, \quad (2-26)$$

$$\tilde{\omega}_l = \omega_{eq}$$

where  $\tilde{\delta}_l$  and  $\tilde{\omega}_l$  represent the reconstructed rotor angle and rotor speed of generator  $l$ . Equation (2-26) describes the dynamic variables of the individual generators in terms of the corresponding equivalent generator. Using (2-26), the state variables of the individual generators in the full-order system can be estimated and a good estimation is achieved only when the reduced-order system is accurate. Suppose that  $(\tilde{\mathbf{y}}_d^{n+1}, \tilde{\mathbf{y}}_a^{n+1})$  represents the reconstructed state of the full-order system. If the reduced-order system is accurate, this solution will be close to the solution of the full-order system, and accordingly,  $(\tilde{\mathbf{y}}_d^{n+1}, \tilde{\mathbf{y}}_a^{n+1})$  along with  $(\tilde{\mathbf{y}}_d^n, \tilde{\mathbf{y}}_a^n)$  will satisfy the following condition:

$$\|\mathbf{h}(\tilde{\mathbf{y}}_d^{n+1}, \tilde{\mathbf{y}}_a^{n+1}, \tilde{\mathbf{y}}_d^n, \tilde{\mathbf{y}}_a^n)\|_\infty < \varepsilon_T, \quad (2-27)$$

where  $\varepsilon_T$  denotes the specified error tolerance. However, if one or more generators are not tightly coherent in their group, the reduced-order system will not be accurate, and (2-27) will not be satisfied. In this situation, it is required to identify the non-coherent generators and separate them from the rest of the group, which can be carried out by analyzing  $\mathbf{h}(\tilde{\mathbf{y}}_d^{n+1}, \tilde{\mathbf{y}}_a^{n+1}, \tilde{\mathbf{y}}_d^n, \tilde{\mathbf{y}}_a^n)$ . If the reduced-order system is not accurate due to an error in coherency identification, some of the elements of  $\mathbf{h}(\tilde{\mathbf{y}}_d^{n+1}, \tilde{\mathbf{y}}_a^{n+1}, \tilde{\mathbf{y}}_d^n, \tilde{\mathbf{y}}_a^n)$  will be large, which shows that reduced-order system is inaccurate. More importantly, the error mainly appears in the variables corresponding to the misplaced generators, making it is possible to identify the source of inaccuracy in the reduced-order system. The block diagram of this approach is shown in Figure 2-4. According to this approach, the following steps are taken to ensure that the reduced-order system is accurate:

1. Initially, an integration rule is chosen to integrate reduced-order system. The selected integration rule can be implicit or explicit and might use fixed or variable time-step.
2. Using the selected integration criterion, the solution of the reduced-order system is found in each integration time-step, which is denoted by  $(\hat{\mathbf{y}}_d^{n+1}, \hat{\mathbf{y}}_a^{n+1})$  in Figure 2-4.
3. The rotor angle and speed of the individual aggregated generators are reconstructed using (2-26). In this stage, if excitation and/or turbine-governor systems of the aggregated generators are also considered, the state variables corresponding to

these systems should be reconstructed according to the assumptions made to simplify/aggregate the control systems. This step yields an estimate of the state vector of the full-order system, denoted by  $(\hat{\mathbf{y}}_d^{n+1}, \hat{\mathbf{y}}_a^{n+1})$ .

4. Using the integration rule (step 1) and time-step (step 2), the differential equations are discretized using results from step 3. The discretized differential equations along with algebraic equations form the mismatch vector  $\mathbf{h}(\tilde{\mathbf{y}}_d^{n+1}, \tilde{\mathbf{y}}_a^{n+1}, \tilde{\mathbf{y}}_d^n, \tilde{\mathbf{y}}_a^n)$ .
5. The elements of  $\mathbf{h}(\tilde{\mathbf{y}}_d^{n+1}, \tilde{\mathbf{y}}_a^{n+1}, \tilde{\mathbf{y}}_d^n, \tilde{\mathbf{y}}_a^n)$  which do not satisfy (2-27) are identified, and the corresponding generators are separated from their groups.
6. The groups which have been modified are updated, and a new equivalent generator is calculated for each group.

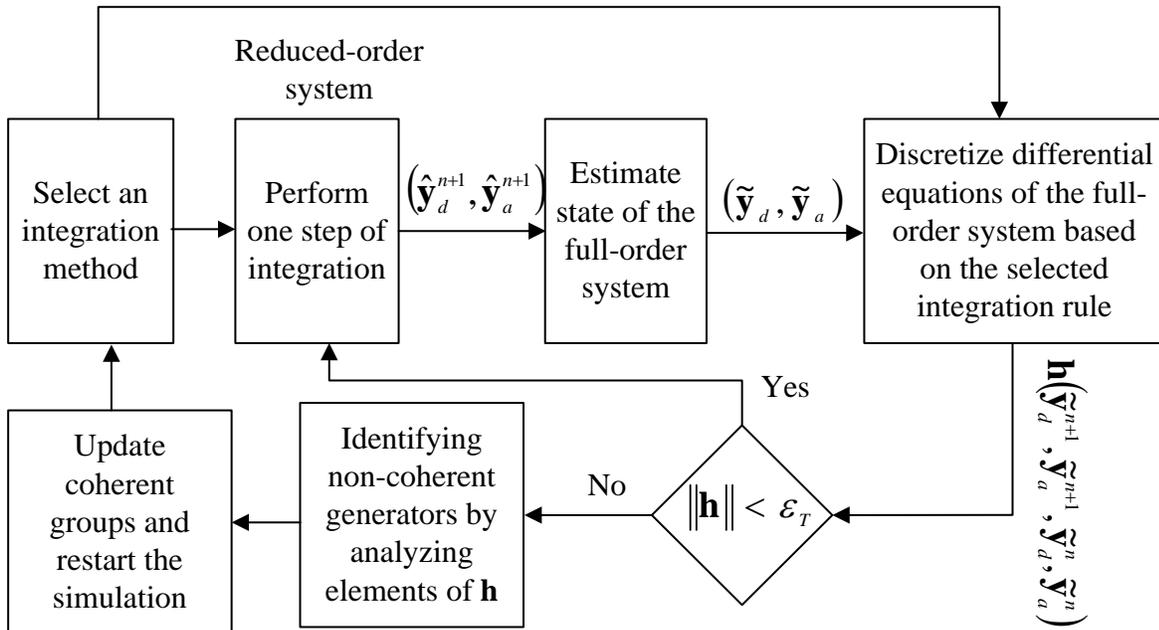


Figure 2-4 Block diagram of the proposed error identification algorithm.

Using the proposed algorithm, it is ensured that the reduced-order system is accurate within the specified tolerance denoted by  $\varepsilon_T$ . It is appropriate to check (2-27) at each integration step for the first 10 to 100 integration steps. If any misplaced generators are identified at this stage, the simulation is restarted using the updated groups. However, sometimes the reduced-order system is accurate for the first 2~3 seconds and then it gradually starts to deviate from the solution of the full-order system. In this case, it might not be efficient to check the validity of the solution at each integration step as computation of  $\mathbf{h}(\tilde{\mathbf{y}}_d^{n+1}, \tilde{\mathbf{y}}_a^{n+1}, \tilde{\mathbf{y}}_d^n, \tilde{\mathbf{y}}_a^n)$  is expensive. In this case, (2-27) may be checked once in several integration steps. If (2-27) is not satisfied, the simulation is restarted using the last verified step.

Since the oscillations among coherent generators are neglected in the reduced-order system, the solution found by integrating the reduced-order system is not as accurate as the solution derived by solving the full-order system. Therefore, the error tolerance  $\varepsilon_T$  used in (2-27) should be generally larger than the error tolerance used in integration of the full-order system. Moreover, it should be noted that while the trapezoidal rule is used in (2-24) to construct  $\mathbf{h}$ , the proposed methodology is not limited to trapezoidal rule only. In fact, as Figure 2-4 suggests, the same integration rule and the same time-step should be used to integrate the reduced-order system and in the proposed error identification algorithm. Finally, it is worth to mention that if there are several non-coherent generators in a group, then a large number of variables will be affected. In this situation, it might be hard to identify the misplaced generators and separate them from the rest of the group. Therefore, the proposed algorithm works best when there are only few misplacements.

### 2.1.6 Simulation results

The IEEE transient stability test system with 50 generators and 145 buses, presented in [91], has been considered here to evaluate the effectiveness of the proposed adaptive aggregation method. In this system, 44 generators are represented using the simplified/classical model, and the rest are represented using the two-axes model equipped with Type AC-4 exciter [92]. A 4th order model with one damper winding on the rotor [93] is used to simulate the detailed generators. The Powertech Labs DSATools is used to generate the time-domain reference solution. For comparison purposes, the proposed network-dependent aggregation algorithm described in sections 2.1.3 and 2.1.4 and the inertial aggregation algorithm described in section 2.1.2 are also coded into MATLAB, which are referred to as NDA and IAA, respectively. A three-phase fault is assumed at bus #7 at  $t = 1s$ . The fault is cleared in 8 cycles by opening the transmission line between buses #7 and #6. The integration time-step is  $10^{-2}s$  and the post-fault simulation time is 10 seconds. It should be mentioned that within the scope of this chapter and simulation studies, it is sufficient to choose the same integration time-step in all three methods: the benchmark solution (DSATools), IAA, and the NDA. Moreover, our simulations show that in the studied system, choosing 10ms time-step covers all dynamics of interest. Additionally, in order to identify coherent generators, the rotor angle swings generated by the DSATools are compared to identify the coherent generators. While this method is tedious, it is assumed to be the most reliable for the purpose of the studies presented here. Also, in the proposed error identification algorithm, the tolerance  $\varepsilon_T$  is set to  $10^{-2}$ .

In the first study, only tightly coherent generators are identified. Table 2-1 presents the results, wherein each curly bracket represents a group of coherent generators, and the numbers inside bracket represent the bus number of each generator. For example, {79, 80} represent a group of coherent generators located at buses #79 and #80. The groups of coherent generators are then used in the NDA and IAA to create the reduced-order systems. In order to compare the results, numerical integration is employed to solve the reduced-order systems. In the test system, the generator #60 is largely affected by the fault since it is very close to the fault location. The rotor angle of generator #60 is plotted in Figure 2-5. As shown in Figure 2-5, the rotor angle is initially in steady state for the first second. As soon as the fault happens, the rotor angle starts to accelerate. After the fault is cleared, the rotor angle oscillates, and the oscillations persist until the end of the simulation indicating that the system has very small damping. This figure shows that both aggregation methods are able to generate sufficiently accurate reduced-order systems and the results are close to reference solution obtained using the DSATools throughout the study. In order to further compare the methods, two error terms are defined as:

$$e_{NDA} = \frac{\|\delta_{DSA}(t) - \delta_{NDA}(t)\|_2}{\|\delta_{DSA}(t)\|_2} \times 100, \quad (2-28)$$

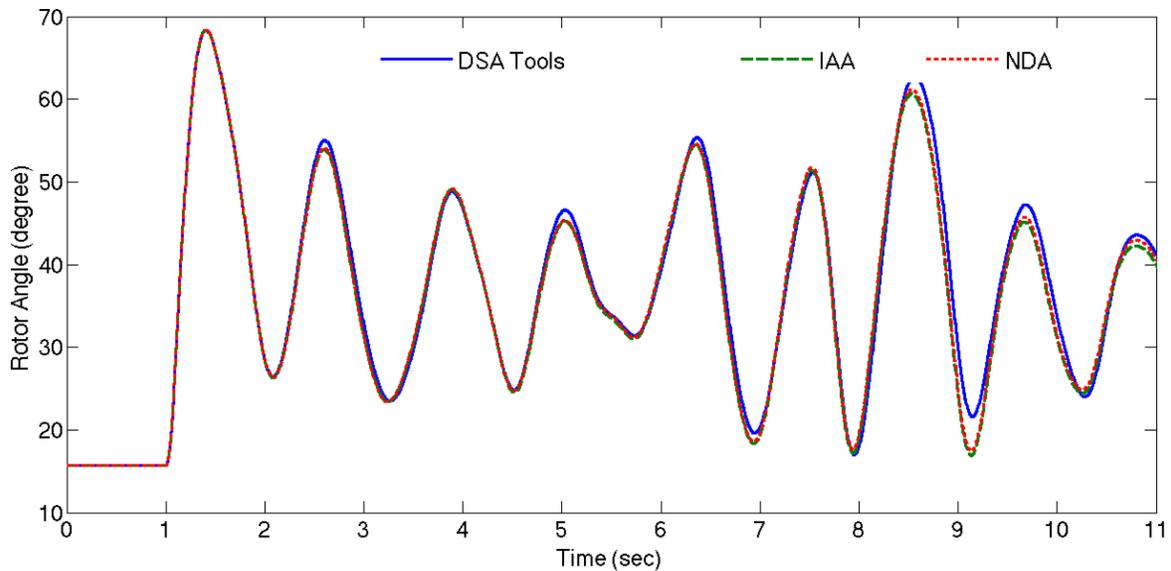
$$e_{IAA} = \frac{\|\delta_{DSA}(t) - \delta_{IAA}(t)\|_2}{\|\delta_{DSA}(t)\|_2} \times 100,$$

where  $\delta_{DSA}(t)$ ,  $\delta_{NDA}(t)$ , and  $\delta_{IAA}(t)$  represent rotor angles calculated by the DSATools, the proposed aggregation algorithm, and the inertial aggregation algorithm, respectively;  $e_{NDA}$  and  $e_{IAA}$  represent the percentage error of the proposed NDA and the inertial aggregation

algorithms, respectively. For the rotor angles shown in Figure 2-5,  $e_{NDA} = 0.38\%$  and  $e_{IAA} = 0.42\%$ , which shows that NDA slightly outperforms IAA.

**Table 2-1 The Groups of Tightly Coherent Generators**

Group Number	Generators' bus numbers
1	{79, 80}
2	{93, 110}
3	{82, 109}
4	{101, 112}
5	{105, 106}
6	{67, 97}
7	{108, 121}



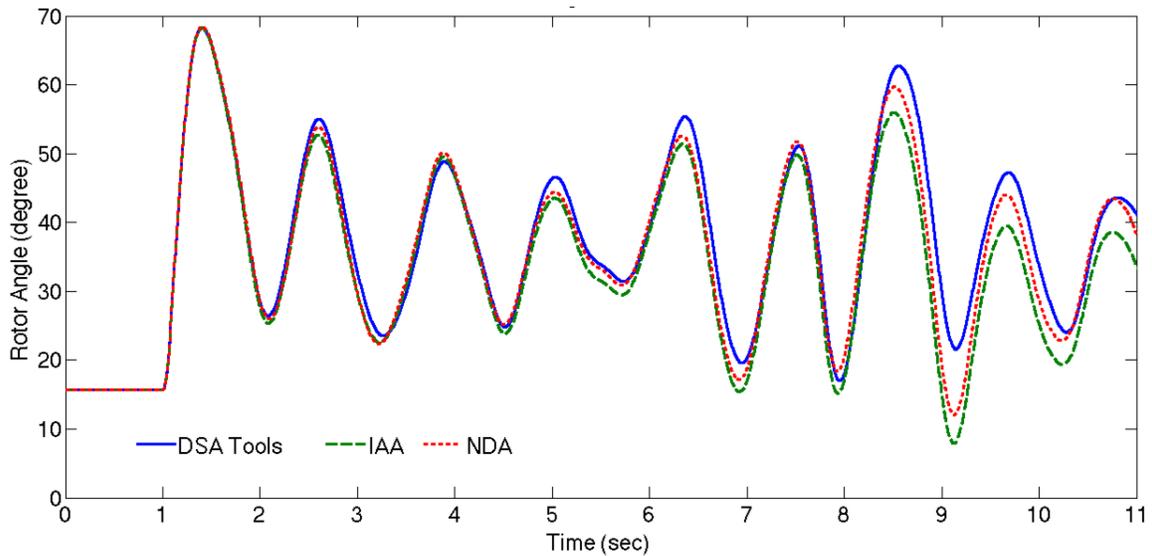
**Figure 2-5 Rotor angle of generator #60 in the first study.**

In the second study, the generators #124 and #91 are added to groups 6 and 7, respectively, to create new coherent groups shown in Table 2-2. These generators are less

coherent with the rest of the group and accordingly, it is expected that the resulting reduced-order system will be less accurate. The coherent groups of generators in Table 2-2 are used to create reduced-order systems. The rotor angle of the generator located at bus #60 is depicted in Figure 2-6, where it can be seen that while the accuracy of the reduced-order system has degraded compared to Figure 2-5, the NDA is still able to create a more accurate reduced-order system and its solution is closer to that predicted by the DSATools. Calculating the error shows that  $e_{NDA} = 0.79\%$  and  $e_{IAA} = 1.29\%$ , which implies that the NDA is almost 63% more accurate than the IAA.

**Table 2-2 The Groups of Coherent Generators in the Second Study**

Group Number	Generators' bus numbers
1	{79, 80}
2	{93, 110}
3	{82, 109}
4	{101, 112}
5	{105, 106}
6	{67, 97, 124}
7	{91, 108, 121}



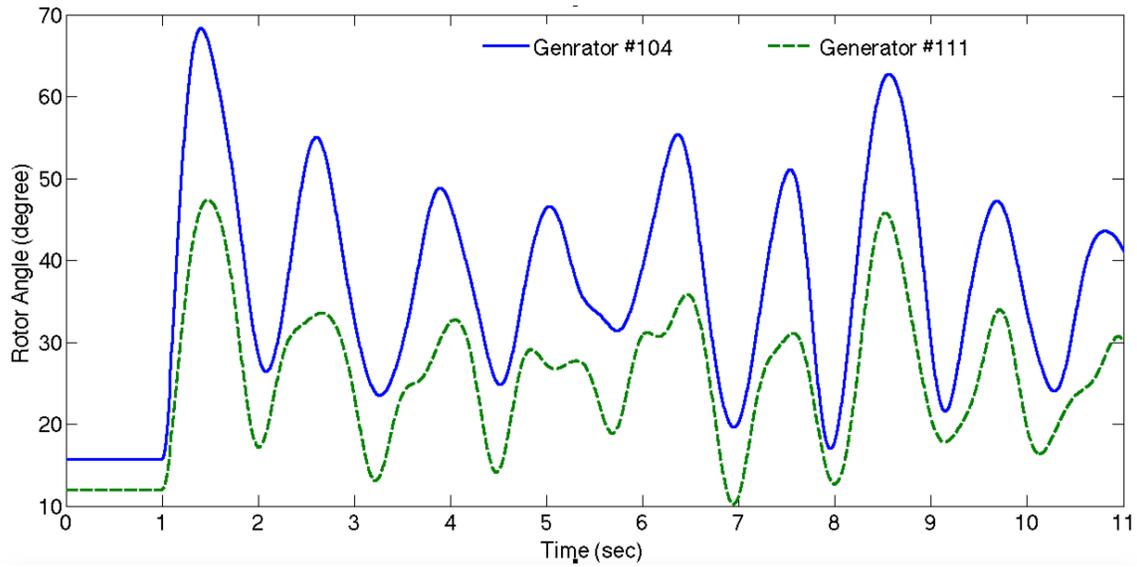
**Figure 2-6 Plot of rotor angle of generator #60 in the second study.**

In the third study, a new group consisting of generators #104 and #111 is added to Table 2-2. The resulting coherent groups of generators are shown in Table 2-3. In this test system, the generators #104 and #111 are coherent in most scenarios since they are physically close. However, when a fault happens at bus #7, the generators are no longer coherent as both generators are largely affected by the fault. To demonstrate this point, the rotor angles of these generators are plotted in Figure 2-7. While both generators tend to oscillate together, the generators #104 and #111 are not coherent since the difference between their rotor angle varies up to  $20^\circ$ . The coherent groups shown in Table 2-3 are used in the NDA and the IAA to create reduced-order systems, and the results are shown in Figure 2-8. As can be seen, while the NDA still outperforms the IAA, the accuracy of the resulting reduced-order systems is not acceptable. The reason is that the aggregation methods are based on assumption that (2-9) holds and all groups of generators are tightly coherent. However, due to nonlinear nature of the transient stability problem, the

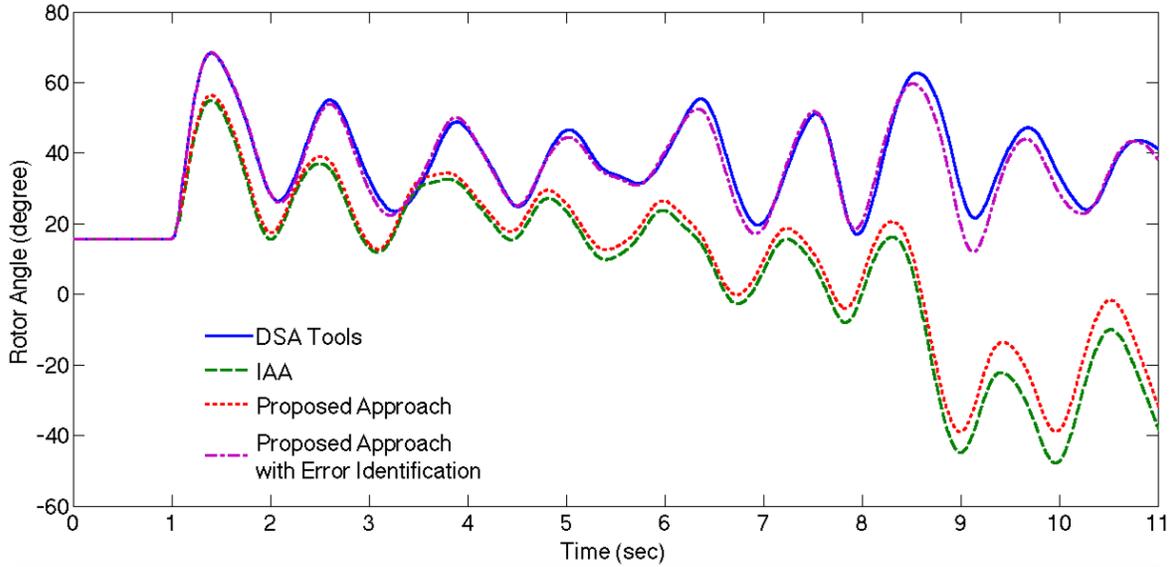
aggregation algorithms are very sensitive to this assumption. Introducing a group of non-coherent generators invalidates this assumption and leads to an inaccurate reduced-order system. In this situation, it is necessary to identify the non-coherent generators and separate them from their groups. Therefore, the error identification algorithm presented in Section 2.1.5 is used to monitor the accuracy of the reduced-order system. By continuously checking (2-27), the proposed algorithm detected that the mismatches corresponding to dynamic and algebraic equations of the generators #104 and #111 are much larger than  $\varepsilon_T$ , which denotes that these two generators are not coherent. Therefore, generators #104 and #111 were separated and the simulation was restarted. The results are shown in Figure 2-8, where it can be seen that the proposed adaptive algorithm has produced more accurate results. It should be stressed that as Figure 2-8 shows, a coherency identification error affects the solution of all the variables in the reduced-order system, which implies that in the presence of non-coherent generators, the solution of the reduced-order system is not reliable. The traditional aggregation methods cannot identify errors in coherency identification and may silently give inaccurate results. In this situation, the proposed error identification algorithm can significantly help to validate the solution of the reduced-order system.

**Table 2-3 The Groups of Coherent Generators with One Non-Coherent Group**

Group Number	Generators' bus numbers
1	{79, 80}
2	{93, 110}
3	{82, 109}
4	{101, 112}
5	{105, 106}
6	{91, 108, 121}
7	{67, 97, 124}
8	{104, 111}



**Figure 2-7 Rotor angle of generators #104 and #111 in the full-order system**



**Figure 2-8 Rotor angle of generator #60 in the presence of an error in coherency identification**

As the last study, the significance of accuracy of the reduced-order system as well as the proposed error identification algorithm are investigated. As mentioned in [91], the described three-phase fault at bus 7 imposes a stability limit on the maximum power generated by generators at buses #104 and #111. In this study, using the same groups presented in the previous studies, the plant generation limit at bus #104 is determined by changing the pre-fault generation at this bus. Using the methodology described in [91], the generation at bus #104 is increased in 10MW steps until the system becomes unstable. The results are presented in Table 2-4, wherein the first column represents the study number. In studies 1, 2, and 3, the groups of coherent generators presented in Table 2-1, Table 2-2, and Table 2-3 are used, respectively. According to Table 2-4, in the first study, wherein only the tightly coherent generators were used, the maximum plant generation predicted by all methods is the same. In the second study, while the NDA and DSATools give the same results, the IAA method has 10MW error in plant generation limit. In the third study, wherein an error in coherency identification exists, the maximum generation predicted by

the IAA and the NDA (without error identification) are much larger than DSATools. If an operator relies on these results to set the generation at bus #104, a serious risk will be imposed to the grid and the power system might become unstable if a fault happens at bus #7. However, the last column in Table 2-4 shows that using the proposed error identification algorithm it is possible to create an accurate reduced-order system, which results in the accurate amount for the plant generation limit.

**Table 2-4 Plant Generation Limit for Generator at bus #104**

Study Number	DSATools (Full-order system)	IAA	NDA without error identification	NDA with error identification
1	2200	2200	2200	2200
2	2200	2210	2200	2200
3	2200	2280	2270	2200

## **2.2 Aggregation in Transmission System**

### **2.2.1 Identifying coherent areas in transmission system**

Nature of coherency in transmission system is different from generation system. In order to investigate the coherency in transmission system, as the first step, the factors affecting bus voltages should be identified. The buses in the power system are the nodes of a large electrical circuit whose voltages are supported by generators (voltage sources) and loads (current sources). The effect of each source on each bus voltage is described by network nodal admittance (Y-bus) and impedance (Z-bus) matrices. Generally, being affected by same sources, voltages of adjacent buses tend to follow the same pattern whereas buses which are physically far from each other are usually affected by different sources and their voltages profile are different. This observation is the basic idea behind the coherency concept in the transmission system, where a coherent area consists of several buses whose voltages are affected by same sources to almost the same degree. Therefore, investigating the coherency in transmission system is equivalent to finding effect of each source on each bus voltage using network nodal admittance and impedance matrices.

Suppose generators are represented by the voltage behind reactance model [94]. This model can be used to represent both classical model (in the simplest form) and full-order model for the generators. In the latter case, the generators are equipped with the excitation system. The magnitude of VBR is constant in classical model and variable in the detailed model, and the proposed method is able to handle both cases. The bus voltages can be

described in terms of generator voltages and currents of the nonlinear loads (i.e. constant current and constant power loads) as

$$\mathbf{v} = \mathbf{N} \begin{bmatrix} \mathbf{E}'_q \\ \mathbf{i}_{cc} \\ \mathbf{i}_{cp} \end{bmatrix}, \quad (2-29)$$

$$\mathbf{N} = [\mathbf{M}, \mathbf{Z}, \mathbf{Z}],$$

where:  $\mathbf{E}'_q = [\mathbf{E}'_{q1}e^{j\delta_1}, \mathbf{E}'_{q2}e^{j\delta_2}, \dots, \mathbf{E}'_{qm}e^{j\delta_m}]^T$  represents vector of voltages behind transient reactances of generators;  $\mathbf{i}_{cc}$  and  $\mathbf{i}_{cp}$  represent vectors of injected currents due to constant current and constant power loads, respectively; vector  $\mathbf{v} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n]^T$  represents bus voltages;  $\mathbf{Z} \in \mathbf{R}^{n \times n}$  is the nodal impedance matrix of the transmission system which contains transmission lines, constant impedance loads, and transient reactances of generators;  $\mathbf{M} \in \mathbf{R}^{n \times m}$  is the matrix connecting internal nodes of generators to bus voltages;  $m$  is the number of generators; and  $n$  is number of buses in the system. It should be stressed that (2-29) does not assume a linear model in the network. The linear loads (i.e. constant impedance loads) are included in the admittance matrix which finally appear in  $\mathbf{N}$ . The nonlinear loads are supposed to be independent sources and included in the vector representing sources (i.e.  $[\mathbf{E}'_q, \mathbf{i}_{cc}, \mathbf{i}_{cp}]^T$ ).

Unless otherwise specified, in the rest of this section, the term “voltages” refers to bus voltages and “sources” refers to voltage and current sources, i.e.  $\mathbf{E}'_q$ ,  $\mathbf{i}_{cc}$ , and  $\mathbf{i}_{cp}$ , respectively.

Each row in  $\mathbf{N}$  describes effect of sources on each voltage. Dependency between two rows indicates that two voltages are affected by same sources to same degree, which implies coherency between buses. Symbolically, the dependency between rows  $i$  and  $j$  can be expresses as

$$\mathbf{v}_i = \alpha \mathbf{v}_j, \quad (2-30)$$

where  $\alpha$  is some complex number. If the difference between angles of  $\mathbf{v}_i$  and  $\mathbf{v}_j$  remains constant for all times, this implies that  $\mathbf{v}_i$  and  $\mathbf{v}_j$  are coherent. Therefore, if two rows in  $\mathbf{N}$  are dependent, the corresponding voltages will be coherent. Generally, a set of dependent rows in  $\mathbf{N}$  creates a coherent area in the transmission system. Accordingly, the objective of the proposed methodology is to find dependent rows in  $\mathbf{N}$ . An algorithm for finding linearly dependent rows in a matrix was proposed in [32]. However, this algorithm requires Gauss-elimination with full pivoting as well as solving a system of equations, which makes it very costly. Here, we propose a more efficient algorithm for finding dependent rows in  $\mathbf{N}$ .

Eq. (2-29) represents the linear relationship between sources and voltages including different loads, i.e. constant power, constant current, and constant impedance. Using this relationship, it is shown in the appendix that rows  $i$  and  $k$  are dependent (and consequently buses  $i$  and  $k$  are coherent) if the following inequality is satisfied:

$$\frac{\sum_{l=1}^n |N_{i,l}| |N_{k,l}| |\sin(\alpha_{i,l} - \alpha_{k,l})|}{\sum_{l=1}^n |N_{i,l}| |N_{k,l}| |\cos(\alpha_{i,l} - \alpha_{k,l})|} < \varepsilon_{eq}. \quad (2-31)$$

This equation describes coherency condition in terms of elements of matrix  $\mathbf{N}$ . As two rows  $i$  and  $k$  in this matrix get closer, the terms  $\sin(\alpha_{i,l} - \alpha_{k,l})$  (nominator) and  $\cos(\alpha_{i,l} - \alpha_{k,l})$  (denominator) in (2-31) become smaller and larger, respectively. Therefore, (2-31) tries to find those rows in  $\mathbf{N}$  which are close to each other, and accordingly, dependent.

### 2.2.2 Reducing redundancy

So far, we assumed that there is no coherency at generation level. In the presence of coherent generators, two buses can be coherent even if corresponding rows in the matrix  $\mathbf{N}$  are not dependent, which implies that the algorithm described in Section 2.2.1 will not be able to find all coherent buses when the coherency at generation level is neglected. Furthermore, coherent machines tend to show similar VBRs and keeping all of them creates redundancy in computations, which ultimately increases the computational cost. By eliminating coherent generators, it is possible to avoid the computational redundancy and improve accuracy and computational performance of the proposed method. To this end, we assume that results of coherency identification at generation level are available and it is shown how the groups of coherent generators can be removed from (2-29) to create a set of non-coherent generators.

By definition, the generators  $i$  and  $j$  are coherent if the ratio between their VBRs is approximately constant for all times, i.e.

$$\frac{E'_{qi}e^{j\delta_i}}{E'_{qj}e^{j\delta_j}} < c_0, \quad (2-32)$$

where  $c_0$  is a small complex number. In a group of coherent generators each pair of the generators keep a constant ratio between the voltages behind reactance. If a generator is not coherent with any other generator, it creates a group which consists of only one machine. Suppose there are  $g$  groups of coherent generators each denoted by  $J_k$ . Additionally, let  $\delta_k^{eq}$  represents rotor angle of the equivalent machine in group  $k$ , and  $\Delta\delta_l$  denotes angle difference between machine  $l$  and its equivalent machine, i.e.  $\delta_l = \delta_k^{eq} + \Delta\delta_l$ . It should be noted that if generators are tightly coherent,  $\delta_k^{eq}$  can be rotor angle of any generator in the group. However, in order to achieve more accurate results, it is more appropriate to use the NDA algorithm described in sections 2.1.3 and 2.1.4 to create the equivalent machine. This would be a safe choice as it tends to keep dynamics of the generators which are more influential.

Since each generator belongs to exactly one group of coherent machines, voltage of bus  $i$  can be written as

$$V_i e^{j\theta_i} = \sum_{k=1}^g \sum_{l \in J_k} N_{i,l} E'_{q,l} e^{j\delta_k^{eq}} e^{j\Delta\delta_l} + \mathbf{Z}(\mathbf{i}_{cc} + \mathbf{i}_{cp}) \quad (2-33)$$

where  $e^{j\delta_k^{eq}}$  is independent from  $l$  and can be taken out of first summation. This results in the following

$$V_i e^{j\theta_i} = \sum_{k=1}^g \tilde{M}_{i,k} e^{j\delta_k^{eq}} + \mathbf{Z}(\mathbf{i}_{cc} + \mathbf{i}_{cp}), \quad (2-34)$$

$$\tilde{M}_{i,k} = \sum_{l \in J_k} M_{i,l} E'_{q,l} e^{j\delta_l} + \mathbf{Z}(\mathbf{i}_{cc} + \mathbf{i}_{cp}),$$

which can be reformulated in the following matrix form

$$\mathbf{V} = \tilde{\mathbf{N}} \begin{bmatrix} \mathbf{E}'_{eq} \\ \mathbf{i}_{cc} \\ \mathbf{i}_{cp} \end{bmatrix}, \quad (2-35)$$

$$\tilde{\mathbf{N}} = [\tilde{\mathbf{M}}, \mathbf{Z}, \mathbf{Z}],$$

$$\mathbf{E}'_{eq} = [e^{j\delta_1^{eq}}, e^{j\delta_2^{eq}}, \dots, e^{j\delta_N^{eq}}].$$

Eq. (2-35) describes the relationship between voltages and equivalent generators. Comparing (2-35) with (2-29), it can be observed that  $\mathbf{E}'_q$  and  $\mathbf{M}$  are replaced by  $\mathbf{E}'_{eq}$  and  $\tilde{\mathbf{M}}$ , respectively. Size of  $\mathbf{E}'_q$  and  $\mathbf{E}'_{eq}$  (as well as number of columns in  $\mathbf{M}$  and  $\tilde{\mathbf{M}}$ ) are equal to number individual generators and equivalent generators, respectively. In the presence of coherency among generators, number of equivalent generators is always less than number of individual generators since the proposed NDA algorithm aggregates each group of coherent generators into one equivalent machine, which implies that size of  $\mathbf{E}'_{eq}$  and  $\tilde{\mathbf{M}}$  is always less than  $\mathbf{E}'_q$  and  $\mathbf{M}$ , respectively. Thus, the size of problem (2-29) can be reduced by replacing coherent generators with an equivalent machine.

### 2.2.3 Simulation results

The IEEE transient stability test system described in section 2.1.6 has been used here to evaluate effectiveness of the proposed method. In this section, different load models are used for different loads as shown in Table 2-5. In this table, first column represents bus number where the load is located and columns two to four represent the percentage of constant power, constant current and constant impedance of the load, respectively. For example in bus #112, the load is supposed to be 20% constant power, 50% constant current, and 30% constant impedance. Also, Powertech's DSATools is used to simulate the system using TDS and the results are used as a benchmark. A fault is applied at bus #59 and cleared by opening transmission line between buses #59 and #72 after 0.08sec. Coherent groups identified by the DSATools as well as the proposed method are summarized in Table 2-6. In this table, each bracket represents a group of coherent buses. In the second column, plain font denotes correct identification. The bold numbers show that the corresponding bus is not identified correctly. For example, in the 6th row, the TDS shows that buses #{54,55,61,62,86} are coherent. However, the proposed method has placed these buses into two groups #{54,55,61} and #{62,86}. As can be seen, buses #{62,86} have been separated from their group and they are denoted by bold font. Further results are summarized in Table 2-7, which shows that the maximum possible reduction is 63% as determined by the TDS results. The proposed method has achieved a 57% reduction, which is close to the TDS result.

In dynamic equivalencing, the accuracy of the aggregated system highly depends on the coherency identification algorithm, which should be able to identify very tightly coherent

buses as most of transient stability simulations tend to last 10 or more seconds to investigate multi swing stability phenomenon [19]. If the buses in a group are not coherent, the aggregated system will not be valid since the aggregation algorithm assumes that all buses in a group are tightly coherent. Therefore, although very important, the reduction percentage is not the only factor in evaluating performance of a coherency identification method. In fact, the method should manifest a conservative behavior and identify only very tightly coherent buses. To further evaluate the performance of the proposed algorithm, two types of errors are defined as follows:

1. Isolation Error: This term refers to a situation where a bus belonging to a group of coherent buses, is isolated from the rest of the group. For example, in Table 2-6, the proposed method has isolated bus #27 from its true group.
2. Misplacement Error: This type of error refers to a bus that is misplaced into a group to which it does not belong. For example, in Table 2-6, a misplacement error occurs if bus #3 is added to the first group, as this bus is not coherent with the rest of the group.

The isolation error only increases size of the aggregated system and does not threaten accuracy of the reduced system. However, the misplacement error places two non-coherent buses in the same group, and therefore, this error violates the underlying assumption in the aggregation algorithm. Thus, the reduced system and final results may not be valid when misplacement errors exist.

The results in Table 2-7 shows that the TDS and the proposed method have identified 60 and 55 buses which can be eliminated via aggregation, respectively, which implies that there are 5 isolation errors in the proposed coherency identification method. Moreover, by comparing the results presented in Table 2-6, it can be seen that the proposed method does not have any misplacement error. Presence of isolation errors and lack of misplacement errors show that the proposed method has a conservative nature. For the sake of illustration, the voltage angles and magnitudes of the second group of coherent buses identified by the proposed method are shown in Figure 2-9 and Figure 2-10. As can be seen, all buses in this group are tightly coherent even after 10 seconds.

The algorithms described in this chapter can be used to reduce size of original system and create an accurate and robust reduced-order model. The resulting system is then solved using the TDS approach to simulate system dynamics and predict status of the electrical grid following the disturbance. In the next two chapters, it is assumed that the original system can be reduced using the proposed methods, and we focus on expediting the TDS using parallel processing.

**Table 2-5 Load Models**

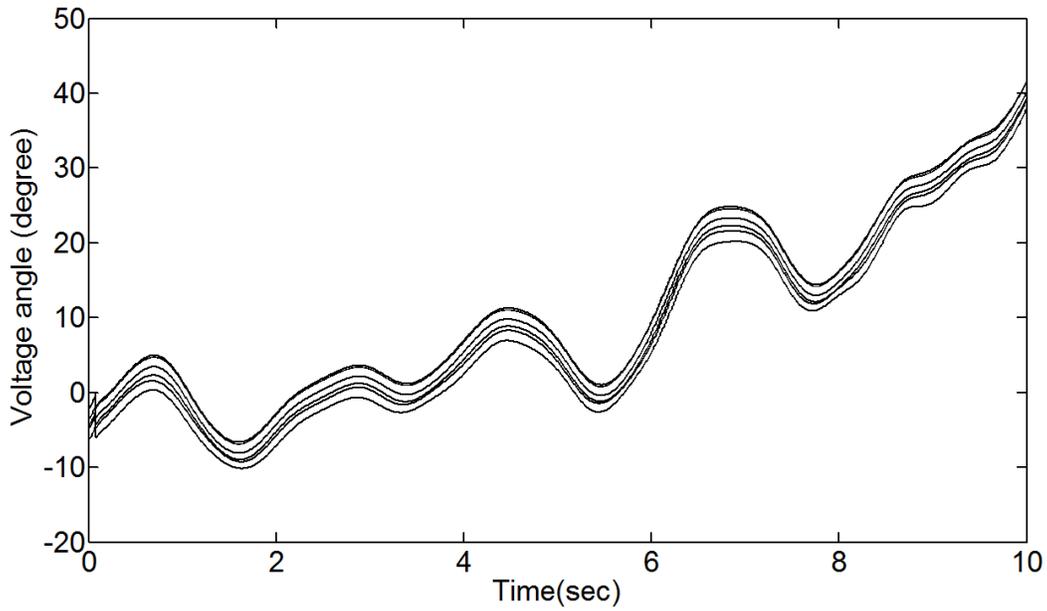
Bus Number	Percent of the load			Bus Number	Percent of the load		
	P,Q	I	Z		P,Q	I	Z
34	0	50	50	112	20	50	30
35	10	40	50	115	10	50	40
51	0	40	60	116	5	35	60
58	10	30	60	117	0	0	100
66	0	0	100	118	0	100	0
68	0	50	50	119	40	20	40
70	0	0	100	120	10	50	40
71	5	30	65	121	20	50	30
74	10	30	60	122	10	50	40
78	20	40	40	123	0	0	100
79	10	45	45	124	0	100	0
80	15	20	65	125	20	40	40
81	20	30	50	126	10	50	40
82	15	15	70	127	10	50	40
84	20	50	30	128	20	20	60
85	10	50	40	129	5	60	35
88	20	55	25	130	30	30	40
89	10	60	30	131	30	50	20
90	20	40	40	132	10	60	30
92	20	50	30	133	0	0	100
93	0	0	100	134	0	100	0
94	30	40	30	135	0	50	50
95	30	30	40	136	10	40	50
99	10	50	40	137	20	40	40
101	0	0	100	138	30	50	20
102	0	100	0	139	5	30	65
104	0	100	0	140	10	50	40
105	0	100	0	141	20	40	40
106	0	80	20	142	10	50	40
107	40	40	20	143	0	0	100
110	20	40	40	144	0	100	0
111	10	50	40	145	20	50	30

**Table 2-6 Coherent Buses Identified by TDS and Proposed Method**

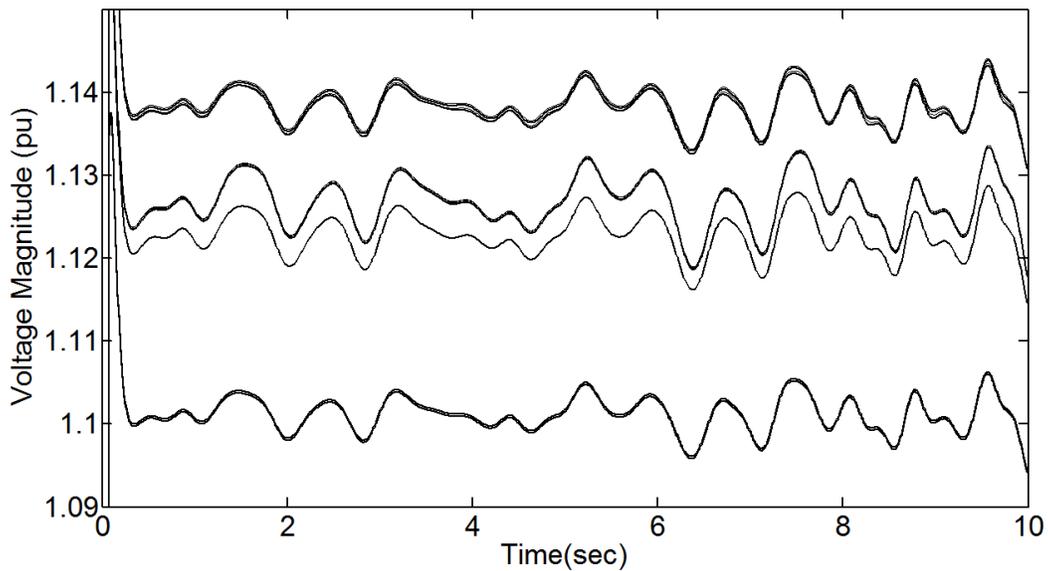
TDS	Proposed Method
{1,2,113,114}	{1, 2,113,114}
{3,4,5,6,9,10,12, 33,34,35,36,37,38,39,40,47,48,49,50,70,71,72}	{3,4,5,6, 9,10,12,33,34,35,36,37,38,39,40,47,48,49,50,72}{ <b>70</b> <b>,71</b> }
{11,32,69}	{11,32,69}
{41,42,43,44,51}	{41,42,43,44,51}
{45,46,52,53}	{45,46,52,53}
{54,55,61,62,86}	{54,55,61}{ <b>62,86</b> }
{84,88}	{84,88}
{76,77}	{76,77}
{27,28,29,75}	{28,29,75}{ <b>27</b> }
{18,19,20,21,59}	{18,19,20,21,59}
{22,30,78, 23,83}	{23,30,78,83}{ <b>22</b> }
{26,31,73,74,81}	{26,31,73,74,81}
{64,65}	{64,65}
{14,15,16,56,57,58}	{15,16,56,57,58}{ <b>14</b> }

**Table 2-7 Reduction Achieved After Aggregating Coherent Buses**

	TDS	Proposed Method
Total number of buses after aggregation	35	40
Size of the system after aggregation (% of base system)	37%	43%



**Figure 2-9 Voltage angle of second group consisting of buses #{3,4,5,6, 9,10,12,33,34,35,36,37,38,39,40,47,48,49,50,72}.**



**Figure 2-10 Voltage magnitudes of second group consisting of buses #{3,4,5,6, 9,10,12,33,34,35,36,37,38,39,40,47,48,49,50,72}.**

# CHAPTER 3: MULTI-DECOMPOSITION

## APPROACH

### 3.1 Conventional Time-Domain Simulation Approach

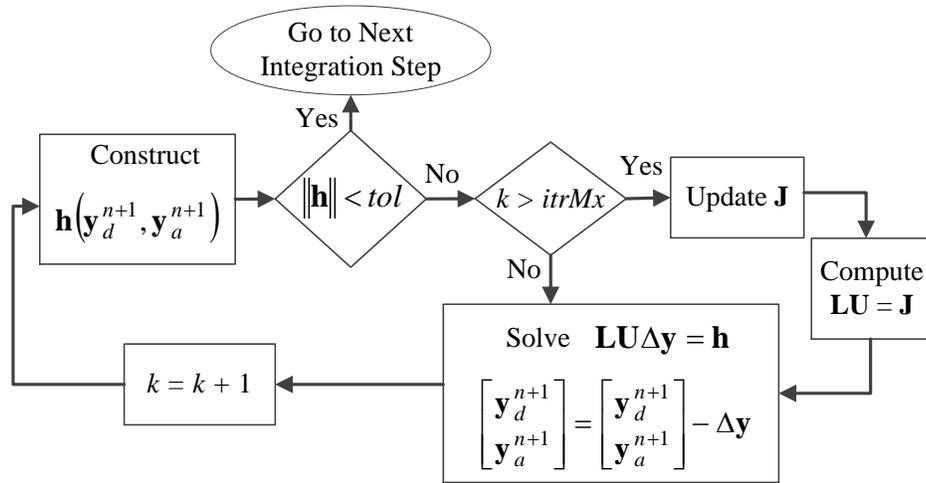
As discussed in section 1.2.1, the nonlinear DAEs in transient stability problem can be solved using simultaneous or alternative approach [17]. In this thesis, we focus on simultaneous approach for solving (2-1), wherein the dynamic and algebraic variables are solved simultaneously. In simultaneous approach, an implicit integration method such as backward Euler or trapezoidal is typically used to discretize differential equations, which results in a system of nonlinear equations. For example, discretizing (2-1) using trapezoidal rule yields the following nonlinear equations:

$$\mathbf{h} = \begin{bmatrix} \mathbf{y}_d^{n+1} - \mathbf{y}_d^n - \frac{T}{2} (\mathbf{f}_d(\mathbf{y}_d^{n+1}, \mathbf{y}_a^{n+1}) + \mathbf{f}_d(\mathbf{y}_d^n, \mathbf{y}_a^n)) \\ \mathbf{f}_a(\mathbf{y}_d^{n+1}, \mathbf{y}_a^{n+1}) \end{bmatrix} = \mathbf{0}, \quad (3-1)$$

where,  $\mathbf{h}$  is called mismatch vector;  $T$  and  $n$  represent integration time-step and number of current integration step, respectively. Eq. (3-1) should be solved iteratively to find  $\mathbf{y}_d^{n+1}$  and  $\mathbf{y}_a^{n+1}$ . The Newton-Raphson method [89] has been widely used to solve (3-1), which encompasses four steps: (a) Construct  $\mathbf{h}$  (b); Calculate partial derivatives of  $\mathbf{h}$  with respect to  $\mathbf{y}_d^{n+1}$  and  $\mathbf{y}_a^{n+1}$  to construct Jacobian matrix; (c) Decompose the Jacobian matrix using LU factorization [89]; and (d) Update the solution by solving the linearized system using

forward/backward substitution. This approach is referred to as Full Newton method since the Jacobian matrix is updated at every iteration.

The Full Newton method typically requires few iterations (1~3) to solve the system of nonlinear equations. However, performing LU factorization at every iteration is very time-consuming, which makes Full Newton inefficient for large-scale power systems. In order to reduce the amount of calculations, a Very DisHonest Newton (VDHN) method [95] was proposed, wherein the Jacobian matrix is updated only when the number of iterations exceeds a specific threshold (typically 4~5) [96]. While the number of iterations in VDHN is larger than Full Newton method, the LU factorizations are performed far less, and for this reason the VDHN method has been proven to be the fastest available sequential approach [24]. The block diagram of VDHN is shown in Figure 3-1. This figure depicts the steps required to solve a single integration step. In Figure 3-1,  $k$  and  $itrMx$  represent iteration number and maximum number of iterations before updating Jacobian matrix, respectively;  $\mathbf{J}$  represents the Jacobian matrix; and matrices  $\mathbf{L}$  and  $\mathbf{U}$  represent lower and upper triangular factors of  $\mathbf{J}$ , respectively. As Figure 3-1 suggests, the mismatch vector  $\mathbf{h}$  in each iteration depends on the solution from the previous iteration, which implies that VDHN is inherently sequential and cannot be expedited using parallel computing. Instead, parallelism is mainly exploited in solving the system of linear equations, which is known as parallelism-in-space [50]-[59]. More importantly, the local convergence property of the Full Newton does not exist in VDHN, and a complex mechanism is required to ensure that VDHN converges.



**Figure 3-1 Block diagram of the VDHN method for solving system of nonlinear equations at each integration step.**

### 3.2 Proposed Multi-Decomposition Approach

In this section, we propose a new integration technique for solving nonlinear DAEs (2-1) based on a Volterra-like series method [97]-[99]. In the proposed MDA, the original nonlinear system is decomposed into several linear subsystems using multivariable Taylor series. This procedure is schematically depicted in Figure 3-2. The linear subsystems are subsequently used to approximate the solution of the original nonlinear system (2-1). At each decomposition step, the MDA separates linear and nonlinear terms. The linear part participates in approximating the solution while the nonlinear part is further decomposed into a new linear and nonlinear subsystem. The sum of the solutions of linear subsystems approximates the solution of the original nonlinear system. In general, the accuracy of such approximation can be improved by increasing the number of decompositions. However, the computational cost of constructing such subsystems is also increasing. To better understand this approach, it is instructive to consider the multivariable Taylor series.

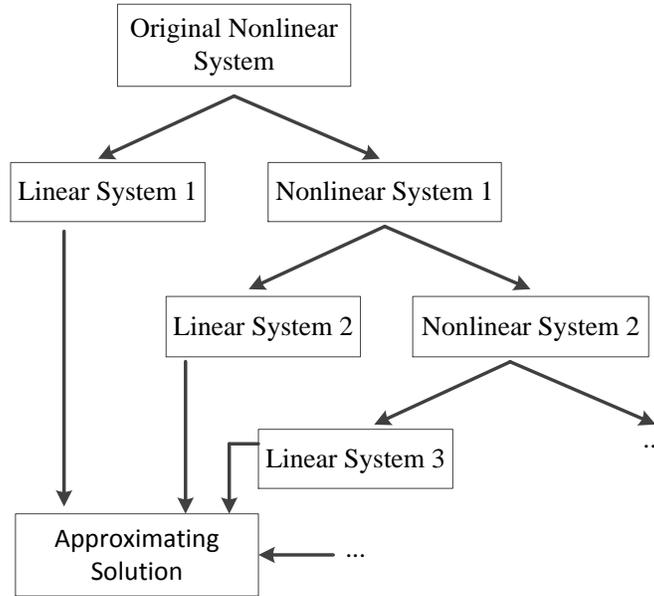


Figure 3-2 Decomposing nonlinear system into linear subsystems using MDA.

### 3.2.1 Taylor series of multivariable functions

A vector function  $\mathbf{f}(\mathbf{z}) = [f_1, f_2, \dots, f_N]^T : \mathbf{R}^N \rightarrow \mathbf{R}^N$  can be expanded about an operating point  $\mathbf{z}_0$  as

$$\mathbf{f}(\mathbf{z}) = \begin{bmatrix} f_1(\mathbf{z}_0) \\ \vdots \\ f_N(\mathbf{z}_0) \end{bmatrix} + \begin{bmatrix} \sum_{i=1}^N A_{1,i} x_i \\ \vdots \\ \sum_{i=1}^N A_{N,i} x_i \end{bmatrix} + \begin{bmatrix} \sum_{j=1}^N \sum_{i=1}^N H_{1,ij} x_i x_j \\ \vdots \\ \sum_{j=1}^N \sum_{i=1}^N H_{N,ij} x_i x_j \end{bmatrix} + \begin{bmatrix} \sum_{k=1}^N \sum_{j=1}^N \sum_{i=1}^N T_{1,ijk} x_i x_j x_k \\ \vdots \\ \sum_{k=1}^N \sum_{j=1}^N \sum_{i=1}^N T_{N,ijk} x_i x_j x_k \end{bmatrix} + \dots, \quad (3-2)$$

where,  $x_i = z_i - z_{i0}$ ; and  $A_{l,i} = \partial f_l / \partial x_i$ ,  $H_{l,ij} = \partial^2 f_l / (\partial x_i \partial x_j)$ , and  $T_{l,ijk} = \partial^3 f_l / (\partial x_i \partial x_j \partial x_k)$  represent the first-, the second-, and the third-order partial derivatives of  $f_l$ , respectively.

Each vector in (3-2) can be represented in matrix form as

$$\mathbf{f}(\mathbf{z}) = \begin{bmatrix} b_1 \\ \vdots \\ b_N \end{bmatrix} + \begin{bmatrix} A_{1,1} & \cdots & A_{1,N} \\ \vdots & \ddots & \vdots \\ A_{N,1} & \cdots & A_{N,N} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} + \begin{bmatrix} H_{1,11} & H_{1,12} & \cdots & H_{1,NN} \\ H_{2,11} & H_{2,12} & \cdots & H_{2,NN} \\ \vdots & \vdots & \ddots & \vdots \\ H_{N,11} & H_{N,12} & \cdots & H_{N,NN} \end{bmatrix} \begin{bmatrix} x_1 x_1 \\ x_1 x_2 \\ \vdots \\ x_N x_N \end{bmatrix} \quad (3-3)$$

$$+ \begin{bmatrix} T_{1,111} & T_{1,112} & \cdots & T_{1,NNN} \\ T_{2,111} & T_{2,112} & \cdots & T_{2,NNN} \\ \vdots & \vdots & \ddots & \vdots \\ T_{N,111} & T_{N,112} & \cdots & T_{N,NNN} \end{bmatrix} \begin{bmatrix} x_1 x_1 x_1 \\ x_1 x_1 x_2 \\ \vdots \\ x_N x_N x_N \end{bmatrix} + \cdots$$

where  $b_i = f_i(\mathbf{z}_0)$ . The representation can be simplified further using Kronecker product [100] notation as follows

$$\mathbf{f}(\mathbf{z}) = \mathbf{b} + \mathbf{A}\mathbf{x} + \mathbf{H}(\mathbf{x} \otimes \mathbf{x}) + \mathbf{T}(\mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x}) + \cdots, \quad (3-4)$$

$$\mathbf{x} = \mathbf{z} - \mathbf{z}_0,$$

where  $\otimes$  denotes the Kronecker product; and matrices  $\mathbf{A} \in \mathbf{R}^{N \times N}$ ,  $\mathbf{H} \in \mathbf{R}^{N \times N^2}$ , and  $\mathbf{T} \in \mathbf{R}^{N \times N^3}$  denote the first-, the second-, and the third-order partial derivatives of  $\mathbf{f}$ , respectively. In (3-4) vector  $\mathbf{b}$  is called mismatch vector, and matrices  $\mathbf{A}$  and  $\mathbf{H}$  are the Jacobian and Hessian matrices, respectively.

### 3.2.2 Multi-decomposition approach

Consider the following form of the original DAEs (2-1):

$$\mathbf{C}\dot{\mathbf{z}} = \mathbf{f}(\mathbf{z}), \quad (3-5)$$

where, vector  $\mathbf{z} = [\mathbf{y}_d^T, \mathbf{y}_a^T]^T \in \mathbf{R}^N$  represents both dynamic and algebraic variables, and

$\mathbf{C} \in \mathbf{R}^{N \times N}$  is used to separate differential and algebraic equations. This matrix has the following form:

$$\mathbf{C} = \begin{bmatrix} \mathbf{I}_{n \times n} & \mathbf{0}_{n \times m} \\ \mathbf{0}_{m \times n} & \mathbf{0}_{m \times m} \end{bmatrix}, \quad (3-6)$$

where,  $\mathbf{I}_{n \times n}$  and  $\mathbf{0}_{n \times m}$  represent  $n$  by  $n$  identity matrix and  $n$  by  $m$  zero matrix, respectively. Using Taylor series, (3-5) can be expanded around  $\mathbf{z}_0$  as

$$\mathbf{C}\dot{\mathbf{x}} = \mathbf{b} + \mathbf{A}\mathbf{x} + \mathbf{H}(\mathbf{x} \otimes \mathbf{x}) + \mathbf{T}(\mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x}) + \dots, \quad (3-7)$$

$$\mathbf{x} = \mathbf{z} - \mathbf{z}_0.$$

The first step in decomposing (3-5) can be performed by neglecting the nonlinear terms in (3-7) to derive the first linearized system as

$$L_1 : \mathbf{C}\dot{\mathbf{x}}_1 = \mathbf{b} + \mathbf{A}\mathbf{x}_1. \quad (3-8)$$

where,  $\mathbf{x}_1$  represents the solution to the first linear subsystem and is called the first-order term of the response. The linearized system approximates the original nonlinear system well only in the vicinity of  $\mathbf{z}_0$  and later deviates from the solution. Therefore, the linearized system along may only be valid for the first few steps. In order to improve the

approximation, it is required to consider nonlinear terms, which leads to the second step of decomposition.

Let  $\mathbf{x}_{1h}$  denotes the difference between the original solution  $\mathbf{x}$  and its first-order approximate  $\mathbf{x}_1$ , i.e.

$$\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_{1h}. \quad (3-9)$$

Then  $\mathbf{x}_{1h}$  will represent the effect of nonlinear terms in (3-7). The dynamics of  $\mathbf{x}_{1h}$  can be derived by substituting (3-8) and (3-9) into (3-7), as

$$\mathbf{C}\dot{\mathbf{x}}_{1h} = \mathbf{A}\mathbf{x}_{1h} + \mathbf{H}(\mathbf{x}_1 \otimes \mathbf{x}_1) + \mathbf{H}(\mathbf{x}_{1h} \otimes \mathbf{x}_1 + \mathbf{x}_1 \otimes \mathbf{x}_{1h} + \mathbf{x}_{1h} \otimes \mathbf{x}_{1h}) + \dots. \quad (3-10)$$

A linear approximation of  $\mathbf{x}_{1h}$  can be derived by neglecting the higher-order terms in (3-10) as

$$L_2 : \mathbf{C}\dot{\mathbf{x}}_2 = \mathbf{A}\mathbf{x}_2 + \mathbf{H}(\mathbf{x}_1 \otimes \mathbf{x}_1). \quad (3-11)$$

In (3-11),  $\mathbf{x}_2$  represents the linear approximation of  $\mathbf{x}_{1h}$ , and if it is added to  $\mathbf{x}_1$  it would improve the approximate solution.

Similarly, let  $\mathbf{x}_{2h}$  represent the difference between  $\mathbf{x}_{1h}$  and  $\mathbf{x}_2$ , i.e.  $\mathbf{x}_{1h} = \mathbf{x}_2 + \mathbf{x}_{2h}$ . Then, the solution of original nonlinear problem can be written in terms of solutions of the first two linear subsystems as

$$\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_{2h}. \quad (3-12)$$

If  $\mathbf{x}_{2h}$  is neglected in (3-12), it would provide a second-order approximation to the solution  $\mathbf{x}$ .

The approximation can be improved further by decomposing  $\mathbf{x}_{2h}$  and calculating higher order terms. For example, decomposing  $\mathbf{x}_{2h}$  and considering the third order term results in the following third linear subsystem

$$L_3 : \mathbf{C}\dot{\mathbf{x}}_3 = \mathbf{A}\mathbf{x}_3 + \mathbf{H}(\mathbf{x}_1 \otimes \mathbf{x}_2 + \mathbf{x}_2 \otimes \mathbf{x}_1) + \mathbf{T}(\mathbf{x}_1 \otimes \mathbf{x}_1 \otimes \mathbf{x}_1). \quad (3-13)$$

Ideally, if an infinite number of decomposition is performed,  $\mathbf{x}$  can be reconstructed as

$$\mathbf{x} = \sum_{k=1}^{\infty} \mathbf{x}_k. \quad (3-14)$$

This process is depicted in Figure 3-3, wherein  $n$  linear subsystems are used to approximate the solution. The input to each linear subsystem depends on the solutions of all previous linear subsystems and thus, the linear subsystems are solved recursively.

Since the MDA depends on partial derivatives of  $\mathbf{f}$ , construction of Hessian and higher order matrices should be computationally cheap. In transient stability problems, a considerable portion of computational cost is attributed to nonlinear power flow constraints, wherein the trigonometric functions are the main source of nonlinearity. In trigonometric functions, the second- and higher-order derivatives can be represented as a linear combination of the function itself and the first-order derivative. This characteristic makes the transient stability problem a convenient application for the MDA since most of the elements of Hessian, and higher order matrices are implicitly calculated while the

mismatches corresponding to the power flow constraints are evaluated. Therefore, construction of Hessian and higher order matrices requires small additional computational cost.

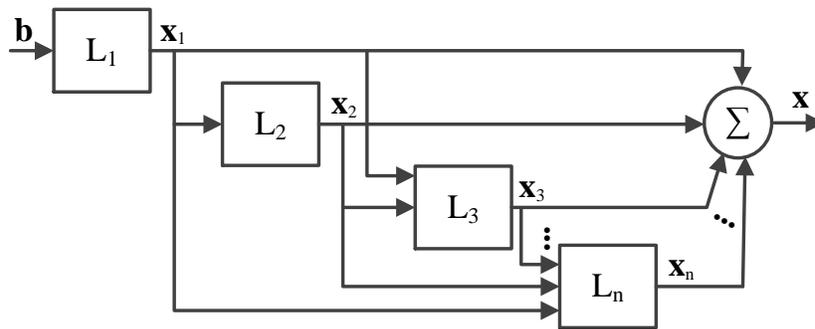


Figure 3-3 Computing system response  $x(t)$  using interconnected linear subsystems according to the proposed MDA.

### 3.3 Applying MDA to Transient Stability Problem

#### 3.3.1 Application of MDA to single machine system

To demonstrate application of the MDA, a single machine infinite bus example system is considered first. This system is described by the following equations:

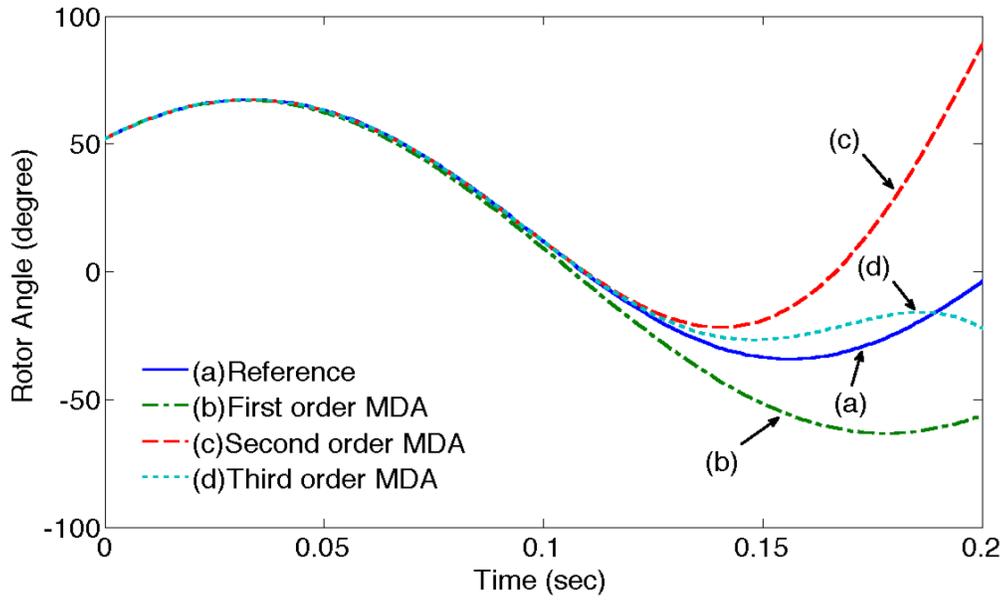
$$\begin{aligned} \dot{\delta} &= \omega, \\ \dot{\omega} &= \frac{1}{M} \frac{P_m - P_e \sin(\delta)}{X}, \end{aligned} \tag{3-15}$$

where,  $\delta$  and  $\omega$  represent the rotor angle and speed of the machine, respectively;  $M$  is the inertia; and  $X$  represent an equivalent reactance between the machine and the infinite bus, respectively;  $P_m$  and  $P_e$  denote the input mechanical power and the maximum output

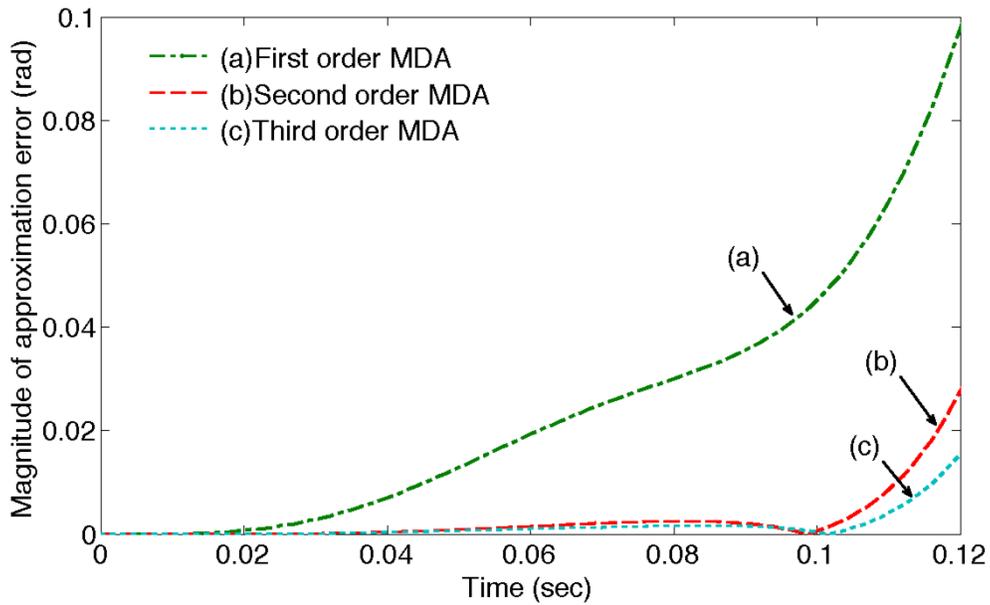
electrical power, respectively. The following parameters and initial conditions are assumed:  $P_m/(MX)=188.5$ ,  $P_e/(MX)=754$ ,  $\delta_0 = 50^\circ$ , and  $\omega_0 = 15$ . The resulting transient of the rotor angle is shown in Figure 3-4. The reference solution is obtained using a traditional TDS and conventional numerical integration of (3-15). Additionally, several approximate solutions have been calculated using the MDA, wherein the  $n$ th order approximation is defined as follows

$$\mathbf{x} \approx \mathbf{x}_1 + \mathbf{x}_2 + \dots + \mathbf{x}_n. \quad (3-16)$$

In order to better illustrate the difference between the orders of approximation, absolute value of approximation error is shown in Figure 3-5. This figure shows that the first-order approximation starts to noticeably deviate from the reference solution after  $t = 0.02$  s. The second- and third-order approximations also start to significantly deviate after  $t = 0.1$  s and  $t = 0.12$  s, respectively. In this way, one can see that the accuracy of approximation can be improved by increasing its order.



**Figure 3-4 Transient in the rotor angle as predicted by various orders of MDA for the single-machine example system (3-15).**



**Figure 3-5 Approximation error in the rotor angle resulting from using MDA.**

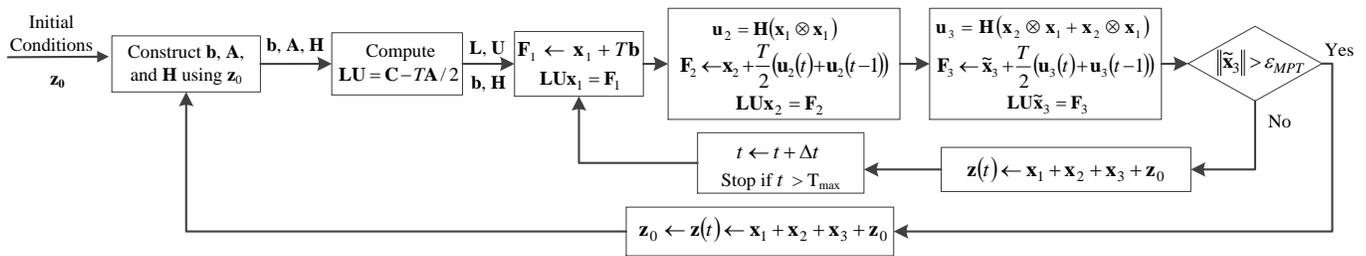
### 3.3.2 Adaptive updating scheme for the MDA

To make the MDA practical for large-scale systems, one can choose to use the second-order approximation and continuously update the mismatch vector  $\mathbf{b}$ , Jacobian matrix  $\mathbf{A}$ , and Hessian matrix  $\mathbf{H}$  (which are also called linear subsystems' parameters) using the last calculated state. Every such update requires a new construction of the Jacobian and Hessian matrices. More importantly, each new Jacobian matrix should be decomposed using LU factorization (which is used later for integrating the linear subsystems). Overall, updating the matrices is computationally expensive. In order to minimize the number of updates, the matrices should be updated only when the considered approximation starts to deviate from the reference solution and exceeds a certain error tolerance. The time when an update is required herein is called maximum prediction time (MPT). For example, for the study presented in Figure 3-4, the MPT may be approximately at  $t = 0.1$  s.

The MPT can also be defined as the maximum time for which the effect of higher order terms remains negligible. For example, for the considered second-order approximation, the update should be performed as soon as  $\|\mathbf{x}_3\|$  (the contribution from the next order of approximation) starts to grow above an acceptable error tolerance,  $\varepsilon_{MPT}$ . So, the system (3-13) can be used as an error function. However, using (3-13) requires third-order partial derivatives. While construction of  $\mathbf{T}$  may be computationally cheap, a matrix-vector multiplication including  $\mathbf{T}$  can be costly. Since (3-13) is used only to approximate  $\|\mathbf{x}_3\|$ , accurate computation of  $\mathbf{x}_3$  is not required and therefore, the third-order partial derivatives are neglected to approximate  $\mathbf{x}_3$  from the following

$$\mathbf{C}\tilde{\mathbf{x}}_3 = \mathbf{A}\tilde{\mathbf{x}}_3 + \mathbf{H}(\mathbf{x}_1 \otimes \mathbf{x}_2 + \mathbf{x}_2 \otimes \mathbf{x}_1). \quad (3-17)$$

A block diagram representation of applying the MDA to transient stability problem is shown in Figure 3-6. The trapezoidal rule is depicted here to solve the linear subsystems. As the first step,  $\mathbf{f}(\mathbf{z})$  is expanded around initial conditions to compute mismatch vector, Jacobian matrix, and Hessian matrix. The LU factors of Jacobian matrix are calculated next.



**Figure 3-6 Block diagram of the MDA assuming trapezoidal integration rule.**

Then, the algorithm enters the inner-loop, wherein the linear systems are solved. Three main steps in direct path calculate  $\mathbf{x}_1$ ,  $\mathbf{x}_2$ , and  $\tilde{\mathbf{x}}_3$ . If  $\|\tilde{\mathbf{x}}_3\|$  exceeds the threshold  $\varepsilon_{MPT}$ , it is concluded that it is necessary to update the linear subsystems. In this condition, the algorithm goes to the outer-loop and updates the matrices using the last calculated state. Otherwise, the algorithm remains in the inner-loop and continues solving the linear subsystems.

As it is depicted in Figure 3-6, the MDA checks validity of the approximation at each integration step and the linear subsystems are updated whenever it is necessary. Therefore, while avoiding unnecessary updates, it is ensured that the approximation remains accurate within the specified tolerance, regardless of the disturbance. Another

important aspect of the MDA is that no iterative solver is used in this method. As shown in Figure 3-6, the MDA solves three systems of linear equations in each integration time step, which does not require any iterations. Accordingly, the MDA does not experience convergence problems associated with iterative solvers.

At the same time, since the MDA performs three forward/backward substitutions in each integration step, it might appear that the MDA is not efficient for practical large-scale systems. In order to clarify this point, it is instructive to compare computational stages in VDHN and MDA methods, respectively. These stages can be described using block diagrams shown in Figure 3-1 and Figure 3-6. In the VDHN method, the LU factors of the Jacobian matrix are computed after fault is cleared and they are updated once the convergence is about to slow down (which is measured in terms of number of iterations required to find the solution). In the VDHN method, the solution in each integration step is found iteratively by solving a system of nonlinear equations, wherein several iterations are typically required to find the solution. Moreover, each iteration requires one evaluation of derivative function and a forward/backward substitution. However, in the proposed MDA, the LU factors of the Jacobian matrix are computed when fault is cleared and the factors are updated once the algorithm reaches to the MPT to satisfy the error tolerance  $\varepsilon_{MPT}$ . In addition, three forward/backward substitutions are performed in each integration step to solve three linear subsystems and find the solution for the specific integration step. Overall, although the MDA performs three forward/backward substitutions in each step, the number of forward/backward substitutions in the VDHN is variable (due to its iterative nature), and can be (and typically is) more than three.

### 3.4 Parallel Multi-Decomposition Approach

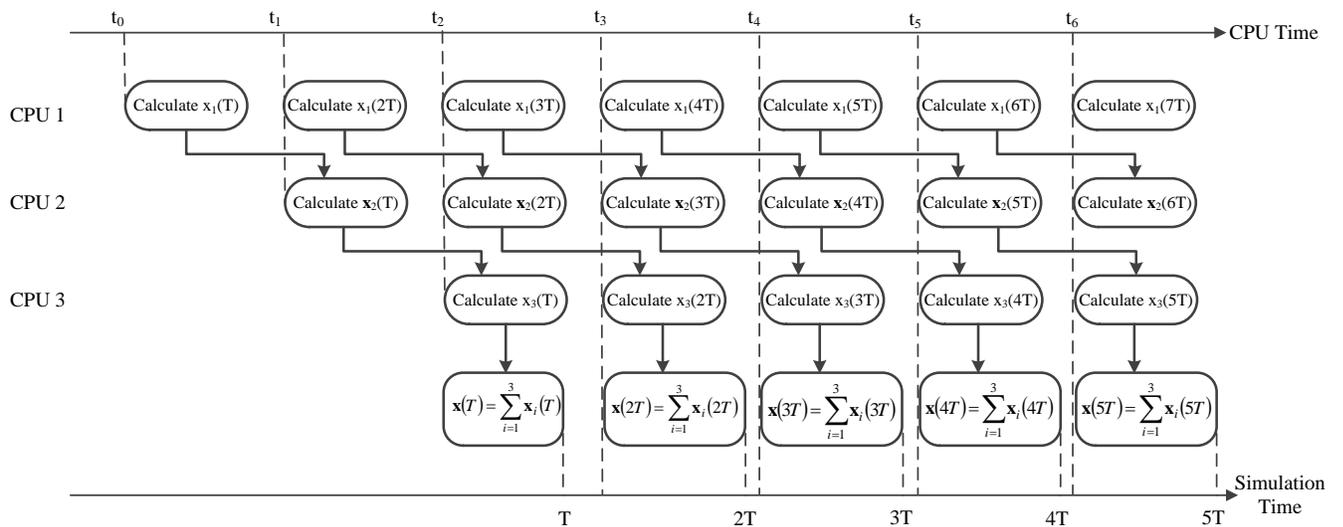
The block diagram of the MDA shown in Figure 3-6 has two major loops: the outer-loop, in which mismatch vector, Jacobian matrix, and Hessian matrix are constructed and LU factorization of  $\mathbf{C} - T\mathbf{A}/2$  is computed; and the inner-loop, in which three linear subsystems are integrated through forward/backward substitution. Similar to the VDHN, the LU factorization is performed not very often and most of the CPU time is spent into the inner-loop. While the LU factorization has been highly optimized and there are several sparse packages available [101]-[103] to perform parallel LU factorization, the parallel forward/backward substitution is still un-scalable due to high communication to computation ratio [104].

In the MDA, according to (3-8), (3-11), and (3-13), there is a unidirectional dependency between the blocks in the inner-loop, i.e. the first linear subsystem is independent from other linear subsystems, and the second linear subsystem is independent from the third one. In this framework, solving the linear subsystems can be shared among different processors as shown in Figure 3-7. As the program enters the inner-loop, each processor waits until a new solution is generated by the previous linear subsystems. For example, the second linear subsystem waits until  $\mathbf{x}_1(T)$  is calculated by the first linear subsystem and then it starts to use this solution to compute  $\mathbf{x}_2(T)$ . The main advantage of the MDA is that while the second linear subsystem is computing  $\mathbf{x}_2(T)$ , the first linear subsystem can concurrently work on computing  $\mathbf{x}_1(2T)$ , which means two forward/backward substitutions are performed simultaneously. Similarly, after two initial steps, three

processors will be busy, which shows that the execution of the inner loop can roughly become three times faster using three processors. The last processor monitors  $\|\tilde{\mathbf{x}}_3\|$  to detect if it exceeds  $\varepsilon_{MPT}$ . If this is the case, a signal is sent to all processors to stop working and the program goes into the outer-loop to update the parameters of the linear subsystems.

The basic parallel structure of the MDA implementation is depicted in Figure 3-7. The inner loop and parallel threads are also depicted in more detail in Figure 3-8. In our implementation, the Pthreads library [105] was used to create and synchronize three parallel threads, each corresponding to one of the linear subsystems. When a thread computes a new solution, the results are stored in a buffer, and a notification signal is sent to the next thread (corresponding to the next linear subsystem). The next thread waits until the notification signal is received, and then uses the solution of the previous linear subsystem (stored in the buffer) to start working on the next integration step. In this implementation, the last thread continuously monitors the norm of solution of third linear subsystem, and if it exceeds the specified threshold  $\varepsilon_{MPT}$ , a stop signal is sent to other threads. After that, the algorithm goes into the outer loop (shown in Figure 3-6). In order to reduce the overhead associated with creation of parallelization paths, the threads are not terminated when the algorithm leaves the inner loop. The threads are kept active and re-used when the algorithm comes back into the inner loop, wherein a signal is sent to the threads to start the calculations again. Using this approach, the threads are created only once and at the beginning of the simulation.

The proposed MDA is demonstrated to use three linear subsystems, which makes this approach easy to implement on three cores. Increasing the number of linear subsystems (cores) requires higher order partial derivatives of the derivative function, which will be computationally expensive to construct and evaluate. However, the main advantage offered by the MDA is that other parallelizable methods proposed in the literature can also be used inside the MDA to employ more processors. The MDA decomposes original nonlinear DAEs into three linear DAEs, which opens the possibility to use waveform relaxation techniques [106] and parallel-in-time methods [47]-[49]. Moreover, since a system of linear equations is solved in each integration step, available parallel-in-space techniques such as SOR [56], and network partitioning techniques [57] can be used as well.



**Figure 3-7 Block diagram of sharing computational tasks among three processors in the inner-loop of MDA.**

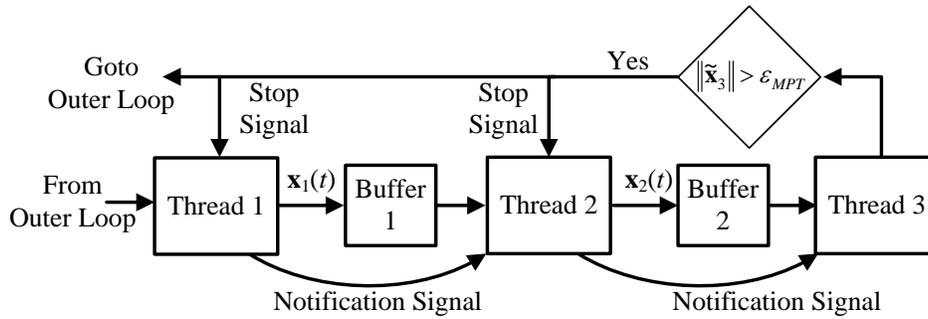


Figure 3-8 Implementation of the inner loop in parallel MDA.

### 3.5 Benchmark System and Simulation Results

The test system described in section 2.1.6 is used here to evaluate the effectiveness of the MDA. This system has 124 dynamic and 302 algebraic variables. The DSATools is used to generate the reference TDS solution and the results are referred to as DSATools. For comparison purposes, the Full-Newton and VDHN methods are coded in standard C. The Full-Newton method updates the Jacobian matrix at each iteration while for the VDHN, the Jacobian matrix is updated only when the number of iterations in the Newton-Raphson method used inside exceeds 4. Moreover, the error tolerance in both methods is set to  $10^{-4}$ . Both sequential and parallel versions of the MDA have been implemented using standard C, wherein the update tolerance is  $\epsilon_{MPT} = 10^{-4}$ . These implementations are referred to as the MDA and MDA\_P, respectively. Also, Math Kernel Library from Intel Corporation [101] is used to perform LU factorizations and forward/backward for solving linear system of equations. This package takes advantage of sparsity of matrices to improve computational performance.

Without loss of generality, the implicit trapezoidal integration rule with the fixed time-step of  $\Delta t = 1ms$  is used for all methods. All simulations are performed using a desktop computer with quad-core Intel *i7 2600@3.4GHz* CPU and 8GB RAM.

### 3.5.1 Case study

A three-phase fault at bus #90 is imposed at time  $t = 1s$ . The fault is cleared in 8 cycles by opening transmission line between buses #90 and #94. The post-fault system is simulated for 8 seconds using DSATools, VDHN, and MDA. To compare the results, voltage magnitude of bus #90 and rotor angle of the generator located at this bus are shown in Figure 3-9. Initially, the voltage and rotor angle are in steady state for the first second. As soon as the fault happens, the voltage at bus #90 drops to zero and the generator at this bus starts to accelerate. After the fault is cleared, both variables start to oscillate and the oscillations persist until the end of the simulation indicating that the system has very small damping. All the results shown in Figure 3-9 are visibly indistinguishable. In order to show the very small difference, a fragment of the rotor angle plot has been magnified, where a slight difference between DSATools, VDHN, and MDA can be observed.

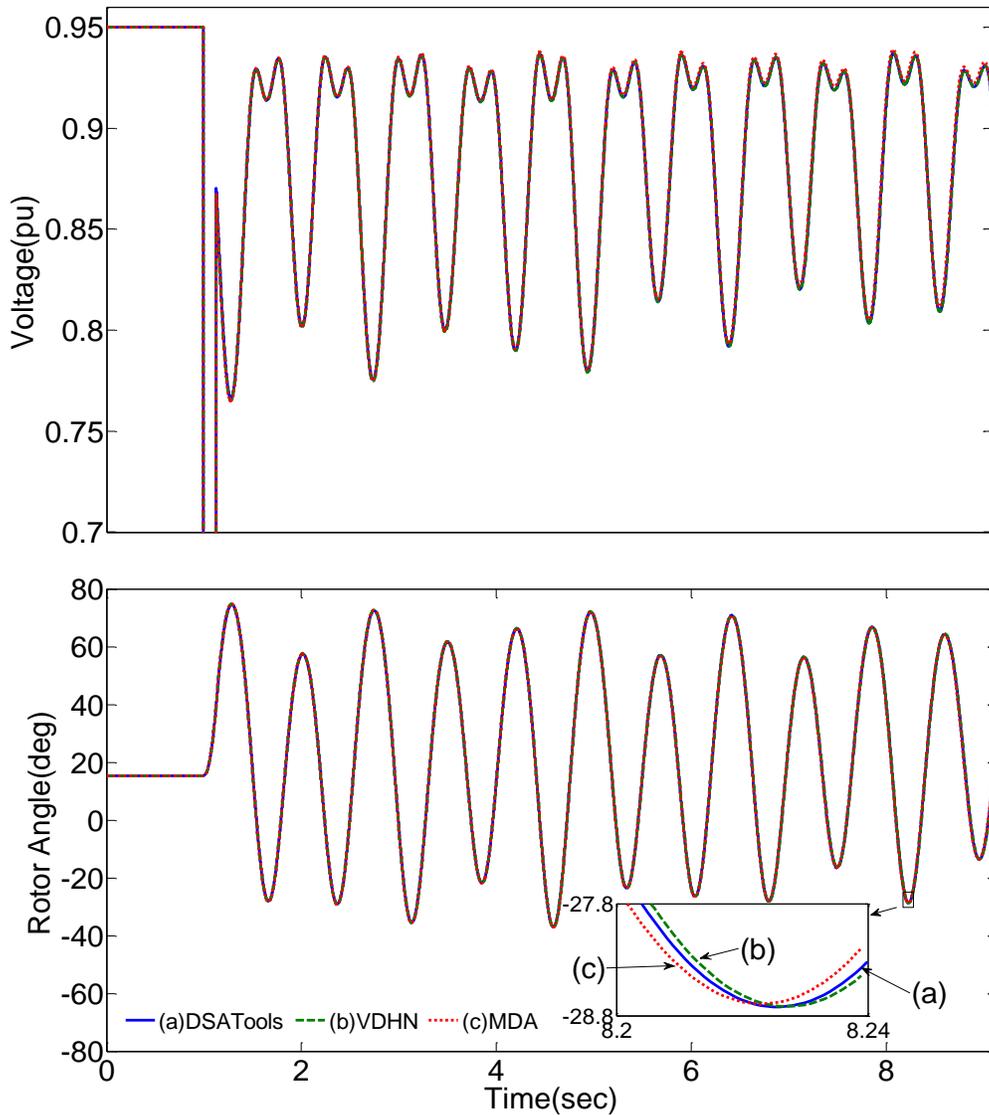


Figure 3-9 Bus voltage and rotor angle of the generator at bus #90.

### 3.5.2 Calculation of critical clearing time

In order to further validate the accuracy of the MDA, the critical clearing time (CCT) has been calculated for a number of scenarios. In each scenario, a three-phase fault is applied at the faulted bus and cleared by opening a transmission line between the faulted bus and one of its adjacent buses. The scenarios presented here are among those which lead to the least CCT in the test system. The CCTs are calculated iteratively and by increasing fault clearance

time. Initially, the fault clearance time is selected so that the post-fault system is stable. Then, the fault clearance time is increased in 1ms steps and until the system becomes unstable. Thus, the resolution of the CCT is 1ms in this study. The results calculated by the DSATools and MDA are presented in Table 3-1. In this table, the second and third columns represent the faulted bus and the tripped transmission line, respectively. Also, the fourth and fifth columns represent the CCTs calculated by the DSATools and the MDA, respectively. According to this table, the MDA and the DSATools result in the same CCT in all scenarios, which demonstrates that the MDA is as accurate as the DSATools considering the same tolerance.

**Table 3-1 Comparison of CCT Calculated by DSATools and MDA.**

Scenario Number	Faulted Bus	Disconnected Line	DSATools (ms)	MDA (ms)
1	142	142-145	63	63
2	142	142-120	63	63
3	121	121-125	64	64
4	131	131-141	66	66
5	141	141-132	85	85
6	122	122-132	195	195
7	7	7-6	152	152
8	67	67-125	174	174
9	90	90-94	178	178
10	60	60-135	194	194
11	96	96-108	208	208
12	80	80-90	212	212
13	109	109-121	264	264
14	120	120-125	238	238

### **3.5.3 Evaluate performance of the MDA during the fault**

A known problem associated with the Newton-type methods is that they might fail to converge, particularly during the fault when the voltages become very low. To investigate this issue, a fault is applied at bus #73 at  $t = 0.1s$  and it is cleared after 12 cycles (0.2s) by opening the transmission line between buses #73 and #69. In order to illustrate the effect of fault on bus voltage in different locations of the network, the voltages of five buses

located in different proximity to the fault are shown in Figure 3-10. When the system is in a pre-fault steady state condition, all bus voltages are within the range of 0.95 to 1.05 pu, as expected. After the fault, the voltages at buses #105, #96, and #27 drop significantly, which shows that these buses are close to the fault location. In contrast, the voltages at buses #113 and #120 are less affected by the fault as these buses are further away from the fault.

The fault scenario of Figure 3-10 has been implemented using several methods. When the system is simulated using Newton method with constant step size, it was observed that Newton method fails to converge at  $t = 0.244s$ . In order to show that the solution exists and Newton method cannot find it, a *damped Newton method* [107], [108] has been implemented and used to find the solution. In this approach, the step-size is adjusted adaptively based on the norm of the mismatch. If norm of the mismatch vector increases in two consecutive iterations, it means that the Newton method is about to diverge and the algorithm reduces step-size. While the damped Newton method is able achieve convergence (unlike the conventional Newton method), the convergence rate is degraded since smaller steps are selected. Using this methodology, the during-fault system was successfully simulated. In addition to the two Newton-type methods, the system was also simulated using the proposed MDA without any modifications. Since the MDA works with linear subsystems, not convergence problems were observed. For example, the transient observed in the rotor angle of generator at bus #101 is shown in Figure 3-11, as calculated by different methods. As Figure 3-11 shows, the Newton method with constant step-size fails to converge at  $t = 0.244s$  while the damped Newton method and the MDA yield the same results.

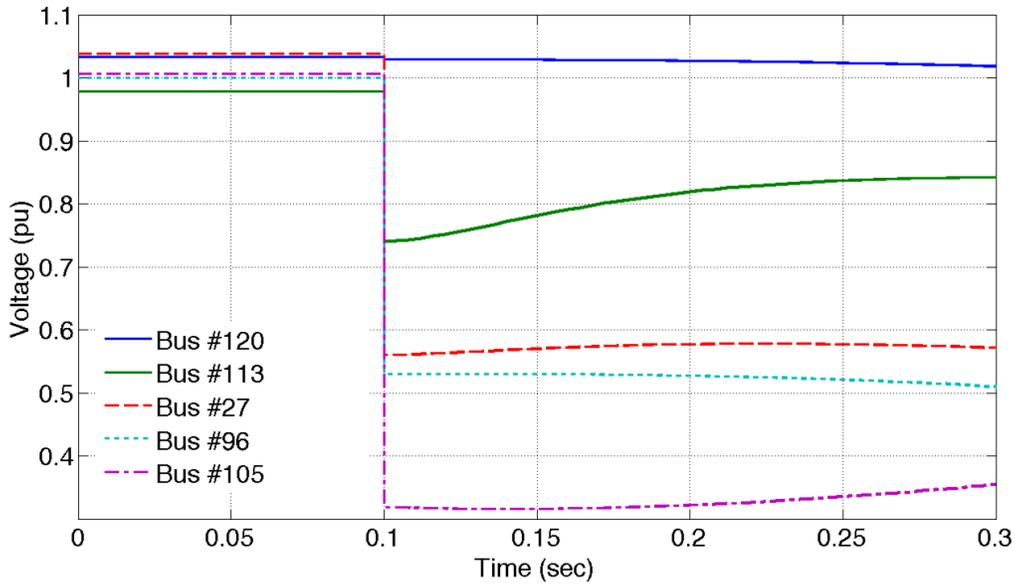


Figure 3-10 Voltages of five buses when fault is applied at bus #73.

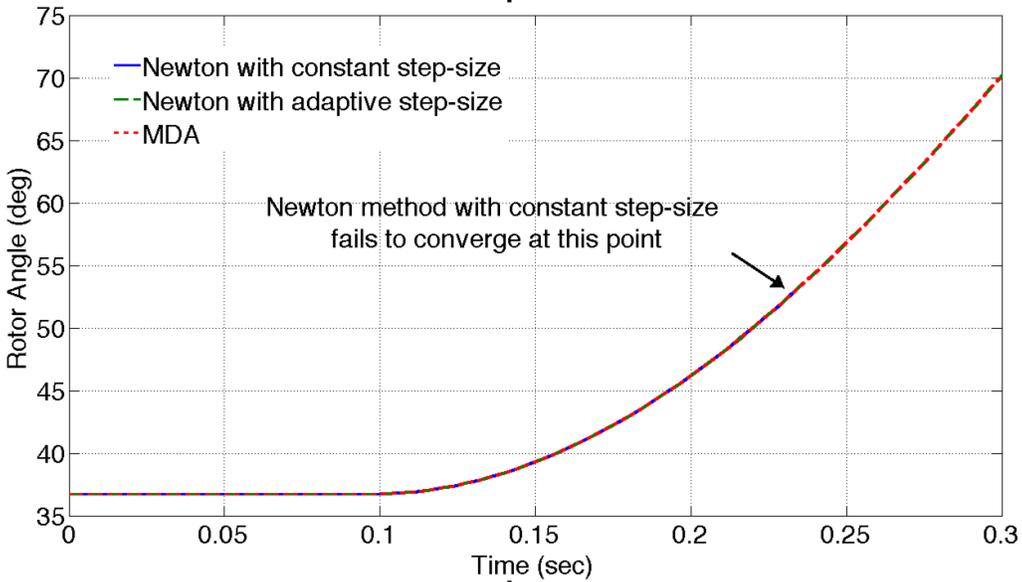


Figure 3-11 Rotor angle of generator at bus #101 simulated by three methods when fault is applied at bus #73.

### 3.5.4 Computational performance

In this section, the speed of MDA is compared against the conventional TDS methods. The execution time of different methods for the considered scenarios is presented in Table 3-2.

The first column in Table 3-2 represents the scenario number, according to Table 3-1, and the rest of columns represent the execution time of each method. In order to summarize the results, the total time required to solve all scenarios as well as the average time required to solve a scenario is shown in the last two rows, where it can be seen that both MDA and MDA\_P outperform Full Newton and VDHN. The last two rows show that MDA is faster than Full Newton and VDHN by 6.75 and 1.09 times, respectively. Using parallel implementation of the MDA, better computational performance can be achieved. Specifically, Table 3-2 shows that MDA\_P is faster than Full Newton and VDHN by 12.9 and 2.08 times, respectively.

As one of the fastest commercial packages for transient stability studies, computational performance of the DSATools is also compared against the MDA and MDA\_P. The *TSAT Scheduler* program was used to run the scenarios and measure execution time of the DSATools. Since the test system was not large enough, it was not possible to measure the execution time of the DSATools precisely for each individual scenario. Instead, the total time required to solve all scenarios is compared. To do so, a single case study with all 14 scenarios was created in the DSATools, and the total time required for solving the case study was recorded. The same procedure was carried out for MDA and MDA\_P and the results are summarized in Table 3-3. As can be seen in this table, the MDA and MDA\_P are faster than DSATools by 1.64 and 3.15 times, respectively.

**Table 3-2 Execution Time for Different Scenarios.**

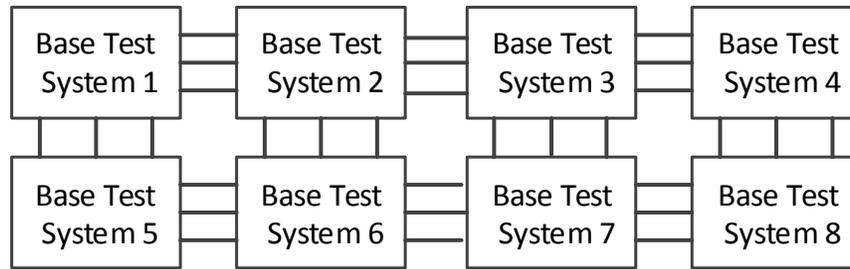
Scenario Number	Full Newton (sec)	VDHN (sec)	MDA (sec)	MDA_P (sec)
1	5.162	0.719	0.641	0.304
2	5.503	0.721	0.643	0.305
3	3.174	0.706	0.570	0.244
4	5.194	0.715	0.570	0.247
5	5.455	0.715	0.613	0.276
6	5.407	0.784	0.803	0.503
7	4.602	0.753	0.746	0.446
8	5.777	0.780	0.746	0.431
9	2.890	0.630	0.580	0.259
10	4.319	0.715	0.683	0.366
11	3.422	0.635	0.619	0.405
12	3.568	0.710	0.645	0.330
13	2.896	0.714	0.648	0.318
14	4.702	0.735	0.694	0.380
<b>Total Time</b>	<b>62.071</b>	<b>10.032</b>	<b>9.201</b>	<b>4.814</b>
<b>Average Time</b>	<b>4.438</b>	<b>0.717</b>	<b>0.657</b>	<b>0.344</b>

**Table 3-3 Total Execution Time for Fourteen Scenarios.**

	DSATools (sec)	MDA (sec)	MDA_P (sec)
Execution Time	15.1	9.2	4.8

### 3.5.5 Large-scale system

In order to explore the efficiency of the proposed MDA in larger-scale systems, the previous test system has been duplicated and interconnected 8 times to create a larger test system as depicted in Figure 3-12. Here, the “Base Test System  $i$ ” refers to the  $i$ th copy of IEEE 50 generators and 145 buses test system discussed in 3.5.1. In Figure 3-12, the adjacent systems are connected using six transmission lines. Thus, the new test system has 400 generators and 1160 buses, which amounts to 992 dynamic variables and 2416 algebraic variables, respectively. The same scenarios as described in Section 3.5.2, were used here. In each scenario, the fault is applied at the corresponding bus in “Base Test System 1”. The computational performances of different methods are summarized in Table 3-4. As it can be seen in this table, on average, the VDHN, MDA, and MDA\_P solve each scenario in 8.59, 7.59, and 3.84 seconds, respectively. These results show that MDA and MDA\_P are faster than VDHN by 1.13 and 2.24 times, respectively. These results are consistent with the performance presented for small-scale system in Section VI-D, wherein the MDA and MDA\_P were faster than VDHN by 1.09 and 2.08 times, respectively. As it has been demonstrated, the MDA is consistently faster than VDHN. Moreover, the performance improvement achieved using parallel MDA becomes more significant for the large-scale system. The additional improvement in parallel MDA is mainly attributed to the reduction in the overhead of creating and synchronizing the threads relative to the system size.



**Figure 3-12 Duplicating base test system to create a large-scale system.**

**Table 3-4 Execution Time for Different Scenarios in Large-Scale System**

Scenario Number	VDHN (sec)	MDA (sec)	MDA_P (sec)
1	8.62	7.288	3.59
2	7.51	6.96	3.57
3	7.804	6.568	2.95
4	7.715	6.618	2.95
5	9.813	8.8	2.74
6	9.022	8.951	5.68
7	8.901	8.11	4.98
8	9.316	7.524	4.56
9	8.959	8.017	3.36
10	9.421	7.245	4.18
11	8.241	7.689	3.49
12	8.002	7.442	3.88
13	8.276	7.756	3.66
14	8.62	7.288	4.19
<b>Total Time</b>	<b>120.22</b>	<b>106.26</b>	<b>53.78</b>
<b>Average Time</b>	<b>8.59</b>	<b>7.59</b>	<b>3.84</b>

# CHAPTER 4: SUCCESSIVE LINEARIZATION AND INTEGRATION TECHNIQUE

In chapter 3, an innovative multi-decomposition approach (MDA) was proposed, which is based on constructing three linear subsystems of the original system of nonlinear DAEs. These subsystems may then be solved sequentially or in parallel. While it was demonstrated in section 3.5 that the proposed MDA outperforms other state-of-the-art TDS methods, this method has its challenges as it requires second-order partial derivatives and the level of parallelization is limited by number of linear subsystems. In order to eliminate these limitations, this chapter proposes Successive Linearization and Integration Technique (SLIT) method, which is an innovative approach for integrating nonlinear DAEs through successive linearization of nonlinear terms. In the SLIT, there is no theoretical limit on number of linearizations performed, and the accuracy of approximated solution can be enhanced by increasing number of linear systems. Alternatively, one might choose to limit number of linear systems and update their parameters when an increasing error is detected, which can lead to better computational performance.

## 4.1 Mathematical Formulation

Assume that  $\mathbf{z}_0$  represents the initial conditions of the problem and  $\Delta\mathbf{z}$  denotes the difference between  $\mathbf{z}$  and  $\mathbf{z}_0$ , i.e.  $\mathbf{z} = \mathbf{z}_0 + \Delta\mathbf{z}$ . The goal is to approximate  $\Delta\mathbf{z}$  and use the results to find an estimation of  $\mathbf{z}$ . For this purpose, Eq. (3-5) is expanded about  $\mathbf{z}_0$  using Taylor series, which yields

$$\mathbf{C}\Delta\mathbf{z} = \mathbf{f}(\mathbf{z}_0 + \Delta\mathbf{z}) = \mathbf{f}(\mathbf{z}_0) + \mathbf{A}\Delta\mathbf{z} + O(\|\Delta\mathbf{z}\|^2) \quad (4-1)$$

where, matrix  $\mathbf{A}$  is the Jacobian of  $\mathbf{f}$  about  $\mathbf{z}_0$  and  $O(\|\Delta\mathbf{z}\|^2)$  represents the remaining nonlinear terms. This equation can be simplified by neglecting nonlinear terms, which results in the following linearized DAEs:

$$\begin{aligned} \mathbf{u}_1 &= \mathbf{f}(\mathbf{z}_0), \\ L_1 : \mathbf{C}\Delta\dot{\mathbf{z}}_1 &= \mathbf{A}\Delta\mathbf{z}_1 + \mathbf{u}_1. \end{aligned} \quad (4-2)$$

Here,  $L_1$  denotes the first linear system and  $\Delta\mathbf{z}_1$  is the solution of this system. This solution approximates  $\Delta\mathbf{z}$  in the vicinity of  $\mathbf{z}_0$  and later deviates from the real solution. In particular, when the nonlinear terms are significant, the approximation provided by  $\Delta\mathbf{z}_1$  is valid only for few steps. In this situation, it is required to consider nonlinear terms to improve the approximation, which leads to the second linear system.

Let  $\mathbf{e}_1$  represents the difference between  $\Delta\mathbf{z}$  and  $\Delta\mathbf{z}_1$  as

$$\mathbf{e}_1 = \Delta\mathbf{z} - \Delta\mathbf{z}_1. \quad (4-3)$$

Taking time derivative of (4-3) and multiplying both sides by  $\mathbf{C}$  results in:

$$\mathbf{C}\dot{\mathbf{e}}_1 = \mathbf{C}\Delta\dot{\mathbf{z}} - \mathbf{C}\Delta\dot{\mathbf{z}}_1. \quad (4-4)$$

By substituting (4-1) and (4-3) into (4-4) and expanding the derivative function using Taylor series, the dynamics of  $\mathbf{e}_1$  can be derived as:

$$\begin{aligned} \mathbf{C}\dot{\mathbf{e}}_1 &= \mathbf{f}(\mathbf{z}_0 + \Delta\mathbf{z}_1 + \mathbf{e}_1) - \mathbf{C}\dot{\mathbf{x}}_1 \\ &\approx \mathbf{f}(\mathbf{z}_0 + \Delta\mathbf{z}_1) + \mathbf{A}\mathbf{e}_1 - \mathbf{C}\Delta\dot{\mathbf{z}}_1 + O(\|\mathbf{e}_1\|^2) \end{aligned} \quad (4-5)$$

Similar to the first linear system, a linear time-invariant approximation of (4-5) is obtained by neglecting higher order terms:

$$\mathbf{u}_2 = \mathbf{f}(\mathbf{z}_0 + \Delta\mathbf{z}_1) - \mathbf{C}\Delta\dot{\mathbf{z}}_1, \quad (4-6)$$

$$L_2 : \mathbf{C}\Delta\dot{\mathbf{z}}_2 = \mathbf{A}\Delta\mathbf{z}_2 + \mathbf{u}_2,$$

where,  $L_2$  represents the second linear system;  $\mathbf{u}_2$  is the input to this system; and  $\Delta\mathbf{z}_2$  is the solution of the system, which approximates  $\mathbf{e}_1$ . This solution can be added to  $\Delta\mathbf{z}_1$  to derive a better approximation of  $\Delta\mathbf{z}$ , i.e.  $\Delta\mathbf{z} \approx \Delta\mathbf{z}_1 + \Delta\mathbf{z}_2$ .

Similarly, the terms neglected in (4-5) can be considered to formulate the third linear system. Assume that  $\mathbf{e}_2$  represents the error of the approximation provided by the first two linear systems, i.e.:

$$\mathbf{e}_2 = \Delta\mathbf{z} - (\Delta\mathbf{z}_1 + \Delta\mathbf{z}_2). \quad (4-7)$$

The dynamics of  $\mathbf{e}_2$  are derived by taking time-derivative of (4-7) and multiplying both sides by  $\mathbf{C}$  as:

$$\mathbf{C}\dot{\mathbf{e}}_2 = \mathbf{C}\Delta\dot{\mathbf{z}} - (\mathbf{C}\Delta\dot{\mathbf{z}}_1 + \mathbf{C}\Delta\dot{\mathbf{z}}_2). \quad (4-8)$$

Substituting (4-1) in (4-8) leads to the following:

$$\begin{aligned} \mathbf{C}\dot{\mathbf{e}}_2 &= \mathbf{f}(\mathbf{z}_0 + \Delta\mathbf{z}_1 + \Delta\mathbf{z}_2 + \mathbf{e}_2) - (\mathbf{C}\Delta\dot{\mathbf{z}}_1 + \mathbf{C}\Delta\dot{\mathbf{z}}_2) \\ &\approx \mathbf{f}(\mathbf{z}_0 + \Delta\mathbf{z}_1 + \Delta\mathbf{z}_2) + \mathbf{A}\mathbf{e}_2 - (\mathbf{C}\Delta\dot{\mathbf{z}}_1 + \mathbf{C}\Delta\dot{\mathbf{z}}_2) + O(\|\mathbf{e}_2\|^2) \end{aligned} \quad (4-9)$$

Then, the third linear system is derived by neglecting higher order terms in (4-9) as:

$$\mathbf{u}_3 = \mathbf{f}(\mathbf{z}_0 + \Delta\mathbf{z}_1 + \Delta\mathbf{z}_2) - (\mathbf{C}\Delta\dot{\mathbf{z}}_1 + \mathbf{C}\Delta\dot{\mathbf{z}}_2), \quad (4-10)$$

$$L_3 : \mathbf{C}\Delta\dot{\mathbf{z}}_3 = \mathbf{A}\Delta\mathbf{z}_3 + \mathbf{u}_3.$$

The described procedure can be generalized to increase number of linear systems. Overall, the  $i$  th linear system has the following general form:

$$\mathbf{z}_{i-1} = \mathbf{z}_0 + \sum_{j=1}^{i-1} \Delta\mathbf{z}_j,$$

$$\mathbf{u}_i = \mathbf{f}(\mathbf{z}_{i-1}) - \sum_{j=1}^{i-1} \mathbf{C}\Delta\dot{\mathbf{z}}_j, \quad (4-11)$$

$$L_i : \mathbf{C}\Delta\dot{\mathbf{z}}_i = \mathbf{A}\Delta\mathbf{z}_i + \mathbf{u}_i,$$

where,  $\mathbf{u}_i$  is the input to  $L_i$ , and  $\Delta\mathbf{z}_i$  is the solution of  $L_i$ . In (4-11),  $\Delta\mathbf{z}_i$  estimates the error in the approximate solution provided by previous linear systems. Therefore, it can be added to solution of previous linear systems to improve accuracy of approximation. Ideally, if infinite number of linear systems are used, the final solution  $\mathbf{z}$  can be reconstructed as:

$$\mathbf{z} = \mathbf{z}_0 + \sum_{i=1}^{\infty} \Delta\mathbf{z}_i. \quad (4-12)$$

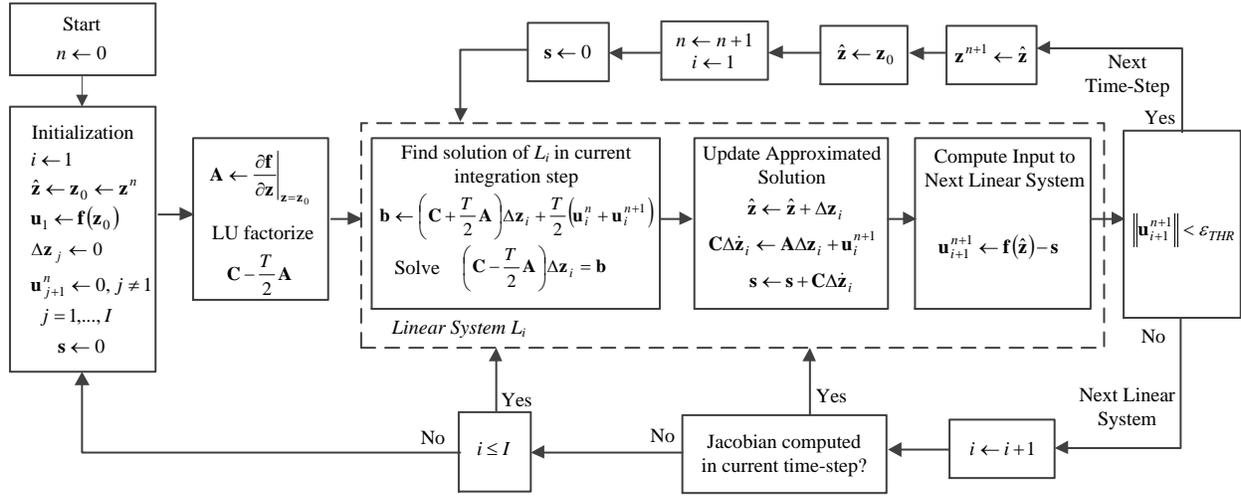
However, in practice, a limited number of systems are sufficient to achieve the desired accuracy.

## 4.2 Practical Considerations and Implementation

In the method described in previous section, the Jacobian matrix  $\mathbf{A}$  is constructed by linearizing derivative function about initial conditions. As the simulation proceeds and the actual solution deviates from the initial conditions, this Jacobian becomes less accurate and subsequently, newer linear systems will be required to maintain the accuracy of approximation. While the method presented has no limitation in terms of number of linear systems, it might become computationally inefficient if large number of linear systems are used to approximate the solution. Therefore, it is more appropriate to update the Jacobian matrix to reduce number of linear systems. However, construction of Jacobian is time-consuming and more importantly, each new Jacobian needs to be LU factorized (which is used later to integrate the linear systems), which implies that frequent update of Jacobian matrix can also be computationally expensive and should be avoided. Similar to the VDHN method, it is possible to keep the Jacobian matrix constant for several integration steps until the convergence rate slows down. For this purpose, a maximum number of linear systems may be considered, and the Jacobian matrix is then updated only once the number of iterations exceeds this value.

The block diagram representation of implementation of the proposed SLIT is shown in Figure 4-1. In this figure, the trapezoidal integration rule is used to integrate the linear systems; however, it should be noted that any implicit integration technique can be used here. The SLIT is started by initializing linear systems. Since the solutions of linear systems approximate  $\Delta \mathbf{z}$  (which is initially zero), the vectors  $\Delta \mathbf{z}_j$  are set to zero to ensure that linear systems have zero initial conditions. However, note that the vector  $\hat{\mathbf{z}}$ , which

approximates  $\mathbf{z}$ , is initialized by  $\mathbf{z}_0$ . Also, the inputs to all linear systems are initially zero except the first one, which has a constant input that equals to  $\mathbf{f}(\mathbf{z}_0)$ . Next, the Jacobian matrix is constructed about  $\mathbf{z}_0$  and LU factors of  $\mathbf{C} - \frac{T}{2}\mathbf{A}$  are computed (which will be used later to integrate the linear systems). The initialized vectors and matrices are passed to the linear systems to compute the first approximate solution. As the first step inside linear system  $i$ , the trapezoidal rule is employed to discretize the linearized DAEs (4-11) and find the solution of  $L_i$  in the current integration step, i.e.  $\Delta\mathbf{z}_i$ . This solution is added to  $\hat{\mathbf{z}}$  to refine the approximation. This step also updates the vector  $\mathbf{s}$ , which represents the term  $\sum_{j=1}^i \mathbf{C}\Delta\dot{\mathbf{z}}_j$  in (4-11). This vector is then subtracted from  $\mathbf{f}(\hat{\mathbf{z}})$  to find the input to the next linear system at  $t = (n+1)T$ , i.e.  $\mathbf{u}_{i+1}^{n+1}$ . If  $\|\mathbf{u}_{i+1}^{n+1}\|$  is smaller than the specified threshold  $\varepsilon_{THR}$ , this would imply that the algorithm has converged and the solution at  $t = (n+1)T$  has been found within acceptable accuracy. Otherwise, the next linear system should be employed to further improve the approximation. At this point, if a Jacobian matrix has already been computed in the current integration step, the algorithm directly goes to the next linear system without updating the Jacobian. Otherwise, the current number of linear systems is checked and if the maximum number of linear systems (i.e.  $I$ ) is reached a new Jacobian is constructed.



**Figure 4-1 Block diagram representation of implementation of the SLIT assuming trapezoidal integration rule.**

In the proposed implementation of SLIT, the norm of  $\mathbf{u}_{i+1}^{n+1}$  is used as the error function to evaluate the accuracy of approximation and detect convergence of the algorithm. Alternatively, one might choose to use the conventional trapezoidal convergence criterion to ensure that the approximate solution is accurate within the specified threshold. This criterion is described by the following condition:

$$\mathbf{h}(\hat{\mathbf{z}}) < \varepsilon_{THR}, \quad (4-13)$$

where  $\mathbf{h}$  represents nonlinear equations presented in (3-1). Both of these criteria can be equally used in the SLIT and they are expected to give the same level of accuracy. However, using  $\|\mathbf{u}_{i+1}^{n+1}\|$  is computationally more efficient as construction of  $\mathbf{h}(\hat{\mathbf{z}})$  requires an additional evaluation of derivate function.

### 4.3 Using Multi-Core Processor Architecture with SLIT

In the proposed SLIT, there is a unidirectional dependency between linear systems, i.e. each linear system depends only on its previous systems. This feature can be easily shown for the first two linear systems. According to (4-2), solution of  $L_1$  depends on  $\mathbf{z}_0$  and  $\mathbf{A}$ . As shown in Figure 4-1, as long as the number of linear systems used inside SLIT does not exceed  $I$ , the parameters  $\mathbf{z}_0$  and  $\mathbf{A}$  remain constant and the solution of the first LTI system will be independent from solutions of next linear systems. Similarly, (4-6) shows that the solution of the second linear system depends only on  $\mathbf{A}$  and solution of the first linear system.

In multi-core processor architecture, this feature can be utilized to distribute the computational tasks among several processing units, as shown in Figure 4-2. In this figure, it is assumed that  $N$  processors are available. In this scheme, processor  $i$  performs the tasks shown inside the dashed rectangle in Figure 4-1, i.e. computing the solution of  $i$ th linear system using a forward/backward substitution and finding the input to  $(i+1)$ th linear system. Once calculated, these results are passed to the next processor to perform the similar steps and compute further refinement to the solution using the next linear system.

In Figure 4-2, each processor waits until a new solution is generated by previous processor. For example, the second processor waits until  $\Delta \mathbf{z}_1^1$  and  $\mathbf{u}_2^1$  are calculated by the first processor, and then it starts computing  $\Delta \mathbf{z}_2^1$  and  $\mathbf{u}_3^1$ . In this configuration, while the second

processors is computing  $\Delta \mathbf{z}_2^1$ , the first processor can work on computing  $\Delta \mathbf{z}_1^2$ , which means that two linear systems are solved concurrently. Similarly, after  $N - 1$  initial steps,  $N$  processors will be busy. The last processor monitors  $\|\mathbf{u}_{N+1}^n\|$ , and if it exceeds  $\varepsilon_{THR}$ , a flag signal is sent to all processors to stop working and then, the algorithm updates  $\mathbf{z}_0$  and  $\mathbf{A}$ .

Since different linear systems are solved by different processors, there will be no time-advantage in skipping some linear systems if the algorithm converges in less than  $N$  systems. Therefore,  $N$  linear systems are solved for all integration steps and the convergence criterion is checked only when the last processor completes its computations. This strategy helps to slightly improve the accuracy of solution. Moreover, in the parallel structure shown in Figure 4-2, each processor solves a system of linear equations in each integration step, which opens the possibility of using available parallel-in-space methods such as Gauss-Seidel [49], SOR [56], or Block Gauss-Jacobi [58]-[59] to employ more processors and achieve further speed improvement.

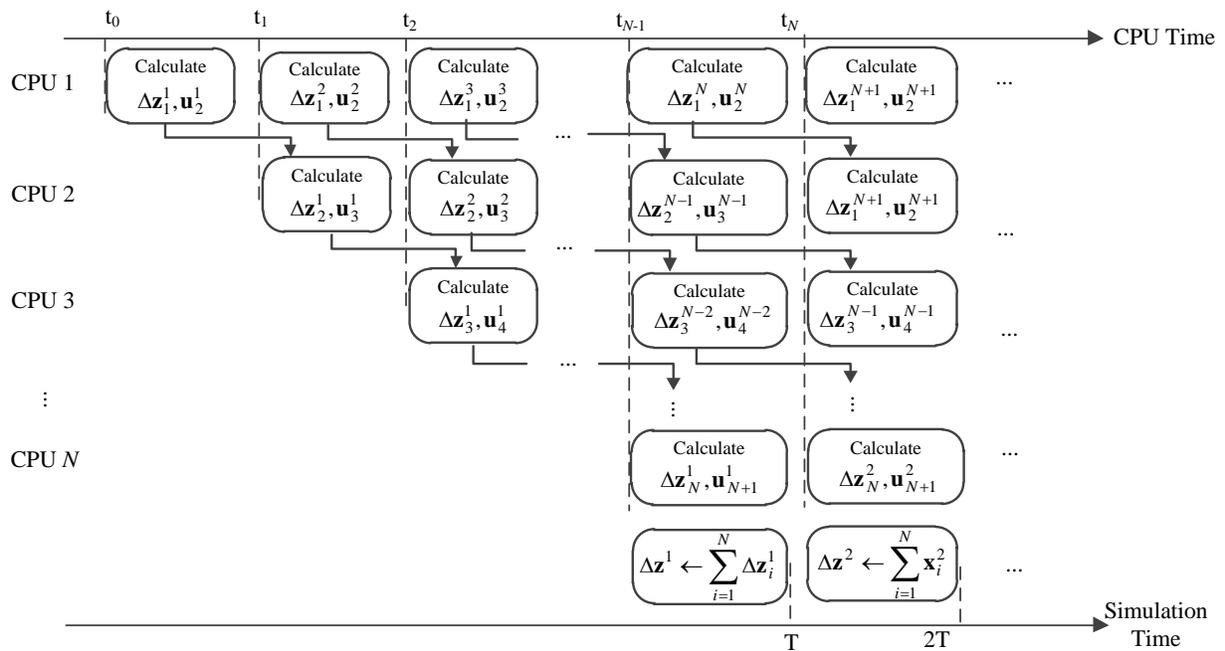


Figure 4-2 Sharing computational tasks among  $N$  processors in the parallel SLIT.

#### 4.4 Handling Saturation Function with Hard Limits

Similar to Newton-type methods, the proposed SLIT is a derivative-based method and the first-order partial derivatives of the derivative function are required to solve the linear systems. However, in transient stability analysis, saturation functions with hard limits are frequently used in control systems of the generators and FACTS devices. Such saturation functions do not have continuous derivatives at certain points. To illustrate this, a saturation function with hard limits at  $x_l$  and  $x_u$  is shown in Figure 4-3. In this figure,  $x$  and  $y$  represent input and output of the function, respectively. The function shown in Figure 4-3 consists of three sections: B1, A, and B2. The derivative of the function in each of these sections is plotted in Figure 4-4, where it can be seen that the derivative is not continuous at points  $(x_l, l_l)$  and  $(x_u, u_l)$ . Whenever solution jumps from section A to B1 or

B2 and vice versa, the corresponding entries in the Jacobian matrix are updated, which subsequently requires a new LU factorization of the Jacobian matrix.

In order to avoid excessive Jacobian updates, the derivative of saturation function can be approximated by a continuous function. This approach might slow down the convergence rate about the discontinuity points. However, it helps to reduce the number of updates of Jacobian matrix, which subsequently reduces the computational penalty when the operating point is on a change in saturation function. For this purpose, the derivative of a saturation function with hard limits may be approximated using a family of continuous functions described by the following:

$$D(x) = \frac{1}{1 + \left( \frac{2(x - x_u)}{x_u - x_l} + 1 \right)^{2n}}, \quad n = 1, 2, \dots \quad (4-14)$$

Function  $D(x)$  is plotted for different values of  $n$  in Figure 4-5. As shown, for large values of  $n$ , the difference between exact derivative and  $D(x)$  becomes visible only in the vicinity of discontinuity points. Moreover, while the accuracy of  $D(x)$  improves as  $n$  increases, the improvement becomes less visible for large values of  $n$ . Therefore, choosing  $n \geq 4$  will be accurate enough in transient stability analysis.

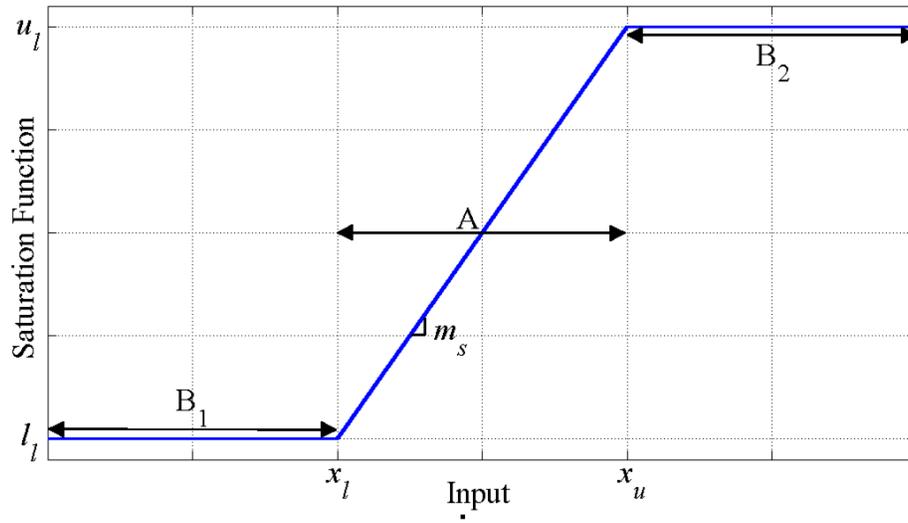


Figure 4-3 Saturation function with hard limits.

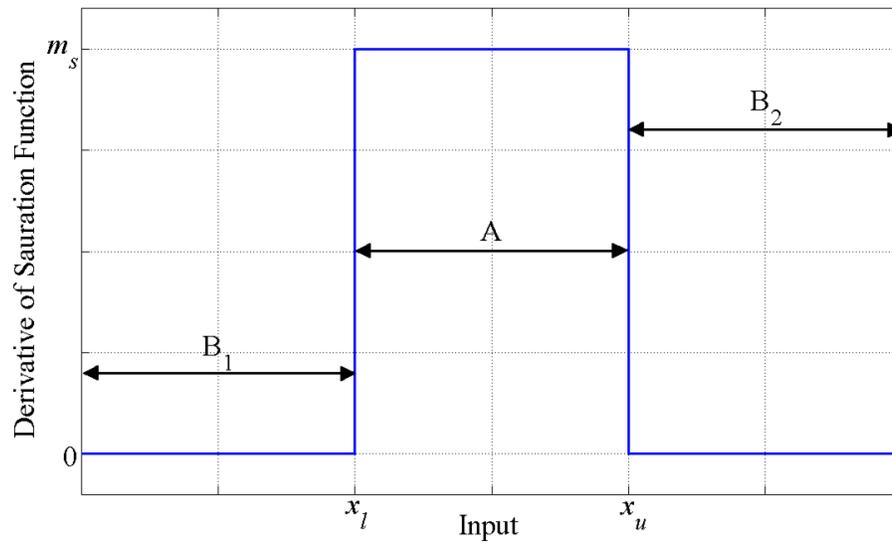


Figure 4-4 Derivative of saturation function with hard limits.

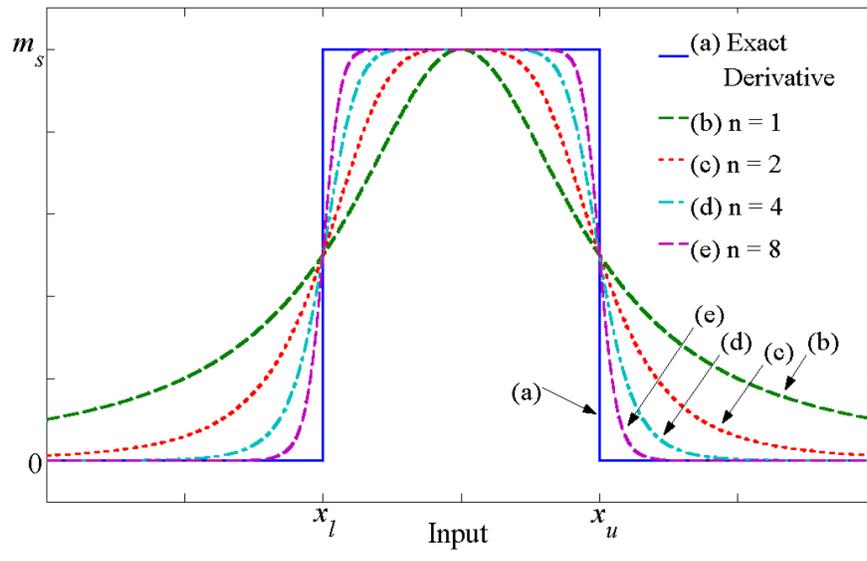


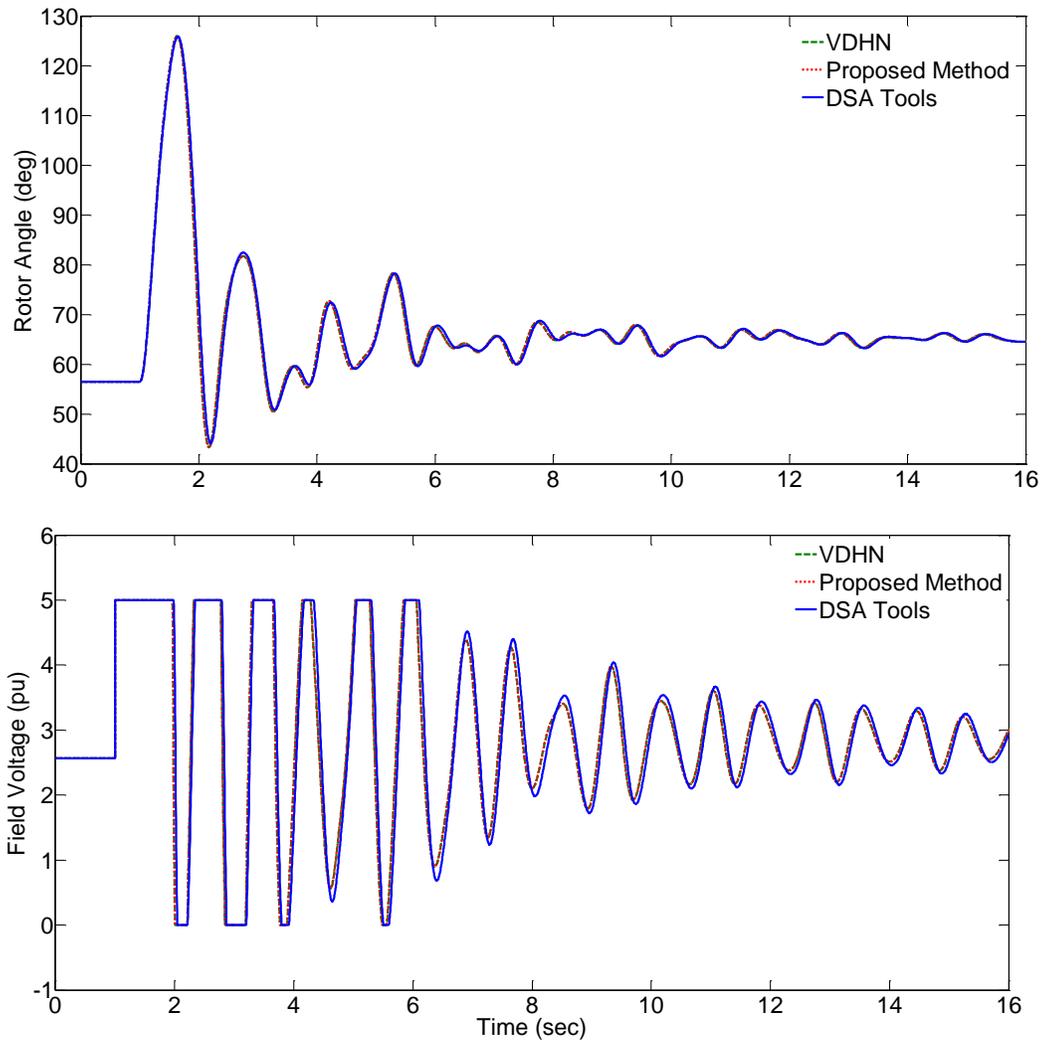
Figure 4-5 Approximating derivative of saturation function using function  $D(x)$ .

## 4.5 Benchmark System and Simulation Results

The effectiveness of the proposed SLIT method is evaluated using the IEEE transient stability test system described in section 2.1.6. The DSATools is used to generate the reference TDS solution. For comparison purposes, both the VDHN and the SLIT methods have been coded in standard C. In both methods, the solution tolerance within the time step iterations is set to  $10^{-4}$ . Also, in both VDHN and SLIT methods, the Jacobian matrix is reused for multiple time steps and is forced to be updated when the number of iterations (linear systems) exceeds 4. Additionally, a parallel version of the SLIT was also implemented in standard C. For consistency of benchmarking, the implicit trapezoidal integration rule with fixed time-step of  $\Delta t = 4ms$  is used for all methods. All simulations are performed using a desktop computer with 16GB RAM and 8-core Intel Xeon processor running at 2.00 GHz.

### 4.5.1 Case study

In order to evaluate accuracy of SLIT method, a three-phase fault is applied at time  $t = 1s$  at bus #7. The fault is cleared after 8 cycles by opening transmission line between buses #7 and #6, and the post-fault system is simulated for 15 seconds. Since the generator at bus #111 is highly affected by this fault, the rotor angle and field voltage of this generator are plotted in Figure 4-6. As can be seen, during the fault, the generator rotor angle increases to almost  $130^\circ$  and the field voltage stays at its maximum limit (5pu). After the fault is cleared, the generator decelerates and the rotor angle decreases, which implies that this scenario is stable. The post-fault oscillations are more apparent in field voltage, which hits the upper and lower limits several times. The results in Figure 4-6 have been computed using the DSATools, VDHN, and SLIT method. These transient responses are very close to each other, which verifies that all methods are in good agreement.



**Figure 4-6 Rotor angle and field voltage of generator at bus #111.**

### **4.5.2 Verification and accuracy evaluation**

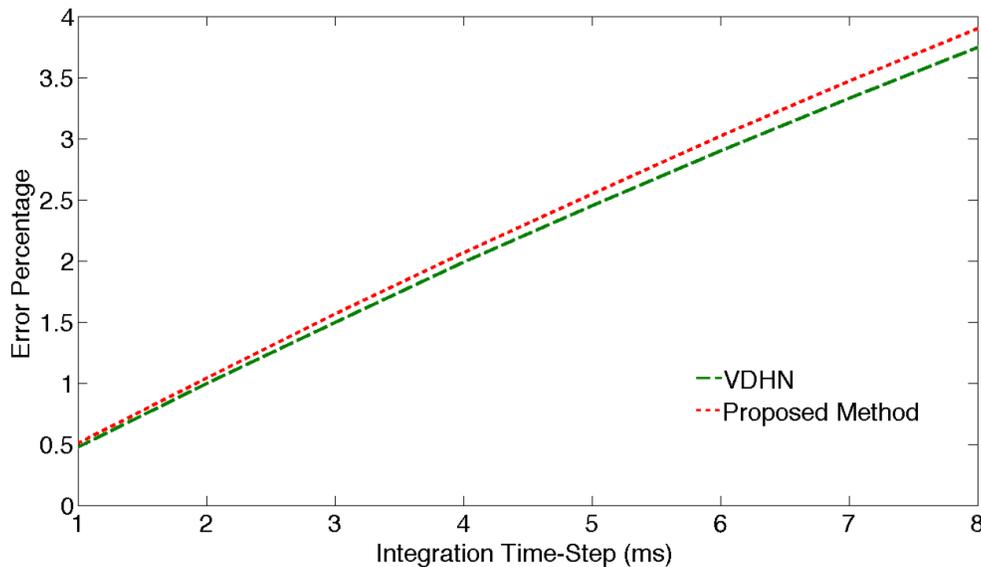
Next, we investigate the accuracy of the proposed SLIT method for different integration time step sizes. For this purpose, the fault scenario described in Section 4.5.1 is solved using different integration time-steps from 1 to 8 ms, and the results are compared against the reference solution, which is obtained using Full-Newton method with 0.1ms time-step. Since the bus #111 is considered to be highly disturbed, only its generator field voltage is

considered for the error calculations. Here, the 2-norm (cumulative) relative error [109] of the field voltage  $\mathbf{E}^{fd}$  is computed as

$$e(T) = \frac{\|\mathbf{E}_{ref}^{fd} - \mathbf{E}^{fd,T}\|_2}{\|\mathbf{E}_{ref}^{fd}\|_2} \times 100\% , \quad (4-15)$$

where superscript  $T$  represents integration time-step size;  $\mathbf{E}_{ref}^{fd}$  is the reference solution; and  $\mathbf{E}^{fd,T}$  is the field voltage solution trajectory computed by the SLIT (or the VDHN), using a given time step  $T$ . In (4-15), the field voltage  $\mathbf{E}^{fd}$  can be replaced by other variables in the system; however, this variable was chosen as to examine the worst case.

The results are plotted in Figure 4-7. As predicted, the error in both SLIT and VDHN methods decreases as smaller integration time-steps are used, which verifies that both methods are consistent and converge to the same solution. The VDHN method shows a slightly better accuracy, but this difference is very small.



**Figure 4-7 Percentage of integration error in VDHN and SLIT methods.**

### 4.5.3 Computational performance using large-scale system

In this section, computational performance of sequential and parallel implementation of SLIT is compared against the conventional VDHN method as well as the MDA. For this study, a larger test system is created by duplicating and interconnecting previous test system four times as shown in Figure 4-8, wherein the adjacent subsystems are connected using 6 transmission lines. The new test system has 200 generators and 580 buses. Additionally, 14 scenarios described in Table 3-1 are considered in this section.

The CPU times achieved by different methods for all 14 scenarios are summarized in Table 4-1, wherein the first column represents scenario number according to Table 3-1. The other columns in Table 4-1 show the computational time achieved by VDHN, sequential MDA, and sequential SLIT, respectively. Also, the last two columns show the speed-up ratios, which are defined as follows:

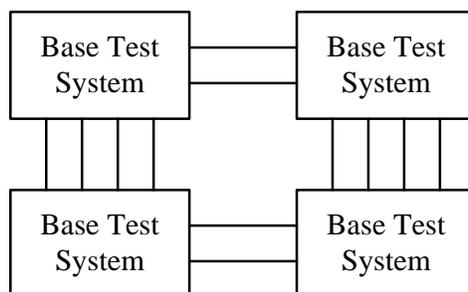
$$R_{Seq}^{MDA} = \frac{T_{VDHN}}{T_{MDA}}, \quad (4-16)$$

$$R_{Seq}^{SLIT} = \frac{T_{VDHN}}{T_{SLIT}}, \quad (4-17)$$

where  $T_{VDHN}$ ,  $T_{MDA}$ , and  $T_{SLIT}$  represent CPU time of VDHN, sequential MDA, and sequential SLIT, respectively. The last row in Table 4-1 shows the average time required to solve a scenario, where it can be seen that MDA and SLIT are faster than VDHN by 6% and 21%, respectively.

To further analyze why SLIT is faster than VDHN and MDA, it is instructive to compare the number of LU factorizations performed in the given scenarios. The results are presented in Table 4-2, wherein it is shown that the SLIT generally performs less LU factorizations which also demonstrates that this method has a better convergence rate. Since LU factorization is a computationally demanding task, it can be observed that the speed-up ratio becomes more significant as the difference between number of LU factorizations performed in VDHN and SLT methods increases.

The computational performance of parallel MDA (MDA\_P) and parallel SLIT (SLIT\_P) is also compared against the VDHN method. In this study, the maximum number of processors used concurrently in MDA\_P and SLT\_P is 3 and 6, respectively. The results are summarized in Table 4-3, wherein  $R_{Par}^{MDA}$  and  $R_{Par}^{SLIT}$  represent the speed-up ratio achieved by the MDA\_P and SLIT\_P, respectively. As it can be seen in Table 4-3, the MDA\_P and SLIT\_P are faster than VDHN by 2.19 and 3.32 times, respectively. These results confirm that the proposed SLIT\_P method outperforms the VDHN and has superior scalability compared to the MDA\_P as it can take advantage of more processors.



**Figure 4-8 Duplicating base test system to create a large-scale system.**

**Table 4-1 CPU Execution Time for Different Scenarios.**

Scenario Number	VDHN (sec)	MDA (sec)	SLIT (sec)	$R_{Seq}^{MDA}$	$R_{Seq}^{SLIT}$
1	2.764	2.632	2.412	1.050	1.15
2	2.803	2.659	2.428	1.054	1.15
3	1.831	1.701	1.940	1.076	0.94
4	2.847	2.473	2.201	1.151	1.29
5	1.903	1.715	2.050	1.110	0.93
6	4.949	4.304	2.993	1.150	1.65
7	3.162	2.847	2.510	1.111	1.26
8	4.252	3.926	2.896	1.083	1.47
9	2.625	2.703	2.882	0.971	0.91
10	3.839	3.800	2.776	1.010	1.38
11	2.911	2.877	2.363	1.012	1.23
12	3.261	3.306	2.860	0.986	1.14
13	3.328	3.284	2.854	1.013	1.17
14	3.709	3.630	2.852	1.022	1.30
<b>Average Time</b>	<b>3.16</b>	<b>2.99</b>	<b>2.57</b>	<b>1.06</b>	<b>1.21</b>

**Table 4-2 Comparing Number of Jacobian Calculations and LU Factorization Performed in Different Methods.**

Scenario Number	$R_{Seq}^{SLIT}$	No. of LU Factorization		
		VDHN	MDA	SLIT
9	0.91	119	622	157
5	0.93	94	188	78
3	0.94	92	191	78
11	1.23	328	625	240
4	1.29	279	490	150
12	1.14	405	827	248
1	1.15	524	890	274
2	1.15	531	908	274
7	1.26	624	714	329
13	1.17	652	1270	339
14	1.30	991	1633	471
10	1.38	971	1746	428
8	1.47	1087	1360	470
6	1.65	1293	1424	554

**Table 4-3 CPU Execution Time Of Parallel MDA and Parallel SLIT**

Scenario Number	VDHN (sec)	MDA_P (sec)	SLIT_P (sec)	$R_{par}^{MDA}$	$R_{par}^{SLIT}$
1	2.764	1.189	0.79	2.32	3.50
2	2.803	1.190	0.83	2.36	3.37
3	1.831	0.798	0.59	2.29	3.12
4	2.847	1.038	0.73	2.74	3.88
5	1.903	0.772	0.47	2.47	4.02
6	4.949	2.500	1.16	1.98	4.28
7	3.162	1.631	0.91	1.94	3.48
8	4.252	2.077	1.10	2.05	3.88
9	2.625	1.206	1.21	2.18	2.17
10	3.839	1.904	1.25	2.02	3.08
11	2.911	1.367	0.98	2.13	2.96
12	3.261	1.578	1.23	2.07	2.64
13	3.328	1.558	1.20	2.14	2.76
14	3.709	1.827	1.10	2.03	3.37
<b>Average Time</b>	<b>3.16</b>	<b>1.474</b>	<b>0.97</b>	<b>2.19</b>	<b>3.32</b>

# CHAPTER 5: USING PHASOR MEASUREMENT UNITS IN REAL-TIME TSA

## 5.1 Problem Statement

For the purpose of DSE, we assume that the generators are represented by detailed model. Ignoring the fast sub-transient dynamics, the model of synchronous machine  $i$  equipped with the AC-Type 4 excitation system can be described by the forth-order differential equations as in (5-1) [17]:

$$\begin{aligned}
 T_{do,i} \frac{dE'_{q,i}}{dt} &= -E'_{q,i} - (X_{d,i} - X'_{d,i})I_{d,i} + L(E_{fd,i}), \\
 T_{qo,i} \frac{dE'_{d,i}}{dt} &= -E'_{d,i} + (X_{q,i} - X'_{q,i})I_{q,i}, \\
 \frac{d\delta_i}{dt} &= \omega_i - \omega_{s,i}, \\
 M_i \frac{d\omega_i}{dt} &= T_{M,i} - D_i(\omega_i - \omega_{s,i}) - E'_{q,i}I_{q,i} + E'_{d,i}I_{d,i} + (X_{q,i} - X_{d,i})I_{d,i}I_{q,i}, \\
 T_{a,i} \frac{dE_{fd,i}}{dt} &= K_{a,i} \frac{T_{c,i}}{T_{b,i}} (V_{ref,i} - V_i) - E_{fd,i} \left( 1 + K_{a,i} S_{e1,i} e^{S_{e2,i} E_{fd,i}} + K_{a,i} R_{f,i} \left( 1 - \frac{T_{c,i}}{T_{b,i}} \right) \right), \\
 T_{b,i} \frac{dR_{f,i}}{dt} &= -R_{f,i} + V_{ref,i} - V_i,
 \end{aligned} \tag{5-1}$$

where:  $E_{fd}$  represents the internal field voltage;  $E'_q$  and  $E'_d$  represent transient voltage along the  $q$  and  $d$  axes, respectively;  $\delta$  and  $\omega$  represent rotor angle and rotor speed of synchronous machine, respectively;  $I_q$  and  $I_d$  represent terminal currents;  $V$  represents magnitude of terminal voltage; and function  $L(\cdot)$  is the saturation function that limits the

$E_{fd}$  to physical limits of the excitation system. The rest of parameters are defined in Table 5-1.

**Table 5-1 Nomenclature of Machine Parameters**

$T'_{qo}$	open circuit time-constant along $q$ axis
$T'_{do}$	open circuit time-constant along $d$ axis
$X'_q$	transient reactance along along $q$ axis
$X'_d$	transient reactance along along $d$ axis
$\omega_s$	rated angular frequency
$M$	machine's inertia
$D$	damping factor
$T_m$	input mechanical torque
$T_a$	time-constant associated with the regulator and/or firing of thyristors
$K_a$	gain associated with the regulator and/or firing of thyristors
$T_b$	time-constants of the lag-lead controller used inside the excitation system
$T_c$	time-constants of the lag-lead controller used inside the excitation system
$S_{e1}$	coefficient of saturation function
$S_{e2}$	coefficient of saturation function
$V_{ref}$	reference and terminal voltage

The stator equations relate terminal voltage and current to the states of the synchronous machine as follows:

$$\begin{aligned}
 E'_{d,i} - V_i \sin(\delta_i - \theta_i) - R_{s,i} I_{d,i} + X'_{q,i} I_{q,i} &= 0, \\
 E'_{q,i} - V_i \cos(\delta_i - \theta_i) - R_{s,i} I_{q,i} - X'_{d,i} I_{d,i} &= 0,
 \end{aligned}
 \tag{5-2}$$

where  $\theta$  represents angle of terminal voltage with respect to network's synchronous reference frame. Overall, the machine equations can be described in the following compact form:

$$\mathbf{x}_i = [E'_{q,i}, E'_{d,i}, \delta_i, \omega_i, E_{fd,i}, R_{f,i}]^T,$$

$$\mathbf{a}_i = [V_i, I_{d,i}, I_{q,i}]^T,$$

$$\dot{\mathbf{x}}_i = \mathbf{f}_{c,i}(\mathbf{x}_i, \mathbf{a}_i), \quad (5-3)$$

$$0 = \mathbf{g}_i(\mathbf{x}_i, \mathbf{a}_i), \quad (5-4)$$

where  $\mathbf{f}_{c,i}$  represents differential equations (5-1) and  $\mathbf{g}_i$  represents stator equations (5-2).

In the local DSE [60] and [65], a PMU is installed at the terminal bus of generator and it measures the vector  $\mathbf{a}_i$ . Then, these measurements along with the generator model (5-3)-(5-4) are used to estimate state vector  $\mathbf{x}_i$ . Ref. [60] uses the EKF and [65] uses the EPF approach to estimate the states of the generator. In contrast, a global DSE utilizes all PMU measurements throughout the network to estimate states of all generators. Since more information is available in the global DSE, it is expected to achieve a better accuracy in estimating states.

In order to formulate the global DSE, it is required to add the model of transmission network to the generator model. As discussed in section 2.1.1, the generator model and the network equations form a set of nonlinear Differential Algebraic Equations (DAEs), which can be described as follows:

$$\mathbf{x} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_{n_1}^T]^T,$$

$$\mathbf{a} = [\mathbf{a}_1^T, \mathbf{a}_2^T, \dots, \mathbf{a}_{n_2}^T]^T,$$

$$\dot{\mathbf{x}} = \mathbf{f}_c(\mathbf{x}, \mathbf{a}), \quad (5-5)$$

$$0 = \mathbf{g}(\mathbf{x}, \mathbf{a}), \quad (5-6)$$

where,  $n_1$  and  $n_2$  represent number of generators and buses, respectively;  $\mathbf{x}$  represents state variables of all machines;  $\mathbf{a}$  represents the algebraic variables that encompasses phasors of voltages and currents; the vector function  $\mathbf{f}_c$  includes the derivative functions of machines' models; and the algebraic function  $\mathbf{g}$  represents the stator equations (5-2) as well as network power flow constraints.

In the global DSE, some or all of the algebraic variables are observed by PMU measurements, which are used to estimate state vector  $\mathbf{x}$ . Since the PMU measurements always carry some level of noise, it is appropriate to rewrite (5-5)-(5-6) as a stochastic model as follows:

$$\dot{\mathbf{x}} = \mathbf{f}_c(\mathbf{x}, \tilde{\mathbf{a}} + \mathbf{v}) + \mathbf{w}, \quad (5-7)$$

$$0 = \mathbf{g}(\mathbf{x}, \tilde{\mathbf{a}} + \mathbf{v}). \quad (5-8)$$

where,  $\tilde{\mathbf{a}}$  represents measured algebraic variables;  $\mathbf{v}$  is the measurement noise; and  $\mathbf{w}$  represents inaccuracies in the model and/or parameters.

## 5.2 Applying Extended Kalman Filter to DSE Problem

### 5.2.1 Review of extended kalman filter

Suppose a nonlinear dynamic system is described by a pair of nonlinear stochastic process and measurement equations as:

$$\mathbf{x}_k = \mathbf{f}_d(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}), \quad (5-9)$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k), \quad (5-10)$$

where,  $k$  represents measurement step;  $\mathbf{x}$  and  $\mathbf{u}$  represent the states and input of the system, respectively;  $\mathbf{z}$  represents vector of measured quantities;  $\mathbf{f}_d$  is the state transition function, which is typically a difference equation that approximates dynamics of original continuous model of the system;  $\mathbf{h}$  is the observation function that relates the states to the measurements;  $\mathbf{w}$  is the process noise; and  $\mathbf{v}$  is the measurement noise. The noise sequences  $\{\mathbf{w}_k\}$  and  $\{\mathbf{v}_k\}$  are assumed to be white, Gaussian, and mutually independent, i.e.:

$$p(\mathbf{w}) \sim N(0, \mathbf{Q}), \quad (5-11)$$

$$p(\mathbf{v}) \sim N(0, \mathbf{R}), \quad (5-12)$$

$$E[\mathbf{v}_k \mathbf{w}_j^T] = 0, \forall k, j, \quad (5-13)$$

$$E[\mathbf{v}_k \mathbf{x}_j^T] = 0, \forall k, j, \quad (5-14)$$

where,  $p$  represents probability distribution function;  $N$  represents a normal distribution function;  $\mathbf{Q}$  and  $\mathbf{R}$  represent process and measurement noise covariance matrices, respectively; and  $E[\cdot]$  represents expected value of a random variable.

For the purpose of describing the EKF approach, let  $\mathbf{x}_k^-$  be a prediction of states at step  $k$  that is derived using the state estimation results from previous step. Based on this definition, predicted estimation error and its covariance matrix are defined as:

$$\mathbf{e}_k^- = \mathbf{x}_k - \mathbf{x}_k^-, \quad (5-15)$$

$$\mathbf{P}_k^- = E\left[\mathbf{e}_k^-(\mathbf{e}_k^-)^T\right]. \quad (5-16)$$

Likewise, let  $\mathbf{x}_k^+$  be an updated estimate of states at step  $k$ , which is computed using the results of previous step as well as the measurements in current step. Based on this definition,

$$\mathbf{e}_k^+ = \mathbf{x}_k - \mathbf{x}_k^+, \quad (5-17)$$

$$\mathbf{P}_k^+ = E\left[\mathbf{e}_k^+(\mathbf{e}_k^+)^T\right], \quad (5-18)$$

are updated estimation error and its covariance matrix, respectively.

The state estimation process is started by choosing an initial guess for  $\mathbf{x}_0^+$  and  $\mathbf{P}_0^+$ . The EKF then estimates the state variables in two stages. In the first stage, which is called the prediction stage, a prediction of states is computed based on system's dynamic model. This step is realized by the following set of equations:

$$\mathbf{x}_k^- = \mathbf{f}_d(\mathbf{x}_{k-1}^+, \mathbf{u}_{k-1}, 0), \quad (5-19)$$

$$\mathbf{P}_k^- = \mathbf{F}_k \mathbf{P}_{k-1}^+ \mathbf{F}_k^T + \mathbf{W}_k \mathbf{Q} \mathbf{W}_k^T, \quad (5-20)$$

where,

$$\mathbf{F}_k = \left. \frac{\partial \mathbf{f}_d}{\partial \mathbf{x}_{k-1}} \right|_{\mathbf{x}_{k-1} = \mathbf{x}_{k-1}^+}, \quad (5-21)$$

$$\mathbf{W}_k = \left. \frac{\partial \mathbf{f}_d}{\partial \mathbf{w}_{k-1}} \right|_{\mathbf{x}_{k-1} = \mathbf{x}_{k-1}^+}. \quad (5-22)$$

The prediction stage essentially uses the dynamic model (5-9) to derive an estimation of states. Due to model inaccuracies and measurement noise, the results might not be accurate enough and therefore, the data measured at step  $k$  are used subsequently to refine  $\mathbf{x}_k^-$ . This stage is called update stage and it is realized by the following set of equations:

$$\mathbf{S} = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}, \quad (5-23)$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{S}^{-1},$$

$$\mathbf{x}_k^+ = \mathbf{x}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{h}(\mathbf{x}_k^-, 0)), \quad (5-24)$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-, \quad (5-25)$$

where,

$$\mathbf{H}_k = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\mathbf{x}_k^-}, \quad (5-26)$$

$$\mathbf{V}_k = \frac{\partial \mathbf{h}}{\partial \mathbf{v}} \Big|_{\mathbf{x}_k^-}. \quad (5-27)$$

Overall, (5-19)-(5-27) represents the formulation of the classic EKF used for estimating states of a dynamic system.

## 5.2.2 Prediction stage in the TSA

The formulation of EKF requires the states at current step to be expressed as explicit function of states and input in the previous step. Therefore, in order to apply the EKF to the TSA problem, the continuous model (5-7) has to be discretized using an explicit integration technique. Explicit Euler method has been used in the literature [60]-[65] to form a discrete model suitable for the EKF. This technique typically provides good results in quasi-steady-state studies. However, when the system experiences a severe disturbance, the accuracy of prediction provided by Euler method can be significantly degraded. Therefore, in this section, a fourth-order Runge-Kutta formula [110] is used to discretize (5-7) as follows:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}_d(\mathbf{x}_{k-1}, \tilde{\mathbf{a}}_{k-1} + \mathbf{v}_{k-1}, \tilde{\mathbf{a}}_k + \mathbf{v}_k, \mathbf{w}_{k-1}) \\ &= \mathbf{x}_{k-1} + \frac{T}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) + \mathbf{w}_{k-1}, \end{aligned} \quad (5-28)$$

where vectors  $\mathbf{k}_1$  to  $\mathbf{k}_4$  are defined as:

$$\mathbf{k}_1 = \mathbf{f}_c(\mathbf{x}_{k-1}, \tilde{\mathbf{a}}_{k-1} + \mathbf{v}_{k-1}), \quad \tilde{\mathbf{x}}_1 = \mathbf{x}_{k-1} + \frac{T}{2}\mathbf{k}_1,$$

$$\tilde{\mathbf{a}}_{1/2} = \frac{1}{2}(\tilde{\mathbf{a}}_{k-1} + \tilde{\mathbf{a}}_k), \quad \tilde{\mathbf{v}}_{1/2} = \frac{1}{2}(\mathbf{v}_{k-1} + \mathbf{v}_k),$$

$$\mathbf{k}_2 = \mathbf{f}_c\left(\tilde{\mathbf{x}}_1, \tilde{\mathbf{a}}_{1/2} + \tilde{\mathbf{v}}_{1/2}\right), \quad \tilde{\mathbf{x}}_2 = \mathbf{x}_{k-1} + \frac{T}{2}\mathbf{k}_2,$$

$$\mathbf{k}_3 = \mathbf{f}_c \left( \tilde{\mathbf{x}}_2, \tilde{\mathbf{a}}_{1/2} + \tilde{\mathbf{v}}_{1/2} \right), \quad \tilde{\mathbf{x}}_3 = \mathbf{x}_{k-1} + T\mathbf{k}_3,$$

$$\mathbf{k}_4 = \mathbf{f}_c \left( \tilde{\mathbf{x}}_3, \mathbf{a}_k + \mathbf{v}_k \right).$$

Using this discretized model (5-28) in the EKF, a prediction of states is computed by setting noises  $\mathbf{v}_{k-1}$ ,  $\mathbf{v}_k$ , and  $\mathbf{w}_{k-1}$  to zero as:

$$\mathbf{x}_k^- = \mathbf{f}_d \left( \mathbf{x}_{k-1}^+, \tilde{\mathbf{a}}_{k-1}, \tilde{\mathbf{a}}_k, 0 \right), \quad (5-29)$$

and the results are used to find a prediction of algebraic variables by solving:

$$0 = \mathbf{g} \left( \mathbf{x}_k^-, \mathbf{a}_k^- \right). \quad (5-30)$$

In order to complete the prediction stage, the covariance matrix  $\mathbf{P}_k^-$  needs to be computed.

However, the partial derivatives of (5-28) with respect to states and noise should be computed first. Since the coefficients  $\mathbf{k}_2$  to  $\mathbf{k}_4$  are related to states, the chain rule can be used to compute the partial derivatives of (5-28) with respect to states as follows:

$$\mathbf{F}_k = \frac{\partial \mathbf{f}_d}{\partial \mathbf{x}_{k-1}} = \mathbf{I} + \frac{T}{6} \left( \frac{\partial \mathbf{k}_1}{\partial \mathbf{x}_{k-1}} + 2 \frac{\partial \mathbf{k}_2}{\partial \mathbf{x}_{k-1}} + 2 \frac{\partial \mathbf{k}_3}{\partial \mathbf{x}_{k-1}} + \frac{\partial \mathbf{k}_4}{\partial \mathbf{x}_{k-1}} \right), \quad (5-31)$$

where,

$$\frac{\partial \mathbf{k}_1}{\partial \mathbf{x}_{k-1}} = \left. \frac{\partial \mathbf{f}_c}{\partial \mathbf{x}} \right|_{\substack{\mathbf{x}=\mathbf{x}_{k-1}^+ \\ \mathbf{a}=\tilde{\mathbf{a}}_{k-1}}},$$

$$\frac{\partial \mathbf{k}_2}{\partial \mathbf{x}_{k-1}} = \frac{\partial \mathbf{k}_1}{\partial \mathbf{x}_{k-1}} \left( \mathbf{I} + \frac{T}{2} \frac{\partial \mathbf{k}_1}{\partial \mathbf{x}_{k-1}} \right),$$

$$\frac{\partial \mathbf{k}_3}{\partial \mathbf{x}_{k-1}} = \frac{\partial \mathbf{k}_1}{\partial \mathbf{x}_{k-1}} \left( \mathbf{I} + \frac{T}{2} \frac{\partial \mathbf{k}_2}{\partial \mathbf{x}_{k-1}} \right),$$

$$\frac{\partial \mathbf{k}_4}{\partial \mathbf{x}_{k-1}} = \frac{\partial \mathbf{k}_1}{\partial \mathbf{x}_{k-1}} \left( \mathbf{I} + T \frac{\partial \mathbf{k}_3}{\partial \mathbf{x}_{k-1}} \right).$$

Here,  $\mathbf{F}_k$  represents the partial derivatives of (5-28) with respect to states. In addition, the partial derivatives of (5-28) with respect to process and measurement noises can be computed as follows:

$$\mathbf{W}_{k-1} = \frac{\partial \mathbf{f}_d}{\partial \mathbf{w}_{k-1}} = \mathbf{I}. \quad (5-32)$$

$$\mathbf{V}_{k-1} = \frac{\partial \mathbf{f}_d}{\partial \mathbf{v}_{k-1}} = \frac{T}{6} \left( \frac{\partial \mathbf{k}_1}{\partial \mathbf{v}_{k-1}} + 2 \frac{\partial \mathbf{k}_2}{\partial \mathbf{v}_{k-1}} + 2 \frac{\partial \mathbf{k}_3}{\partial \mathbf{v}_{k-1}} + \frac{\partial \mathbf{k}_4}{\partial \mathbf{v}_{k-1}} \right), \quad (5-33)$$

$$\mathbf{V}_k = \frac{\partial \mathbf{f}_d}{\partial \mathbf{v}_k} = \frac{T}{6} \left( \frac{\partial \mathbf{k}_1}{\partial \mathbf{v}_k} + 2 \frac{\partial \mathbf{k}_2}{\partial \mathbf{v}_k} + 2 \frac{\partial \mathbf{k}_3}{\partial \mathbf{v}_k} + \frac{\partial \mathbf{k}_4}{\partial \mathbf{v}_k} \right), \quad (5-34)$$

where

$$\frac{\partial \mathbf{k}_1}{\partial \mathbf{v}_{k-1}} = \frac{\partial \mathbf{f}_c}{\partial \mathbf{a}} \bigg|_{\substack{\mathbf{x}=\tilde{\mathbf{x}}_{k-1}^+ \\ \mathbf{a}=\tilde{\mathbf{a}}_{k-1}}},$$

$$\frac{\partial \mathbf{k}_1}{\partial \mathbf{v}_k} = 0,$$

$$\frac{\partial \mathbf{k}_2}{\partial \mathbf{v}_{k-1}} = \frac{\partial \mathbf{k}_2}{\partial \mathbf{v}_k} = \frac{1}{2} \frac{\partial \mathbf{f}_c}{\partial \mathbf{a}} \bigg|_{\substack{\mathbf{x}=\tilde{\mathbf{x}}_1 \\ \mathbf{a}=\tilde{\mathbf{a}}_{1/2}}},$$

$$\frac{\partial \mathbf{k}_3}{\partial \mathbf{v}_{k-1}} = \frac{\partial \mathbf{k}_3}{\partial \mathbf{v}_k} = \frac{1}{2} \frac{\partial \mathbf{f}_c}{\partial \mathbf{a}} \bigg|_{\substack{\mathbf{x}=\tilde{\mathbf{x}}_2 \\ \mathbf{a}=\tilde{\mathbf{a}}_{1/2}}},$$

$$\frac{\partial \mathbf{k}_4}{\partial \mathbf{v}_{k-1}} = 0,$$

$$\frac{\partial \mathbf{k}_4}{\partial \mathbf{v}_k} = \left. \frac{\partial \mathbf{f}_c}{\partial \mathbf{a}} \right|_{\substack{\mathbf{x}=\tilde{\mathbf{x}}_3 \\ \mathbf{a}=\tilde{\mathbf{a}}_k}}.$$

In (5-32)-(5-34),  $\mathbf{V}$  and  $\mathbf{W}$  represent the partial derivatives of (5-28) with respect to measurement and process noise, respectively. Note that in (5-7), the process noise is assumed to be additive, which makes  $\mathbf{W}$  constant identity matrix. However,  $\mathbf{f}_d$  is a nonlinear function of measurements (and therefore measurement noise), which means (5-28) needs to be linearized about the measurement noise to form matrices  $\mathbf{V}_{k-1}$  and  $\mathbf{V}_k$  in each measurement step. The results of (5-31)-(5-34) are then used to find  $\mathbf{P}_k^-$  as:

$$\mathbf{P}_k^- = \mathbf{F}_k \mathbf{P}_{k-1}^+ \mathbf{F}_k^T + \mathbf{V}_{k-1} \mathbf{R} \mathbf{V}_{k-1}^T + \mathbf{V}_k \mathbf{R} \mathbf{V}_k^T + \mathbf{Q}^T \mathbf{W} \mathbf{Q}, \quad (5-35)$$

which completes the prediction stage in the EKF.

### 5.2.3 Update stage in the TSA - simultaneous approach

In the update stage (5-23)-(5-25), the measured quantities are explicit functions of states. However, (5-8) shows that in transient stability problem, the measured quantities (i.e. algebraic variables) are implicitly related to the states through powerflow equations. In this situation, the observation matrix  $\mathbf{H}$  is not available and the update stage cannot be carried out.

This problem can be addressed by developing appropriate formulation for the EKF with implicit observation. Assume  $\mathbf{x}_k^+$  is a linear combination of  $\mathbf{x}_k^-$  and the difference between measured and predicted algebraic variables, i.e.

$$\mathbf{x}_k^+ = \mathbf{x}_k^- + \mathbf{K}_k (\tilde{\mathbf{a}}_k - \mathbf{a}_k^-), \quad (5-36)$$

where,  $\mathbf{K}_k$  is the Kalman gain. In the conventional EKF [111], the Kalman gain is calculated by minimizing trace of  $\mathbf{P}_k^+$ , which ultimately aims at minimizing sum of errors in all measurement steps. Inspired by this method, we derive the formulation of  $\mathbf{P}_k^+$  in the systems with implicit observation. For this purpose, suppose  $\Delta \mathbf{a}$  represents the difference between measured and estimated algebraic variables, i.e.  $\Delta \mathbf{a}_k = \tilde{\mathbf{a}}_k - \mathbf{a}_k^-$ . Using this definition with (5-15), the power flow equations (5-8) can be re-written as:

$$0 = \mathbf{g}(\mathbf{x}_k^- + \mathbf{e}_k^-, \mathbf{a}_k^- + \Delta \mathbf{a}_k + \mathbf{v}_k). \quad (5-37)$$

Expanding (5-37) about  $(\mathbf{x}_k^-, \mathbf{a}_k^-)$  using Taylor series results in the following equation:

$$\begin{aligned} \mathbf{G}_{\mathbf{x},k} &= \left. \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right|_{\substack{\mathbf{x}=\mathbf{x}_k^- \\ \mathbf{a}=\mathbf{a}_k^-}}, \\ \mathbf{G}_{\mathbf{a},k} &= \left. \frac{\partial \mathbf{g}}{\partial \mathbf{a}} \right|_{\substack{\mathbf{x}=\mathbf{x}_k^- \\ \mathbf{a}=\mathbf{a}_k^-}}, \end{aligned} \quad (5-38)$$

$$0 = \mathbf{g}(\mathbf{x}_k^-, \mathbf{a}_k^-) + \mathbf{G}_{\mathbf{x},k} \mathbf{e}_k^- + \mathbf{G}_{\mathbf{a},k} (\Delta \mathbf{a}_k + \mathbf{v}_k) + H.O.T.,$$

where,  $H.O.T$  represents higher order terms;  $\mathbf{G}_{\mathbf{x},k}$  and  $\mathbf{G}_{\mathbf{a},k}$  represent partial derivatives of  $\mathbf{g}$  with respect to states and algebraic variables, respectively. By neglecting higher order terms and considering (5-30),  $\Delta \mathbf{a}_k$  can be approximated by:

$$\Delta \mathbf{a}_k \approx -(\mathbf{G}_{\mathbf{a},k}^{-1} \mathbf{G}_{\mathbf{x},k} \mathbf{e}_k^- + \mathbf{v}_k). \quad (5-39)$$

Substituting (5-39) into (5-36), an updated estimate of states is approximated as:

$$\mathbf{x}_k^+ \approx \mathbf{x}_k^- - \mathbf{K}(\mathbf{G}_{\mathbf{a},k}^{-1} \mathbf{G}_{\mathbf{x},k} \mathbf{e}_k^- + \mathbf{v}_k), \quad (5-40)$$

Subtracting both sides from  $\mathbf{x}_k$ , the updated estimation error is computed as follows:

$$\mathbf{e}_k^+ \approx (\mathbf{I} + \mathbf{K}_k \mathbf{G}_{\mathbf{a},k}^{-1} \mathbf{G}_{\mathbf{x},k}) \mathbf{e}_k^- + \mathbf{K}_k \mathbf{v}_k. \quad (5-41)$$

This equation describes the updated estimation error in terms of predicted estimation error and the measurement noise. Observing that  $\mathbf{e}_k^-$  and  $\mathbf{v}_k$  are two independent random variables,  $\mathbf{P}_k^+$  is computed as follows:

$$\mathbf{P}_k^+ = E[\mathbf{e}_k^+ (\mathbf{e}_k^+)^T] = (\mathbf{I} + \mathbf{K}_k \mathbf{G}_{\mathbf{a},k}^{-1} \mathbf{G}_{\mathbf{x},k}) \mathbf{P}_k^- (\mathbf{I} + \mathbf{K}_k \mathbf{G}_{\mathbf{a},k}^{-1} \mathbf{G}_{\mathbf{x},k})^T + \mathbf{K}_k \mathbf{R} \mathbf{K}_k^T. \quad (5-42)$$

In (5-42), Kalman gain is the only unknown, which is computed by minimizing trace of  $\mathbf{P}_k^+$ .

The minimization process is omitted here, more information can be found in [111]. Overall, minimizing trace of  $\mathbf{P}_k^+$  results in the following pair of equations:

$$\begin{aligned} \mathbf{S} &= \mathbf{G}_{\mathbf{a},k}^{-1} \mathbf{G}_{\mathbf{x},k} \mathbf{P}_k^- \mathbf{G}_{\mathbf{x},k}^T (\mathbf{G}_{\mathbf{a},k}^{-1})^T + \mathbf{R}, \\ \mathbf{K}_k &= -\mathbf{P}_k^- \mathbf{G}_{\mathbf{x},k}^T (\mathbf{G}_{\mathbf{a},k}^{-1})^T \mathbf{S}^{-1}. \end{aligned} \quad (5-43)$$

Equation (5-43) shows that the formulation of Kalman gain in a system with implicit observation.

Comparing (5-23) and (5-43), it can be seen that observation matrix  $\mathbf{H}_k$  is replaced by  $-\mathbf{G}_{\mathbf{a},k}^{-1} \mathbf{G}_{\mathbf{x},k}$ . Considering the formulation of  $\mathbf{G}_{\mathbf{a},k}$  and  $\mathbf{G}_{\mathbf{x},k}$  in (5-38), the term  $-\mathbf{G}_{\mathbf{a},k}^{-1} \mathbf{G}_{\mathbf{x},k}$  represents linear sensitivity of algebraic variables to the states, and therefore, it is herein called the sensitivity factors.

Once Kalman gain is computed using (5-43), the states can be updated as:

$$\mathbf{x}_k^+ = \mathbf{x}_k^- + \mathbf{K}_k (\tilde{\mathbf{a}}_k - \hat{\mathbf{a}}_k), \quad (5-44)$$

$$\mathbf{P}_k^+ = \left( \mathbf{I} + \mathbf{K}_k \mathbf{G}_{\mathbf{a},k}^{-1} \mathbf{G}_{\mathbf{x},k} \right) \mathbf{P}_k^-. \quad (5-45)$$

Overall, (5-43)-(5-45) describe the update stage of the EKF in a global DSE scheme in the TSA.

#### 5.2.4 Update stage in the TSA – sequential approach

In order to compute the Kalman gain in (5-43), matrix  $\mathbf{S}$  needs to be inverted in each step. In power systems with large number of measurements, the size of this matrix will be large, making it computationally expensive to construct and invert  $\mathbf{S}$  for each set of PMU measurements. However, this matrix inversion can be effectively avoided by sequential processing of the measurements. The pseudo code representation of this approach is shown in Algorithm 1, wherein  $\mathbf{p}_{m,k}^-$  represents  $m$  th row of  $\mathbf{P}_k^-$ ;  $\mathbf{g}_{m,k}$  represents  $m$  th row of  $\mathbf{G}_{\mathbf{a},k}^{-1} \mathbf{G}_{\mathbf{x},k}$ ;  $r_m$  represents  $m$  th diagonal element in matrix  $\mathbf{R}$ ;  $\tilde{a}_{k,m}$  and  $a_{k,m}^-$  represent  $m$  th element of  $\tilde{\mathbf{a}}_k$  and  $\mathbf{a}_k^-$ , respectively; and  $N_m$  represents number of measured quantities.

In Algorithm 1, the measurements are processed one by one whereas in simultaneous approach presented in Section 5.2.3, all the measurements are processed simultaneously. In Algorithm 1, the Kalman gain matrix  $\mathbf{K}$  is replaced by a series of vectors  $\mathbf{k}_{m,k}$ , which corresponds to the Kalman gain computed for each measurement. These vectors are used

subsequently to update  $\mathbf{x}_k^+$  and  $\mathbf{P}_k^+$ . As can be seen, this approach replaces matrix inversion  $\mathbf{S}^{-1}$  by a series of scalar inversions, which can help reducing the overall computational cost.

---

**Algorithm 1.** Pseudo-code representation of update step in the EKF with sequential processing of measurements.

---

**for**  $m = 1$  to  $N_m$

$$s_m = \mathbf{g}_{m,k} \mathbf{P}_k^- \mathbf{g}_{m,k}^T + r_m$$

$$\mathbf{k}_{m,k} = -\mathbf{P}_k^- \mathbf{g}_{m,k} (s_m)^{-1}$$

$$\mathbf{x}_k^+ = \mathbf{x}_k^- + \mathbf{k}_{m,k} (\tilde{a}_{k,m} - a_{k,m}^-)$$

$$\mathbf{P}_k^+ = (\mathbf{I} + \mathbf{k}_{m,k} \mathbf{g}_{m,k}) \mathbf{P}_k^-$$

**end for**

---

## 5.3 Implementation

### 5.3.1 Avoid inverting matrix of sensitivity factors

In both methods presented in sections 5.2.3 and 5.2.4, the inverse of matrix  $\mathbf{G}_{\mathbf{a},k}$  needs to be computed. This inversion is a consequence of existence of implicit observation in transient stability problem, which is an integral part of the EKF with implicit observation. The size of  $\mathbf{G}_{\mathbf{a},k}$  is typically large, and computation of  $\mathbf{G}_{\mathbf{a},k}^{-1} \mathbf{G}_{\mathbf{x},k}$  for each set of PMU measurements is a computationally expensive task. However, since the term  $\mathbf{G}_{\mathbf{a},k}^{-1} \mathbf{G}_{\mathbf{x},k}$  represents the linear sensitivity of algebraic variables to states of the system, it does not

change significantly in a given contingency. Therefore, unless a major disturbance and a change in network configuration happen, it is not necessary to update the sensitivity factors. Based on this observation, we propose to compute the sensitivity factors at the beginning of simulation and update them only when the configuration of network changes.

### 5.3.2 Summary

The required steps are shown in Algorithm 2. As the first step, the filter is initialized using the initial guess for state variables and the error covariance. The initial guess of state variables is used to compute the initial guess for algebraic variables ( $\mathbf{a}_0^+$ ) and the sensitivity factors. Then, for each set of PMU measurements, the filter performs prediction and update stages to estimate the state variables. In prediction stage, (5-29)-(5-35) are used to predict the state and algebraic variables at the next step. In the update stage, the sensitivity factors can be updated for each set of measurements, or they might be kept constant until the configuration of network changes. Also, (5-43)-(5-45) can be used to simultaneously process the measurements and perform the update stage. Alternatively, one might choose to use Algorithm 1 to sequentially process the measurements in the update stage to achieve a better computational performance.

---

**Algorithm 2.** The required steps in the proposed dynamic state estimator.

---

**Initialization**

Initialize the filter at  $k = 0$  by choosing  $\mathbf{x}_0^+$  and  $\mathbf{P}_0^+$ .

Solve  $\mathbf{g}(\mathbf{x}_0^+, \hat{\mathbf{a}}_0^+) = 0$  to find initial guess of algebraic variables.

Compute the sensitivity factors  $\mathbf{G}_{\mathbf{a},0}^{-1} \mathbf{G}_{\mathbf{x},0}$

for  $k = 1, 2, \dots$  perform the following steps

**Prediction**

Use (5-29) to find  $\mathbf{x}_k^-$ .

Use (5-30) to find  $\mathbf{a}_k^-$ .

Use (5-31), (5-33), and (5-34) to find  $\mathbf{F}_k$ ,  $\mathbf{V}_k$ , and  $\mathbf{V}_{k-1}$ .

Use (5-35) to find  $\mathbf{P}_k^-$ .

**Update**

**IF** update of sensitivity factors should be skipped

Update sensitivity factors only if a major disturbance has happened.

**ELSE IF** sensitivity factors are updated for each measurement

Update sensitivity factors by computing  $\mathbf{G}_{\mathbf{a},k}^{-1} \mathbf{G}_{\mathbf{x},k}$ .

**END IF**

**IF** simultaneous approach is used:

Use (5-43) to find  $\mathbf{K}_k$ .

Use (5-44) to find  $\mathbf{x}_k^+$ .

Use (5-45) to find  $\mathbf{P}_k^+$ .

**ELSE IF** sequential approach is used:

Use Algorithm 1 to compute  $\mathbf{x}_k^+$  and  $\mathbf{P}_k^+$ .

**END IF**

**end for**

---

## 5.4 Simulation Studies

The IEEE test system described in section 2.1.6 is used here to evaluate the effectiveness of the proposed DSE technique. As stated in IEEE standard C37.118-2005 [112], the accumulation of magnitude, angle, and timing errors of PMU measurements should be less than 1%. Therefore, 1% of white Gaussian noise is added to the measured voltage phasors. Moreover, the DSATools is used here to generate the reference TDS solution.

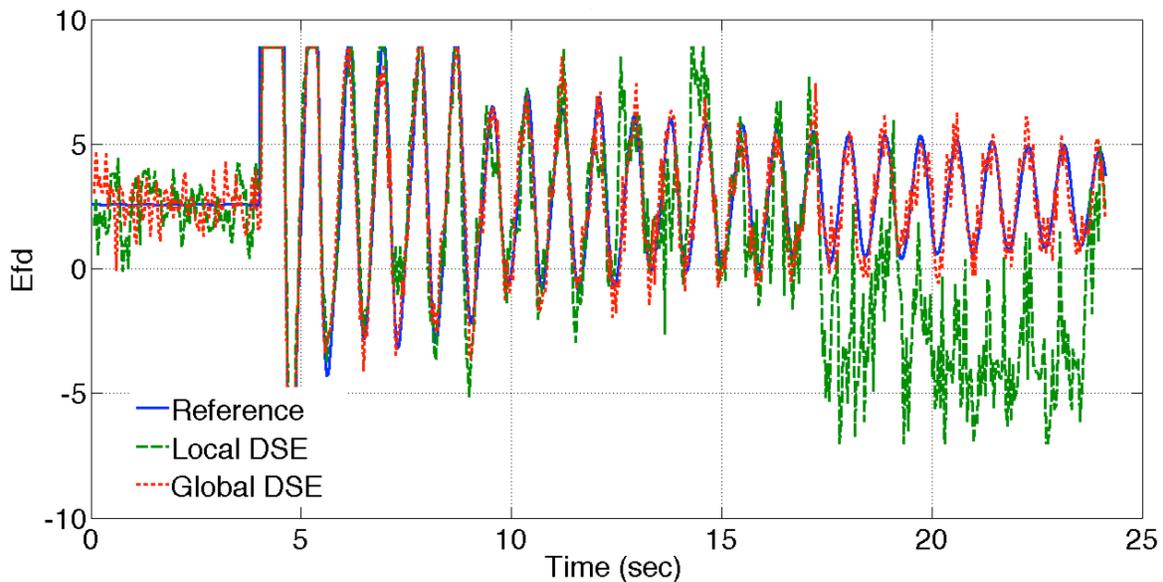
In order to simulate a disturbance in the system, a three-phase fault at bus #7 is applied at  $t = 4s$ . The fault is cleared in 8 cycles by opening the transmission line between buses #7 and #6. This scenario will be used throughout this section. To quantitatively evaluate the performance of different DSE schemes, the 2-norm (cumulative) relative error defined in (4-15) is used here.

### 5.4.1 Comparing accuracy of local DSE against global DSE

As the first study, the accuracy of global DSE is compared against local DSE scheme. In the global DSE, Algorithm 2 is used to estimate states of the system. In the local DSE, similar steps are carried out, except system model (5-5)-(5-6) is replaced by generator model (5-3)-(5-4). In this case, it is assumed that a PMU is installed at bus #111 and it measures the vector  $\mathbf{a}_i$  that corresponds to the generator located at this bus. The measurements are then used in Algorithm 2 to estimate the states of this generator. In both schemes, the Runge-Kutta method described in Section 5.2.2 is considered for discretizing differential equations; and the simultaneous approach described in Section 5.2.3 is used for processing

the PMU measurements. The sensitivity factors are updated in each measurement step. Also, for the purpose of this study, the PMU sampling rate is assumed to be 25 samples/sec.

The state estimation results are shown in Figure 5-1. As this figure shows, the local DSE has started to noticeably deviate from the reference solution at about  $t = 12s$ , and the estimation completely diverged after about  $t = 17s$ . In contrast, the global DSE is able to provide good estimation throughout the simulation. The calculated estimation error for the global and local DSE is 10.89% and 70.54%, respectively. These results confirm that the global DSE is able to provide an acceptable estimation of generator's field voltage.



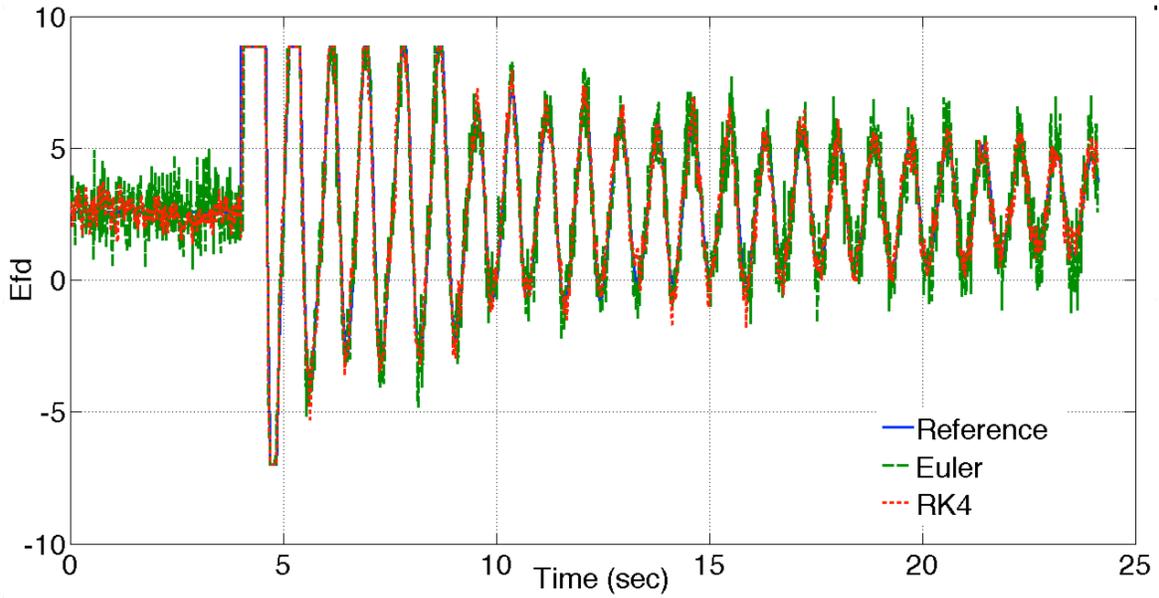
**Figure 5-1. Comparing global versus local dynamic state estimation.**

### 5.4.2 Effect of discretization method

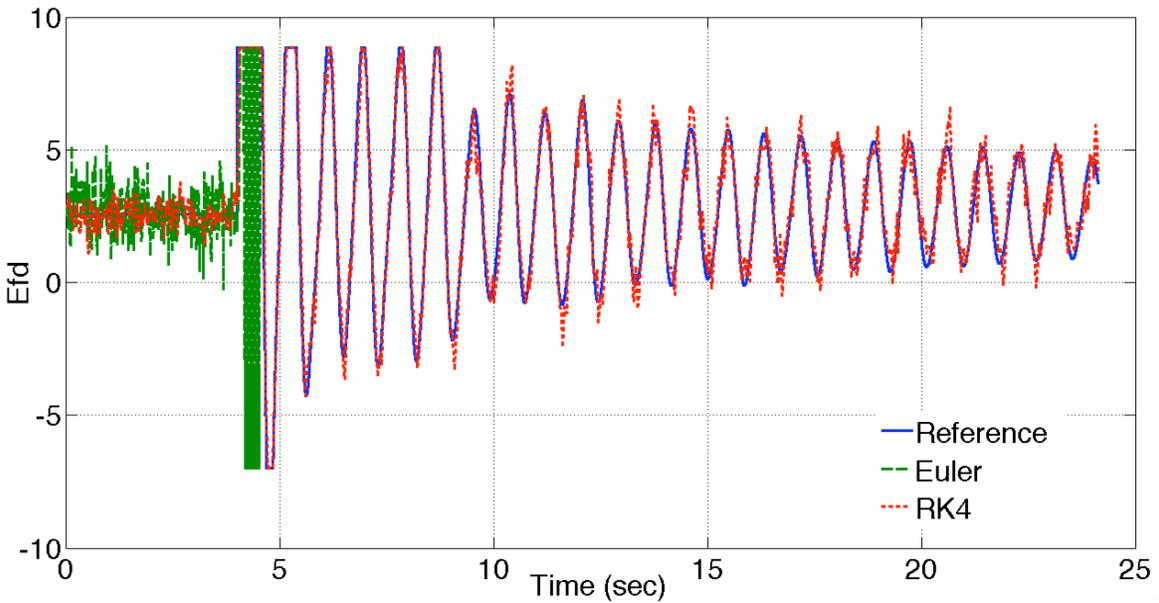
In this section, the effectiveness of using forth-order Runge-Kutta vs. Euler (which has been used by [60]-[65]) discretization method is investigated. In both methods, the global DSE scheme is implemented according to Algorithm 2, and simultaneous approach is used to

process the PMU measurements. The sensitivity factors are updated in each measurement step. First, it is assumed that the PMUs send data at a rate of 60 samples/sec. Figure 5-2 shows the estimated output voltage of the excitation system when the Euler and Runge-Kutta methods are used in the prediction stage. As Figure 5-2 shows, both integration techniques are able to estimate the field voltage throughout the simulation. The estimation error and CPU timings for these methods are summarized in Table 5-2. The first column in this table shows that the Runge-Kutta is almost 3 times more accurate. The second column in this table represents the total time required to complete the simulation, wherein there is a slight difference between Euler and Runge-Kutta methods. However, note that this quantity also includes the time that algorithm spends to perform other operations such as solving (5-30), computing kalman gain, and computing sensitivity factors. Therefore, in order to better compare computational performance of Euler and Runge-Kutta, it is more appropriate to evaluate the time that algorithm spends to compute  $\mathbf{x}_k^-$  and  $\mathbf{P}_k^-$ , since the computational cost associated with calculation of these quantities is directly related to the choice of discretization method. These results are shown in the third column of Table 5-2, wherein it is demonstrated that the Euler is almost 5 times faster than Runge-Kutta.

Next, the PMU sampling rate is reduced to 30 samples/sec, which is the practical sampling rate for some utilities such as BC Hydro. The results are shown in Figure 5-3, where it is shown that the DSE with Runge-Kutta method is able to provide a good estimation of field voltage, while the DSE with Euler method has diverged very quickly once the fault is cleared. In this case, the estimation error for the DSE with RK4 method is  $\varepsilon_{RK4} = 2.48\%$ , which is slightly larger than it was in previous case with PMU sampling of 60 samples/sec.



**Figure 5-2 Dynamic state estimation provided by Euler and RK4 methods for PMU sampling rate at 60 samples/sec.**



**Figure 5-3 Dynamic state estimation provided by Euler and RK4 methods for PMU sampling rate at 30 samples/sec.**

**Table 5-2 Evaluation of Error and CPU Timing for Euler and Runge-Kutta Methods**

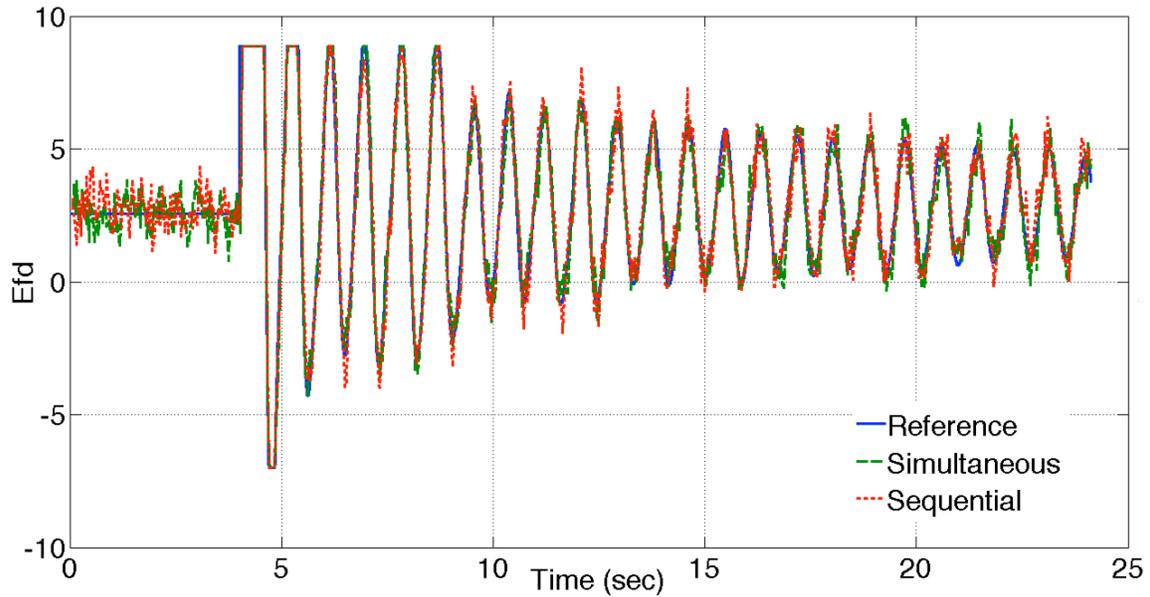
Discretization Technique	Error	CPU Time (Total Simulation Time)	CPU Time (Only Computation of $\mathbf{x}_k^-$ and $\mathbf{P}_k^-$ )
Euler	5.3%	53s	3s
Runge-Kutta	1.75%	65s	15s

### 5.4.3 Simultaneous vs. sequential processing of measurements

The measured quantities can be processed sequentially to avoid inverting matrix  $\mathbf{S}$  as discussed in Section 5.2.3. In this study, the DSE problem is solved using simultaneous and sequential processing of measurements. In the first case, which corresponds to simultaneous processing of the measured data, (5-43)-(5-45) are used in the update stage. In this case, the matrix  $\mathbf{S}$  is formed and inverted for each set of PMU measurements. In the second case, Algorithm 1 is used to perform the update stage. In both cases, the prediction stage is solved using Runge-Kutta method as described in Algorithm 2. The PMU sampling rate of 30 samples/sec is assumed.

The state estimation results for these cases are presented in Figure 5-4, wherein green and red lines represent the results provided by simultaneous and sequential processing of measured quantities, respectively. As this figure suggests, the sequential processing of measured data provides almost the same accuracy as the simultaneous approach. The results can be further compared by evaluating the estimation error and CPU timing for both cases, as summarized in Table 5-3. These results verify that while simultaneous processing of data provides slightly more accurate estimation, the accuracy of both approaches are

very close. Moreover, the results show that 25% improvement in the computational performance can be achieved using sequential approach.



**Figure 5-4 Dynamic state estimation results derived by simultaneous and sequential processing of measured data.**

**Table 5-3 Comparing Performance of Simultaneous and Sequential Approach for Processing Measurements**

Processing Measurements	Error	CPU Time
Simultaneous	2.75%	32s
Sequential	2.90%	24s

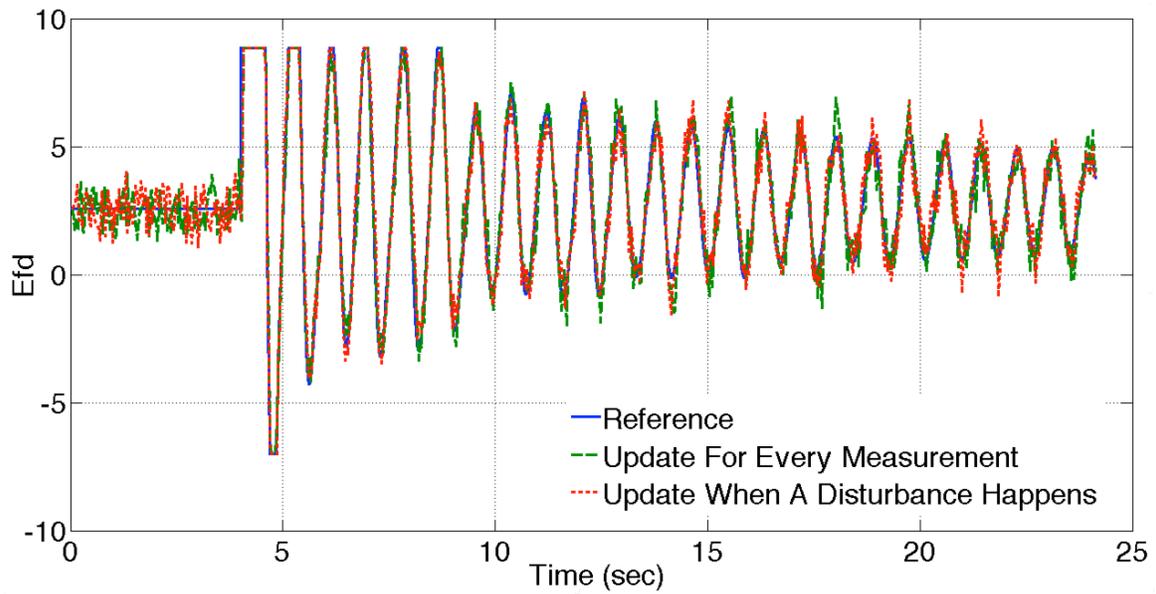
#### **5.4.4 Effect of updating sensitivity factors**

It was mentioned in Section 5.3.1 that it may not be necessary to frequently update the sensitivity factors, and that it may be more computationally efficient if the sensitivity factors are updated only when the system configuration changes. The accuracy of this strategy will be investigated here. For this purpose, the scenario described at the beginning

of this section is considered. In this scenario, the states of generator at bus #111 are largely disturbed and the nonlinear terms are highly excited. Therefore, it can represent a system wherein the sensitivity factors experience a wide change.

It is assumed that DSE with Runge-Kutta method in prediction step is used; the measurements are processed sequentially; and the PMUs send measured quantities at a rate of 30 samples/sec. The results of this study are shown in Figure 5-5, where the blue line represents the reference solution, the green line represents the DSE results when the sensitivity factors are updated for every set of PMU measurements, and the red line represents the case wherein the sensitivity factors are updated only at the beginning of simulation when the fault happens and when the fault is cleared. As Figure 5-5 shows, updating sensitivity factors does not have a significant effect on the estimated field voltage.

The accuracy and performance of the proposed strategy is further summarized in Table 5-4, wherein the 2-norm error and CPU timings are presented. These results verify that by skipping the update of sensitivity factors further computational speed-up may be achieved without significant sacrifice in accuracy.



**Figure 5-5 Effect of updating sensitivity factors on accuracy of DSE results.**

**Table 5-4 Evaluating Accuracy and Performance of DSE When the Update of Sensitivity Factors is Skipped**

Updating Sensitivity Factors	Error	CPU Time
At each measurement step	2.76%	24s
Only when the network configuration changes	3.05%	18s

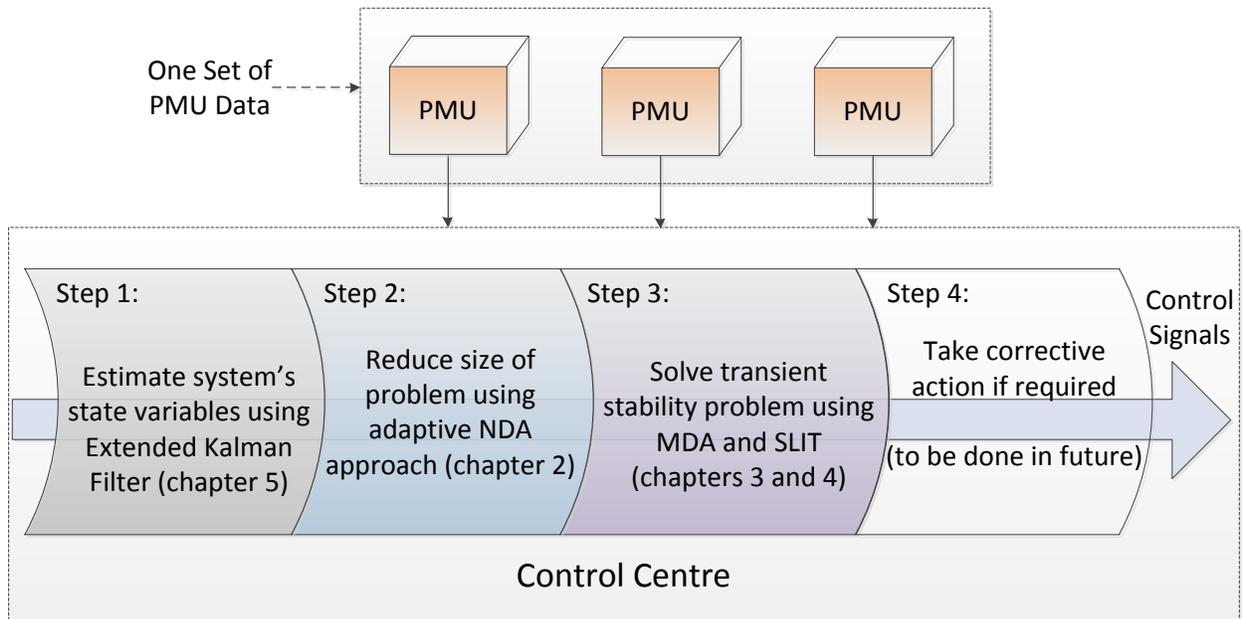
# **CHAPTER 6: SUMMARY OF CONTRIBUTIONS AND FUTURE WORKS**

## **6.1 Conclusions and Contributions**

The main objective of this thesis was to investigate challenges that exist in a TSM package and develop new algorithms to address these challenges. Based on the structure of a TSM tool shown in Figure 1-2 and according to the discussion presented in section 1.4, several contributions were made to expedite the TDS approach and take advantage of PMU measurements in monitoring applications. Figure 6-1 demonstrates how the newly developed methods are integrated into a TSM tool, where it is shown that the contributions made by this research address the challenges existing in steps 1 to 3 of a TSM package.

To summarize, a novel NDA algorithm was proposed in chapter 2 to aggregate coherent generators and buses, which helps improve accuracy of reduced system in dynamic equivalencing. The proposed NDA method is able to identify and remove the errors in coherency identification step. Also, to further expedite the simultaneous solution of the nonlinear DAEs appearing in the TSA time-domain simulation, two new methods, namely MDA and SLIT, were presented in chapters 3 and 4. Both MDA and SLIT techniques decompose the original nonlinear DAEs into a series of linear systems, which can be solved concurrently. However, the SLIT approach is easier to parallelize among many processors and does not need hessian matrix, which represents additional practical advantage. Finally, a general framework for using PMUs' measured quantities in transient stability monitoring

was presented in chapter 5. This framework aims at improving accuracy of state estimation in transient stability studies with practical PMU's sampling rate. More specifically, with respect to the initial objectives of this research, the contributions of the thesis can be summarized as follows:



**Figure 6-1 Integrating the methods developed in this thesis into a transient stability monitoring tool.**

## Objective 1

In Section 2.1, a new network-dependent aggregation algorithm of coherent generators for constructing the reduced-order transient stability problem was proposed. In the proposed NDA method, the weight of each generator is determined based on its contribution to the mismatch between full- and reduced-order systems – which is different from traditional methods that assign a relative weight simply based on the generator inertia and/or power level. The proposed NDA approach keeps the dynamics of the generators which can cause

larger errors in the reduced-order system. Moreover, based on trapezoidal integration rule, a new criterion is defined to evaluate the accuracy of the reduced-order system. In the proposed approach, while integrating the reduced-order system, the accuracy of the results is continuously monitored. If an error in coherency identification is detected, corresponding generator will be isolated from its group and a new reduced-order system will be constructed. Computer studies show that the proposed NDA is more accurate and robust compared to the conventional IAA algorithm.

In Section 2.2, a new method to identify coherent buses in transmission system was proposed. Using the algebraic relationship between generation and transmission systems, the sufficient condition for coherency between each two buses was derived. The presented simulation results demonstrate that the proposed method is able to identify almost all tightly coherent buses in the IEEE 50-gen 95-bus benchmark system. The results are valid for more than 10 seconds, which makes it possible to analyze multi-swing stability problems using reduced system.

## **Objective 2**

In Chapter 3, we presented a multi-decomposition approach for solving transient stability problem. The proposed MDA decomposes the nonlinear DAEs of the transient stability problem into three linear subsystems, which are subsequently solved to approximate the solution. The linear subsystems are updated as necessary to ensure that the approximation remains accurate within the user-specified tolerance. The accuracy of the MDA is verified against the conventional time-domain simulation using the DSATools from Powertech Labs

Inc. It was also shown that sequential MDA is as accurate and slightly faster than VDHN. Moreover, contrary to Full Newton and VDHN, the MDA is easily parallelizable into three processors, which represents a significant potential for extending this method to larger systems.

Chapter 4 proposed successive linearization integration technique (SLIT), which is a new integration technique for solving transient stability problems in power systems. In the proposed method, a series of linear systems are constructed by successively linearizing the original nonlinear DAEs. The sum of the solutions of linear systems approximates the trajectory of actual solution. Since different linear systems can be solved in parallel, the proposed method can be readily parallelizable and implemented over several processors. The DSATools was used to verify the SLIT method. It was demonstrated that SLIT is faster than the VDHN and the MDA methods without compromising accuracy. Moreover, it was demonstrated that the parallel version of the SLIT is more scalable compared to the MDA.

### **Objective 3**

Chapter 5 presents a general framework for using extended Kalman filter (EKF) in transient stability studies. In the proposed technique, a fourth-order Runge-Kutta method is used to discretize differential equations and find a prediction of states. Also, it was demonstrated that the conventional EKF cannot be applied to the formulation of global DSE since the measured quantities are implicitly related to the states in transient stability problem. The conventional formulation of the Kalman gain was modified and extended to the systems with implicit observation, such as the transient stability problem. The

performance of the proposed framework was evaluated using IEEE 145 bus test system and it was demonstrated that the proposed technique can provide an accurate approximation of state variables even when the system is largely disturbed and the PMU sampling rate is fairly low ( $\sim 25$  samples/sec).

## **6.2 Potential Impacts of Contributions**

The proposed NDA method improves accuracy and robustness of dynamic equivalencing packages. With the rapid growth of size of power systems, these features are becoming increasingly important in evaluating performance of reduced-order systems. Moreover, the NDA is applicable to online dynamic reduction techniques [72],[73], wherein real-time PMU measurements are used to create an equivalent machine. In this applications, the NDA can help creating more accurate equivalents and better monitoring transfer paths [81].

It is envisioned that the proposed MDA and SLIT will find wide application in commercial transient stability solvers. In particular, the SLIT method requires the same vectors/matrices as used in an implicit integration technique (i.e. mismatch vector and Jacobian matrix), which should make it easier for the software developers to implement this method in programs such as DSATools. The proposed fast and accurate integration techniques allow user to model larger systems, analyze larger number of contingencies, and take advantage of multi-core processing units.

As mentioned in section 1.1, in order to make TSM computationally feasible, available transient stability solvers need to become at least 8 times faster. Currently, BC Hydro is

decreasing size of the WECC system from 15,000 buses to 5,000 buses in dynamic studies, which helps improving computational performance by at least 3 times. Moreover, the results presented in section 4.5.3 demonstrated that an extra 3-time speed-up can be achieved by parallelizing solution in the SLIT method. Therefore, reducing size of problem using dynamic reduction technique combined with parallelizing solution in the SLIT can exceed the 8-time speed-up criterion, which further verifies that the proposed SLIT is a promising approach for the TSM applications.

Also, the proposed dynamic state estimation scheme makes it possible to use PMUs with low measurement rate (25~30 samples/sec) in a TSM application. Since the proposed methodology is not limited to any specific model or assumptions, it can be used by operator to estimate states of all generators as well as other dynamic devices.

### **6.3 Future Work**

As mentioned in Section 2.1.5, the algorithm proposed for finding coherency identification errors might fail when there are numerous errors in coherency identification step. In practice, the operator does not know how many errors exist and therefore, this algorithm needs to be improved to become more robust against large number of misplacements.

The MDA and SLIT methods are capable of parallelizing solution of transient stability problem by decomposing original nonlinear DAEs into a series of linear sub-problems. In order to increase number of processors, it is possible to take advantage of available parallel techniques, such as Gauss-Seidel [49], MATE [50]-[54], SOR [56], Gauss-Jacobi [58]-[59],

etc., inside the MDA and SLIT to perform more decompositions on each sub-problem. The newly decomposed sub-problems can then be shared among more processors to further improve computational performance of transient stability solver.

For more accurate dynamic state estimation, the effect of a malfunction PMU should be considered. In this thesis, it was assumed that all PMUs are working properly. However, in reality, a malfunction PMU might send noisy or invalid data. The dynamic state estimation algorithm should be able to detect and eliminate these wrong measurements. Also, in Chapter 6, it was assumed that there are enough measurements available to make the whole system observable. While this assumption might become true in near future, there is still significant difficulty in having enough PMU measurements considering economic challenges associated with installing PMUs and communication infrastructures. Therefore, it is required to extend the method described in Chapter 6 to the cases where unobservable areas exist in the system.

As the last step in the TSM, an appropriate corrective action is required if system is found to be unstable. The corrective action is typically performed by either reducing generated power or adding breaking resistors. In both cases, the amount of generation shed from the system should be minimized. Choosing appropriate location for performing the corrective action can significantly help reducing the amount of generation shed. Our initial analysis shows that the solutions of high-order linear systems inside the SLIT may be a good starting point to determine the optimal places amount of required energy shedding.

Further analysis is required to investigate the possibility of using the linearized subsystems of either MDA or SLIT for identifying the appropriate remedial action.

Finally, developing a prototype is an essential step toward realizing an industrial-grade transient stability monitoring package. This prototype should be able to read PMU measurements and perform a transient stability simulation on a large-scale system, e.g. WECC.

## REFERENCES

- [1] "Dynamic Security Assessment Software," Power Tech Labs Inc., Surrey, BC, Canada, 2015. [Online]. Available: [http:// www.dsatools.com](http://www.dsatools.com) [Feb. 1, 2015].
- [2] "Siemens Power Technologies International," Siemens Corporation, Germany, 2015. [Online]. Available: <http://w3.siemens.com/smartgrid/global/en/products-systems-solutions/software-solutions/planning-data-management-software/planning-simulation/pages/pss-e.aspx> [Feb. 1, 2015].
- [3] "Power World Simulator," Power World Corporation, Champaign, Illinois, USA, 2015. [Online]. Available: <http://www.powerworld.com> [Feb. 1, 2015].
- [4] "Positive Sequence Loadflow Software," General Electric Energy, New York, USA, 2015. [online] Available: <http://www.geenergyconsulting.com/practice-area/software-products/pslf> [Feb. 1, 2015].
- [5] L. Wang, "Applications of High Performance Computing for Dynamic Security Assessment of Power Systems," *IEEE General Meeting of Power Engineering Society*, July, 2013. [Online]. [http://ewh.ieee.org/cmte/pes/etcc/L\\_Wang\\_HPC\\_Applications\\_For\\_Dynamic\\_Security\\_Assessment.pdf](http://ewh.ieee.org/cmte/pes/etcc/L_Wang_HPC_Applications_For_Dynamic_Security_Assessment.pdf) [Feb. 1, 2015].
- [6] T. Athay, R. Podmore, and S. Virmani, "A Practical Method for the Direct Analysis of Transient Stability," *IEEE Trans. Power Apparatus and Systems*, vol. PAS-98, no. 2, pp. 573-584, March 1979.
- [7] N. Kakimoto, Y. Ohnogi, H. Matsuda, and H. Shibuya, "Transient Stability Analysis of Large-Scale Power System by Lyapunov's Direct Method," *IEEE Trans. Power Apparatus and Systems*, vol. PAS-103, no. 1, pp. 160-167, Jan. 1984.
- [8] H. D. Chiang, F. F. Wu, and P. P. Varaiya, "A BCU method for direct analysis of power system transient stability," *IEEE Trans. Power Systems*, vol. 9, no. 3, pp. 1194-1208, Aug 1994.

- [9] H. D. Chiang, and C. C. Chu, "Theoretical foundation of the BCU method for direct stability analysis of network-reduction power system. Models with small transfer conductances," *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, vol. 42, no. 5, pp. 252-265, May 1995.
- [10] H. D. Chiang, C. S. Wang, and H. Li; "Development of BCU classifiers for on-line dynamic contingency screening of electric power systems," *IEEE Trans. Power Systems*, vol. 14, no. 2, pp. 660-666, May 1999.
- [11] M. Pavella, D. Ernst, and D. Ruiz-Vega, *Transient Stability of Power Systems: a Unified Approach to Assessment and Control*, 2000 :Kluwer Academic Publishers .
- [12] Y. Xue, T. Van Cutsem, and M. Ribbens-Pavella, "A simple direct method for fast transient stability assessment of large power systems," *IEEE Trans. Power Systems*, vol. 3, no. 2, pp. 400-412, May 1988.
- [13] M. Yin, C. Y. Chung, K. P. Wong, Y. Xue, and Y. Zou, "An Improved Iterative Method for Assessment of Multi-Swing Transient Stability Limit," *IEEE Trans. Power Systems*, vol. 26, no. 4, pp. 2023-2030, Nov. 2011.
- [14] L. Wang and K. Morison, "Implementation of Online Security Assessment," *IEEE Power and Energy Magazine*, vol. 4, no. 5, pp. 46-59, Sept.-Oct. 2006.
- [15] Mladen Kezunovic, *Application of Time-Synchronized Measurements in Power System Transmission Network*, Springer, 2014, ch. 3.
- [16] K. Prabhashankar and W. Janischewsyj, "Digital Simulation of Multimachine Power Systems for Stability Studies," *IEEE Trans. Power Apparatus and Systems*, vol. PAS-87, no. 1, pp. 73-80, Jan. 1968.
- [17] P. W. Sauer and M. A. Pai, *Power System Dynamic and Stability*, Prentice. Hall, 1997, ch. 7.
- [18] H. W. Dommel and N. Sato, "Fast Transient Stability Solutions," *IEEE Trans. Power Apparatus and Systems*, vol. PAS-91, no. 4, pp. 1643-1650, July 1972.

- [19] P. Kundur, J. Paserba, V. Ajjarapu, G. Andersson, A. Bose, C. Canizares, N. Hatziaargyriou, D. Hill, A. Stankovic, C. Taylor, T. Van Cutsem, and V. Vittal, "Definition and classification of power system stability IEEE/CIGRE joint task force on stability terms and definitions," *IEEE Trans. Power Systems*, vol. 19, no. 3, pp. 1387-1401, Aug. 2004.
- [20] F. Alvarado, R. Betancourt, K. Clements, G. T. Heydt, G. Huang, M. Ilic, M. La Scala, M. A. Pai, C. Pottle, S. Talukdar, J. V. Ness, F. Wu, "Parallel processing in power systems computation," *IEEE Trans. Power Systems*, vol. 7, no. 2, pp. 629-638, May 1992.
- [21] U. Ascher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, Siam, 1998.
- [22] "Transient Security Assessment Tool," *User Manual*, pp. 21, Power Tech Labs Inc., Surrey, BC, Canada, 2012.
- [23] U. Ascher and C. Greif, *A First Course in Numerical Methods*, Siam, 2011, ch. 3.
- [24] J. S. Chai, N. Zhu, A. Bose, and d. J. Tylavsky, "Parallel Newton type methods for power system stability analysis using local and shared memory multiprocessors," *IEEE Trans. Power Systems*, vol. 6, no. 4, pp. 1539-1545, Nov 1991.
- [25] H. D. Chiang, *Direct Methods for Stability Analysis of Electric Power Systems*, A John Wiley & Sons, 2011, ch. 15.
- [26] N. Amjady and S. F. Majedi, "Transient Stability Prediction by a Hybrid Intelligent System," *IEEE Trans. Power Systems*, vol. 22, no. 3, pp. 1275-1283, Aug. 2007.
- [27] F. R. Gomez, A. D. Rajapakse, U. D. Annakkage, and I. T. Fernando, "Support Vector Machine-Based Algorithm for Post-Fault Transient Stability Status Prediction Using Synchronized Measurements," *IEEE Trans. Power Systems*, vol. 26, no. 3, pp. 1474-1483, Aug. 2011.

- [28] I. Kamwa, S. R. Samantaray, and G. Joos, "Development of Rule-Based Classifiers for Rapid Stability Assessment of Wide-Area Post-Disturbance Records," *IEEE Trans. Power Systems*, vol. 24, no. 1, pp. 258-270, Feb. 2009.
- [29] R. Podmore "Identification of Coherent Generators for Dynamic Equivalents," *IEEE Trans. Power Apparatus and Systems*, vol. PAS-97, no. 4, pp. 1344-1354, July 1978.
- [30] R. Nath, S. S. Lamba, and K. S. Prakasa Rao, "Coherency Based System Decomposition into Study and External Areas Using Weak Coupling," *IEEE Trans. Power Apparatus and Systems*, vol. PAS-104, no. 6, pp. 1443-1449, June 1985.
- [31] G. C. Verghese, I. J. Perez-Arriaga, and F. C. Schweppe, "Selective Modal Analysis With Applications to Electric Power Systems, Part II: The Dynamic Stability Problem," *IEEE Trans. Power Apparatus and Systems*, vol. PAS-101, no. 9, pp. 3126-3134, Sept. 1982.
- [32] J. H. Chow, J. Cullum, and R. A. Willoughby, "A Sparsity-Based Technique for Identifying Slow-Coherent Areas in Large Power Systems," *IEEE Trans. Power Apparatus and Systems*, vol. PAS-103, no. 3, pp. 463-473, Mar. 1984.
- [33] P. Kundur, *Power System Stability and Control*, McGraw-Hill, 1993, ch. 12.
- [34] H. Kim, G. Jang, and K. Song, K., "Dynamic reduction of the large-scale power systems using relation factor," *IEEE Trans. Power Systems*, vol.19, no.3, pp. 1696- 1699, Aug. 2004.
- [35] R. Podmore, "Development of Dynamic Equivalents for Transient Stability Studies," EPRI Report EL-456, 1977.
- [36] J. H. Chow, R. Galarza, P. Accari, and W. W. Price, "Inertial and slow coherency aggregation algorithms for power system dynamic model reduction," *IEEE Trans. Power Systems*, vol. 10, no. 2, pp. 680-685, May 1995.
- [37] M. L. Ourari, L. A. Dessaint, and V. Q. Do, "Dynamic equivalent modeling of large power systems using structure preservation technique," *IEEE Trans. on Power Systems*, vol. 21, no. 3, pp. 1284-1295, Aug. 2006.

- [38] H. Kim, G. Jang, and K. Song, "Dynamic reduction of the large-scale power systems using relation factor," *IEEE Trans. on Power Systems*, vol. 19, pp. 1696-1699, Aug. 2004.
- [39] G. N. Ramaswamy, L. Rouco, O. Fillatre, G. C. Verghese, P. Panciatici, B. C. Lesieutre, and D. Peltier, "Synchronic modal equivalencing (SME) for structure-preserving dynamic equivalents," *IEEE Trans. Power Systems*, vol. 11, no. 1, pp. 19-29, Feb 1996.
- [40] C. D. Vournas and J. C. Mantzaris, "Quasi-Steady-State modeling of interarea oscillations," *IEEE PowerTech Conference*, Bucharest, June 28-July 2, 2009.
- [41] C. D. Vournas, C.D. and J. C. Mantzaris, "Application of QSS Modeling to Stabilizer Design for Interarea Oscillations," *IEEE Trans. Power Systems*, vol. 25, no. 4, pp. 1910-1917, Nov. 2010.
- [42] J. C. Mantzaris, A. Metsiou, A., and C. D. Vournas, "Analysis of Interarea Oscillations Including Governor Effects and Stabilizer Design in South-Eastern Europe," *IEEE Trans. Power Systems*, vol. 28, no. 4, pp. 4948-4956, Nov. 2013.
- [43] S. B. Yusof, G. J. Rogers, and R. T. H. Alden, "Slow coherency based network partitioning including load buses," *IEEE Trans. On Power Systems*, vol. 8, no. 3, pp. 1375-1382, Aug. 1993.
- [44] W.W. Price, A. W. Hargrave, B. J. Hurysz, J. H. Chow, P. M. Hirsch, "Large-scale system testing of a power system dynamic equivalencing program," *IEEE Trans. on Power Systems*, vol. 13, no. 3, pp. 768-774, Aug 1998.
- [45] I. Kamwa, A. K. Pradhan, G. Joos, and S. R. Samantaray, "Fuzzy Partitioning of a Real Power System for Dynamic Vulnerability Assessment," *IEEE Trans. on Power Systems*, vol. 24, no. 3, pp. 1356-1365, Aug. 2009.
- [46] M. A. M. Ariff and B. C. Pal, "Coherency Identification in Interconnected Power System—An Independent Component Analysis Approach," *IEEE Trans. Power Systems*, vol. 28, no. 2, pp. 1747-1755, May 2013.

- [47] F. L. Alvarado, "Parallel Solution of Transient Problems by Trapezoidal Integration," *IEEE Trans. Power Apparatus and Systems*, vol. PAS-98, no.3, pp. 1080-1090, May 1979.
- [48] M. La Scala, M. Brucoli, F. Torelli, and M. Trovato, "A Gauss-Jacobi-Block-Newton method for parallel transient stability analysis [of power systems]," *IEEE Trans. Power Systems*, vol.5, no.4, pp.1168,1177, Nov 1990.
- [49] M. La Scala, R. Sbrizzai, and F. Torelli, "A pipelined-in-time parallel algorithm for transient stability analysis [power systems]," *IEEE Trans. Power Systems*, vol. 6, no. 2, pp. 715-722, May 1991.
- [50] M. Armstrong, J. R. Marti, L. R. Linares, P. Kundur, "Multilevel MATE for efficient simultaneous solution of control systems and nonlinearities in the OVNI simulator," *IEEE Trans. Power Systems*, vol. 21, no. 3, pp. 1250-1259, Aug. 2006.
- [51] F. A. Moreira, J. R. Marti, L. C. Zanetta, and L. R. Linares, "Multirate Simulations With Simultaneous-Solution Using Direct Integration Methods in a Partitioned Network Environment," *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 53, no. 12, pp. 2765-2778, Dec. 2006.
- [52] T. De Rybel, M. Tomim, A. Singh, and J. R. Marti, "An introduction to open-source linear algebra tools and parallelization for power system applications," *IEEE Electric Power Conference*, Canada, 6-7 Oct. 2008.
- [53] M. A. Tomim, "Parallel Computation of Large Power System Networks Using Multi-Area Thevenin Equivalentents," UBC PhD Thesis, 2009.
- [54] M. A. Tomim, J. R. Marti, T. De Rybel, L. Wang, and M. Yao, "MATE network tearing techniques for multiprocessor solution of large power system networks," *IEEE Power and Energy Society General Meeting*, 25-29 July 2010.
- [55] W. L. Hatcher, F. M. Brasch, and J. E. Van Ness, "A Feasibility Study for the Solution of Transient Stability Problems by Multiprocessor Structures," *IEEE Trans. Power Apparatus and Systems*, vol. 96, no. 6, pp. 1789-1797, Feb. 1977.

- [56] J. S. Chai, N. Zhu, A. Bose, and d. J. Tylavsky, "Parallel Newton type methods for power system stability analysis using local and shared memory multiprocessors," *IEEE Trans. Power Systems*, vol. 6, no. 4, pp. 1539-1545, Nov 1991.
- [57] J. Shu, W. Xue, and W. Zheng, "A Parallel Transient Stability Simulation for Power Systems," *IEEE Trans. Power Systems*, vol. 20, no. 4, pp. 1709-1717, Nov. 2005.
- [58] V. Jalili-Marandi, and V. Dinavahi, "Instantaneous Relaxation-Based Real-Time Transient Stability Simulation," *IEEE Trans. Power Systems*, vol. 24, no. 3, pp. 1327-1336, Aug. 2009.
- [59] V. Jalili-Marandi, Z. Zhou, and V. Dinavahi, "Large-Scale Transient Stability Simulation of Electrical Power Systems on Parallel GPUs," *IEEE Trans. Parallel and Distributed Systems*, vol. 23, no. 7, pp. 1255-1266, July 2012.
- [60] E. Ghahremani and I. Kamwa, "Dynamic State Estimation in Power System by Applying the Extended Kalman Filter With Unknown Inputs to Phasor Measurements," *IEEE Trans. Power Systems*, vol. 26, no. 4, pp. 2556-2566, Nov. 2011.
- [61] J. Zhang, G. Welch, G. Bishop, and Z. Huang, "A Two-Stage Kalman Filter Approach for Robust and Real-Time Power System State Estimation," *IEEE Trans. Sustainable Energy*, vol. 5, no. 2, pp. 629-636, April 2014.
- [62] E. Ghahremani and I. Kamwa, "Online State Estimation of a Synchronous Generator Using Unscented Kalman Filter From Phasor Measurements Units," *IEEE Trans. Energy Conversion*, vol. 26, no. 4, pp. 1099-1108, Dec. 2011.
- [63] S. Wang, W. Gao, and A. P. S. Meliopoulos, "An Alternative Method for Power System Dynamic State Estimation Based on Unscented Transform," *IEEE Trans. Power Systems*, vol. 27, no. 2, pp. 942-950, May 2012.
- [64] Y. Wehbe and L. Fan, "UKF based estimation of synchronous generator electromechanical parameters from phasor measurements," *North American Power Symposium (NAPS)*, 9-11 Sept. 2012.

- [65] N. Zhou, D. Meng, and S. Lu, "Estimation of the Dynamic States of Synchronous Machines Using an Extended Particle Filter," *IEEE Trans. Power Systems*, vol. 28, no. 4, pp. 4152-4161, Nov. 2013.
- [66] N. Zhou, Z. Huang, Y. Li, and G. Welch, "Local sequential ensemble Kalman filter for simultaneously tracking states and parameters," *North American Power Symposium (NAPS)*, 9-11 Sept. 2012.
- [67] Dynamic Reduction, Power Tech Labs Inc., Surrey, BC, Canada, 2012. [Online]. Available: <http://www.dsatools.com/downloads/DYNRED.pdf> [Feb. 1, 2015].
- [68] F. Ma, X. Luo, and V. Vittal, "Application of dynamic equivalencing in large-scale power systems," *Power and Energy Society General Meeting*, 24-29 July 2011.
- [69] F. Ma and V. Vittal, "Right-Sized Power System Dynamic Equivalents for Power System Operation," *IEEE Trans. Power Systems*, vol. 26, no. 4, pp. 1998-2005, Nov. 2011.
- [70] F. Ma, "Improved Coherency-based Dynamic Equivalent," PhD Dissertation, Arizona State University, November 2011. ch. 4.
- [71] A. Chakraborty and T. R. Khan, "Graph-theoretic model reduction of oscillation propagation in spatially distributed power system networks," *Annual Conference on Decision and Control (CDC)*, 10-13 Dec. 2012.
- [72] A. Chakraborty, J. H. Chow, and A. Salazar, "A measurement-based framework for dynamic equivalencing of large power systems using WAMS," *Innovative Smart Grid Technologies (ISGT)*, 19-21 Jan. 2010.
- [73] A. Chakraborty, J. H. Chow, A. Salazar, "A Measurement-Based Framework for Dynamic Equivalencing of Large Power Systems Using Wide-Area Phasor Measurements," *IEEE Trans. Smart Grid*, vol. 2, no. 1, pp. 68-81, Mar. 2011.
- [74] Lawrence Livermore National Laboratory, Livermore, California, USA. Website: <http://www.llnl.gov> [Feb. 1, 2015].

- [75] S. G. Smith, D. T. Van Zandt, B. Thomas, S. Mahmood, and C. S. Woodward, "HPC4Energy Final Report GE Energy," May 2013. [online] Available: <https://e-reports-ext.llnl.gov/pdf/770200.pdf> [Feb. 1, 2015].
- [76] A. Gomez-Exposito, A. Abur, A. de la Villa Jaen, and C. Gomez-Quiles, "A Multilevel State Estimation Paradigm for Smart Grids," *Proceedings of the IEEE*, vol. 99, no. 6, pp. 952-976, June 2011.
- [77] M. Gol, A. Abur, and F. Galvan, "Metrics for Success: Performance Metrics for Power System State Estimators and Measurement Designs," *IEEE Power and Energy Magazine*, vol. 10, no. 5, pp. 50-57, Sept.-Oct. 2012.
- [78] R. Emami and A. Abur, "External system line outage identification using phasor measurement units," *IEEE Trans. Power Systems*, vol. 28, no. 2, pp. 1035-1040, May 2013.
- [79] M. Gol and A. Abur, "A Robust PMU Based Three-Phase State Estimator Using Modal Decoupling," *IEEE Trans. Power Systems*, vol. 29, no. 5, pp. 2292-2299, Sept. 2014.
- [80] M. Göl and A. Abur, "LAV Based Robust State Estimation for Systems Measured by PMUs," *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 1808-1814, July 2014.
- [81] J. H. Chow, A. Chakraborty, M. Arcak, B. Bhargava, A. Salazar, "Synchronized Phasor Data Based Energy Function Analysis of Dominant Power Transfer Paths in Large Power Systems," *IEEE Trans. Power Systems*, vol. 22, no. 2, pp. 727-734, May 2007.
- [82] E. Farantatos, R. Huang, G. J. Cokkinides, A. P. S. Meliopoulos, "Real-time power system dynamic equivalencing to preserve system center of oscillations via PMU-based dynamic state estimator," *Power and Energy Society (PES) General Meeting*, 21-25 July 2013.
- [83] E. Farantatos, R. Huang, G. J. Cokkinides, A. P. S. Meliopoulos, "Physical parameters identification of synchronous generators by a dynamic state estimator," *Power and Energy Society (PES) General Meeting*, 21-25 July 2013.

- [84] A. P. S. Meliopoulos, "Distributed dynamic state estimator enables seamless DSA," *IEEE PES General Meeting / Conference & Exposition*, 27-31 July 2014.
- [85] Y. Wehbe, L. Fan, and Z. Miao, "Least squares based estimation of synchronous generator states and parameters with phasor measurement units," *North American Power Symposium (NAPS)*, 9-11 Sept. 2012.
- [86] L. Fan, Z. Miao, and Y. Wehbe, "Application of Dynamic State and Parameter Estimation Techniques on Real-World Data," *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 1133-1141, June 2013.
- [87] S. A. Soman, S.A. Khaparde, and S. Pandit, *Computational Methods for Large Sparse Power Systems Analysis: An Object Oriented Approach*, Springer, Nov. 2001, ch. 12.
- [88] R. J. Galarza, J. H. Chow, W. W. Price, A. W. Hargrave, and P. M. Hirsch, "Aggregation of exciter models for constructing power system dynamic equivalents," *IEEE Trans. Power Systems*, vol. 13, no. 3, pp. 782-788, Aug 1998.
- [89] U. Ascher and C. Greif, *A First Course in Numerical Methods*, Siam, 2011, ch. 6.
- [90] M. A. Pai, *Computer Techniques in Power System Analysis*, McGraw Hill, 2nd ed., pp. 63, 2006.
- [91] V. Vittal, "Transient stability test systems for direct stability methods," *IEEE Trans. Power Systems*, vol. 7, no. 1, pp. 37-43, Feb 1992.
- [92] IEEE Recommended Practice for Excitation System Models for Power System Stability Studies, IEEE Std. 421.5, Aug. 1992.
- [93] P. M. Anderson and A. A. Fouad, *Power System Control and Stability*. Ames, IA: Iowa State Univ. Press, 1977, ch. 7, page 162.
- [94] L. Wang, J. Jatskevich, W. Chengshan, and P. Li, "A Voltage-Behind-Reactance Induction Machine Model for the EMTP-Type Solution," *IEEE Trans. Power Systems*, vol. 23, no. 3, pp. 1226-1238, Aug. 2008.

- [95] F. Milano, *Power System Modeling and Scripting*. Heidelberg, Germany: Springer, 2010, pp. 85–86.
- [96] G. Hou and V. Vittal, “Trajectory Sensitivity Based Preventive Control of Voltage Instability Considering Load Uncertainties,” *IEEE Trans. Power Systems*, vol. 27, no. 4, pp. 2280-2288, Nov. 2012.
- [97] W. Rugh, *Nonlinear System Theory: The Volterra-Wiener Approach*. Baltimore, MD: Johns Hopkins Univ. Press, 1981.
- [98] L. Peng and L. T. Pileggi, “Compact reduced-order modeling of weakly nonlinear analog and RF circuits,” *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 2, pp. 184-203, Feb. 2005.
- [99] C. Gu, “QLMOR: A Projection-Based Nonlinear Model Order Reduction Approach Using Quadratic-Linear Representation of Nonlinear Systems,” *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 9, pp. 1307-1320, Sept. 2011.
- [100] Y. Chen, “Model order reduction for nonlinear systems”, MIT master thesis, 1999.
- [101] “Math Kernel Library,” Intel Corporation, California, USA, 2013. [Online]. Available: <http://software.intel.com/en-us/intel-mkl> [Feb. 1, 2015].
- [102] O. Schenk and K. Gärtner, “Solving Unsymmetric Sparse Systems of Linear Equations with PARDISO,” *Journal of Future Generation Computer Systems*, vol. 20, no. 3, pp. 475-487, 2004.
- [103] “MA41 Sparse Unsymmetric System,” Science and Technology Facilities Council, May 2011. [Online]. Available: <http://www.hsl.rl.ac.uk/catalogue/ma41.xml> [Feb. 1, 2015].
- [104] A. Gupta and V. Kumar, “Parallel Algorithms for Forward and Back Substitution in Direct Solution of Sparse Linear Systems,” *Proceedings of the IEEE/ACM SC95 Conference Supercomputing*, pp. 74, 1995.

- [105] “Information technology - Portable Operating System Interface (POSIX) Operating System Interface (POSIX),” ISO/IEC/IEEE 9945 (First edition 2009-09-15), pp. c1-3830, Sep. 15, 2009.
- [106] M. L. Crow and M. Ilic, “The parallel implementation of the waveform relaxation method for transient stability simulations,” *IEEE Trans. Power Systems*, vol. 5, no. 3, pp. 922-932, Aug 1990.
- [107] P. Wriggers, *Nonlinear Finite Element Methods*, Springer, 2008, ch. 5.
- [108] “Newton-Like Methods,” *Nonlinear Finite Element Methods*, Colorado University, 2014. [Online]. Available: <http://www.colorado.edu/engineering/cas/courses.d/NFEM.d/NFEM.Ch23.d/NFEM.Ch23.pdf> [Feb. 1, 2015].
- [109] W. Gautchi, *Numerical Analysis: An Introduction*. Boston, MA, USA: Birkhauser, 1997.
- [110] U. M. Ascher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM, 1998.
- [111] B. Perreault, “Introduction to the Kalman Filter and its Derivation,” Apr. 2012. [Online]. Available: [http://www.academia.edu/1512888/Introduction\\_to\\_the\\_Kalman\\_Filter\\_and\\_its\\_Derivation](http://www.academia.edu/1512888/Introduction_to_the_Kalman_Filter_and_its_Derivation) [Feb. 1, 2015].
- [112] K. E. Martin, D. Hamai, M. G. Adamiak, S. Anderson, M. Begovic, G. Benmouyal, G. Brunello, J. Burger, J. Y. Cai, B. Dickerson, V. Gharpure, B. Kennedy, D. Karlsson, A. G. Phadke, J. Salj, V. Skendzic, J. Sperr, Y. Song, C. Huntley, B. Kasztenny, and E. Price, “Exploring the IEEE Standard C37.118–2005 Synchrophasors for Power Systems,” *IEEE Trans. Power Delivery*, vol. 23, no. 4, pp. 1805-1811, Oct. 2008.

## APPENDIX A: VALIDATION OF EQUATION (2-31)

Coherency between buses  $i$  and  $j$  implies  $|\theta_i - \theta_j| < \varepsilon_\theta$

where  $\varepsilon_\theta$  is some small angle (e.g.  $5^\circ$ ). Since  $\tan(x) \approx x$  as  $x \rightarrow 0$ , following condition is considered for coherency identification.

$$\tan(\theta_i - \theta_k) < \tan(\varepsilon_\theta) \quad (\text{A.1})$$

Left hand side of (A.1) can be expanded using trigonometric identities.

$$\tan(\theta_i - \theta_k) = \frac{\tan(\theta_i) - \tan(\theta_k)}{1 + \tan(\theta_i)\tan(\theta_k)} \quad (\text{A.2})$$

Since  $\tan(\theta_i) = \text{Im}(V_i e^{j\delta_i}) / \text{Re}(V_i e^{j\delta_i})$ , (A.2) can be reformulated using real and imaginary part of voltages.

$$\tan(\theta_i - \theta_k) = \frac{\frac{\text{Im}(V_i e^{j\delta_i})}{\text{Re}(V_i e^{j\delta_i})} - \frac{\text{Im}(V_k e^{j\delta_k})}{\text{Re}(V_k e^{j\delta_k})}}{1 + \frac{\text{Im}(V_i e^{j\delta_i})}{\text{Re}(V_i e^{j\delta_i})} \frac{\text{Im}(V_k e^{j\delta_k})}{\text{Re}(V_k e^{j\delta_k})}} \quad (\text{A.3})$$

This equation can be simplified by choosing  $\text{Re}(V_i e^{j\delta_i})\text{Re}(V_k e^{j\delta_k})$  as common denominator.

$$\tan(\theta_i - \theta_k) = \frac{\operatorname{Im}(V_i e^{j\delta_i}) \operatorname{Re}(V_k e^{j\delta_k}) - \operatorname{Re}(V_i e^{j\delta_i}) \operatorname{Im}(V_k e^{j\delta_k})}{\operatorname{Re}(V_k e^{j\delta_k}) \operatorname{Re}(V_i e^{j\delta_i}) + \operatorname{Im}(V_i e^{j\delta_i}) \operatorname{Im}(V_k e^{j\delta_k})} \quad (\text{A.4})$$

Moreover,

$$\operatorname{Im}(V_i e^{j\delta_i}) = \sum_{l=1}^n |M_{i,l}| \sin(\delta_l + \alpha_{i,l}) \quad (\text{A.5})$$

$$\operatorname{Re}(V_i e^{j\delta_i}) = \sum_{l=1}^n |M_{i,l}| \cos(\delta_l + \alpha_{i,l}) \quad (\text{A.6})$$

In (A.4), there are four multiplications between real and imaginary parts which can be expanded using (A.5) and (A.6).

$$\tan(\theta_i - \theta_k) = \frac{A - B}{C + D} \quad (\text{A.7})$$

$$A = \sum_{l_1=1}^n \sum_{l_2=1}^n |M_{i,l_1}| |M_{k,l_2}| \sin(\delta_{l_1} + \alpha_{i,l_1}) \cos(\delta_{l_2} + \alpha_{i,l_2})$$

$$B = \sum_{l_1=1}^n \sum_{l_2=1}^n |M_{i,l_1}| |M_{k,l_2}| \cos(\delta_{l_1} + \alpha_{i,l_1}) \sin(\delta_{l_2} + \alpha_{i,l_2})$$

$$C = \sum_{l_1=1}^n \sum_{l_2=1}^n |M_{i,l_1}| |M_{k,l_2}| \cos(\delta_{l_1} + \alpha_{i,l_1}) \cos(\delta_{l_2} + \alpha_{i,l_2})$$

$$D = \sum_{l_1=1}^n \sum_{l_2=1}^n |M_{i,l_1}| |M_{k,l_2}| \sin(\delta_{l_1} + \alpha_{i,l_1}) \sin(\delta_{l_2} + \alpha_{i,l_2})$$

In (A.7),  $A$  and  $B$  in nominator, and  $C$  and  $D$  in denominator can be aggregated to create one summation. The results can be further aggregated using trigonometric identity  $\sin(x+y) = \sin(x)\cos(y) + \sin(y)\cos(x)$  to get (A.8).

$$\tan(\theta_i - \theta_j) = \frac{\sum_{l_1=1}^n \sum_{l_2=1}^n |M_{i,l_1}| |M_{k,l_2}| \sin(\delta_{l_1} + \alpha_{i,l_1} - \delta_{l_2} - \alpha_{i,l_2})}{\sum_{l_1=1}^n \sum_{l_2=1}^n |M_{i,l_1}| |M_{k,l_2}| \cos(\delta_{l_1} + \alpha_{i,l_1} - \delta_{l_2} - \alpha_{i,l_2})} \quad (\text{A.8})$$

Generally, (A.8) is time-variant since  $\delta$  is a function of time  $t$  and accordingly, nothing can be concluded from (A.8). Instead, summations in nominator and denominator are decomposed into time-dependent and time-independent summations.

$$\tan(\theta_i - \theta_j) = \frac{S_{\sin} + C_{\sin}}{S_{\cos} + C_{\cos}} \quad (\text{A.9})$$

$$S_{\sin} = \sum_{l=1}^n |M_{i,l}| |M_{k,l}| \sin(\alpha_{i,l} - \alpha_{k,l})$$

$$S_{\cos} = \sum_{l=1}^n |M_{i,l}| |M_{k,l}| \cos(\alpha_{i,l} - \alpha_{k,l})$$

$$C_{\sin} = \sum_{l_1=1}^n \sum_{l_2=1}^n |M_{i,l_1}| |M_{k,l_2}| \sin(\delta_{l_1} + \alpha_{i,l_1} - \delta_{l_2} - \alpha_{i,l_2})$$

$$C_{\cos} = \sum_{l_1=1}^n \sum_{l_2=1}^n |M_{i,l_1}| |M_{k,l_2}| \cos(\delta_{l_1} + \alpha_{i,l_1} - \delta_{l_2} - \alpha_{i,l_2})$$

In (A.9),  $S_{\sin}$  represents difference between elements from same columns in rows  $i$  and  $k$  while  $C_{\sin}$  represents mutual effect of different columns in rows  $i$  and  $k$  on  $\tan(\theta_i - \theta_k)$ . The main difference between  $S_{\sin}$  and  $C_{\sin}$  as well as  $S_{\cos}$  and  $C_{\cos}$  relies in their dependency on time. While  $S_{\sin}$  and  $S_{\cos}$  are time-independent,  $C_{\sin}$  and  $C_{\cos}$  depend on time. In the next step, effect of  $C_{\sin}$  and  $C_{\cos}$  on coherency is investigated.

Suppose,

$$|M_{k,l}| = |M_{i,l}| + \varepsilon_{M,l}, \quad (\text{A.10})$$

$$\alpha_{k,l} = \alpha_{i,l} + \varepsilon_{\alpha,l},$$

where  $\varepsilon_{M,l}$  and  $\varepsilon_{\alpha,l}$  are two small numbers and means columns of row  $k$  are slightly different from row  $i$ . Also, note that for each term in  $C_{\sin}$  with  $l_1 = p$  and  $l_2 = q$ , there is exactly one other term with  $l_1 = q$  and  $l_2 = p$ . Separating these terms and computing their summation results in:

$$C_{\sin}|_{l_1=p, l_2=q} + C_{\sin}|_{l_1=q, l_2=p} = |M_{i,p}| \left( |M_{i,q}| + \varepsilon_{M,q} \right) \sin(v + \varepsilon_{\alpha,q}) + |M_{k,q}| \left( |M_{i,p}| + \varepsilon_{M,p} \right) \sin(v + \varepsilon_{\alpha,p}) \quad (\text{A.11})$$

Therefore, for sufficiently small  $\varepsilon_M$  and  $\varepsilon_{\alpha}$ , following relationship holds:

$$C_{\sin}|_{l_1=p, l_2=q} + C_{\sin}|_{l_1=q, l_2=p} \leq |M_{i,p}| |M_{i,q}| \mathcal{O}(\varepsilon_{Max}), \quad (\text{A.12})$$

where  $\varepsilon_{Max} = \max\{\varepsilon_{\alpha,q}, \varepsilon_{\alpha,p}, \varepsilon_{M,q}, \varepsilon_{M,p}\}$ . Since there are  $n^2 - n$  terms in  $C_{\sin}$ ,

$$C_{\sin} \leq (n^2 - n) M_{i,p} \|M_{i,q}\| o(\varepsilon_{Max}) \quad (\text{A.13})$$

Additionally, in a power system, voltage of bus  $i$  is highly affected by adjacent generators whose rotor angle, in turn, are close to each other (less than  $90^\circ$ ) and therefore if assumption (A.10) holds,  $C_{\cos}$  is always positive. Accordingly,

$$\tan(\theta_i - \theta_j) \leq \frac{S_{\sin} + o(\varepsilon_{Max})}{S_{\cos}} \quad (\text{A.14})$$

In order to  $o(\varepsilon_{Max})$  be small, (A.10) should be met. Each term in  $S_{\sin}$  represents the difference between  $M_{i,l}$  and  $M_{k,l}$  and accordingly, if all terms in  $S_{\sin}$  are small, (A.10) is satisfied automatically. Therefore in order to make sure, (A.10) is satisfied and  $\tan(\theta_i - \theta_j) \leq \tan(\varepsilon_\theta)$ , following inequality should hold.

$$\frac{\sum_{l=1}^n |M_{i,l}| \|M_{k,l}\| \sin(\alpha_{i,l} - \alpha_{k,l})}{\sum_{l=1}^n |M_{i,l}| \|M_{k,l}\| \cos(\alpha_{i,l} - \alpha_{k,l})} < \tan(\varepsilon_\theta) + \varepsilon \quad (\text{A.15})$$

where  $\varepsilon_\theta$  is some small angle.