Fast Simulation, Monitoring, and Mitigation of Cascading Failure

Final Project Report

Power Systems Engineering Research Center

*Empowering Minds to Engineer
the Future Electric Energy System*

Fast Simulation, Monitoring and Mitigation of Cascading Failure

Final Project Report

Project Team

**Ian Dobson, Project Leader,
University of Wisconsin-Madison**

James McCalley, Iowa State University

Chen-Ching Liu, University College Dublin

PSERC Publication 10-18

October 2010

Information about this project

For information about this project contact:

Ian Dobson
Professor
Department of Electrical and Computer Engineering
University of Wisconsin - Madison
1415 Engineering Drive
Madison, Wisconsin 53706
dobson@engr.wisc.edu
608-262-2661

Power Systems Engineering Research Center

The Power Systems Engineering Research Center (PSERC) is a multi-university Center conducting research on challenges facing the electric power industry and educating the next generation of power engineers. More information about PSERC can be found at the Center's website: http://www.pserc.org.

Power Systems Engineering Research Center
Arizona State University
577 Engineering Research Center
Tempe, Arizona 85287-5706
Phone: 480-965-1643
Fax: 480-965-0745

Notice Concerning Copyright Material

Acknowledgements

This is the final report for the Power Systems Engineering Research Center (PSERC) research project S-32 titled “Fast Simulation, Monitoring, and Mitigation of Cascading Failure.” We express our appreciation for the support provided by PSERC’s industry members.

We also thank the following for their technical support:

Ali Chowdhury - CAISO
Innocent Kamwa - IREQ
Lawrence Jones - ALSTOM Grid
Mahendra Patel - PJM
Jim Viikinsalo - Southern Company

These individuals are with companies that are industry members of PSERC.

Executive Summary

Cascading failure occurs when an initial disturbance in a power transmission system propagates to cause a widespread blackout. Large cascading failures are not frequent, but they have substantial risk due to their large societal costs. The August 2003 cascading blackout of portions of the northeastern U.S. and eastern Canada had direct costs of the order of $6-$10 billion. Large blackouts can also have significant indirect effects, including negative perceptions of the electric power industry and changes in industry regulation.

The project has pursued advances in methods to address cascading failure risk, particularly fast simulation for online support for operators, fast computation of emergency islanding control to mitigate cascading, and methods to monitor and quantify cascading risk from recorded or simulated data.

Part A: High Speed, Extended-Term Time Domain Simulation for Online Cascading Analysis of Power Systems

Use of online tools by system operators to mitigate cascading failure risk has been limited by computational complexity and time. Part A makes algorithmic improvements in numerical methods.

In a previous PSERC project we made progress toward providing operators with very fast computational tools to predict system response and to identify corrective actions for low probability, high-consequence catastrophic events (e.g., blackouts). These tools relied on analytical modeling and fast numerical simulation studies. The computational efficiency was mainly harnessed by reducing the solution time for solving sparse linear systems.

In this project we have made further progress towards increasing computational efficiency of time domain simulators through algorithmic improvements and application of new integration algorithms. We used the Hammer-Hollingsworth 4 (HH4) algorithm which has all the properties of Trapezoidal integration algorithms commonly found in time domain simulation in commercial software. Hammer-Hollingsworth 4 has higher order precision because it is a higher order algorithm. It can speed simulation solutions because it is capable of larger time steps during a simulation.

We developed a new algorithm implementation design using parallel platforms. The essential design feature is the separation of the stiff and non-stiff part from differential algebraic equations (DAE) via a recursive projection method. Both parts are then solved through parallel implementation. This work exploits the error estimation for stiffness detection. For the stiff part, waveform relaxation is used for further division of the non-linear equations obtained from the DAE via discretization with integration algorithms. The non-linear equations are solved with Newton-based methods using fast sparse libraries from the public domain.

To achieve high computationally efficiency suitable for online time domain simulation, we used a combination of (1) state-of-the-art algorithms for integration, (2) solvers for non-linear and linear equations, and (3) efficient hardware.

We plan to continue our online tool development research through implementation of the parallel strategy.

Part B: Power System Reconfiguration Based on Multilevel Graph Partitioning

Power system reconfiguration is an emergency control against cascading events. Reconfiguration responds to a worsening operating condition and isolates the impact of a disturbance by isolating a specific area. In anticipation of a vulnerable operation condition, the power network can be partitioned into two or more subsystems (i.e., electrical islands). If a major problem occurs in a subsystem and it is successfully isolated, the remaining system can still operate in an acceptable condition. Such a network reconfiguration strategy is a “Smart Grid” technique to enhance the shock absorption capabilities of a power system.

Determination of the system partitioning strategy is a challenging task due to the large-scale nature of power systems and the complexity of system operating constraints. In our research, we develop an area-partitioning algorithm that applies the state-of-the-art multi-level graph partitioning technique to obtain a new system configuration with minimized real and reactive power flows on the partitioning boundary. The proposed algorithm is intended to sustain the system frequency and voltage after islanding by minimizing the real and reactive power imbalance within each island.

Simulation results on a 200-bus system showed that the area-partitioning method performs better than an algorithm that only considers real power balance. The higher performance is achieved by incorporating reactive power balance, thereby improving the voltage profile. The proposed algorithm successfully separated a 200-bus system into two stable islands and avoided a system collapse. Simulation results on a 22,000-bus system demonstrated that the proposed algorithm is highly efficient. As a result, we found that it is feasible to determine the optimal partitioning strategy and identify the new configuration in a real-time environment.

The next steps are to validate on very large test systems, adjust the partitioning to take account of company boundaries, and determine the wide area protection and control systems needed for implementation. Moreover, post-islanding controls, such as automatic capacitor switching and automatic generation control, should be developed to further stabilize the islands formed by system reconfiguration. An automatic defense system scheme that integrates area partitioning operations, load shedding and post-islanding control actions should be investigated in the future research.

Part C: Methods to Estimate Propagation and the Distribution of Blackout Extent for Cascading Blackouts

In cascading failure blackouts, initial outages become widespread by propagating to cause further failures. We developed and tested methods of quantifying the amount of propagation using both observed utility data and simulated cascading outages. The amount of propagation describes the power system resilience to cascading failure and is a key parameter of high-level models that can estimate the statistics of the blackout size. The project showed that two such high level probabilistic models of cascading failure were close to each other and generally consistent with the observed and simulated data. That is, high-level models can, in most of the cases tested, estimate the distribution of blackout size from the size of the initial failures and the amount of propagation.

For cascading failure simulations, we showed that the chance of large cascading blackouts can be estimated with many fewer simulation runs. Two measures of blackout size are (1) number of transmission lines outaged and (2) amount of customer load that is outaged. We developed methods for both measures and analyzed them with simulated data.

The observed blackout data was the transmission line outage data that must be reported to North American Electric Reliability Corporation using the Transmission Availability Data System. We computed how these transmission line outages propagated, showed that the data was consistent with a high-level probabilistic model, and then showed how to estimate the statistics of the total number of line outages. This appears to be a practical method of quantifying cascading failure based on data available to the industry. One outcome is that the statistics of the total number of lines outaged can be determined from much shorter data records. (The alternative of waiting for enough large blackouts to occur to get meaningful statistics takes too long.) Access to testing data is so far limited, but the results on the available data are very good.

The next steps are to obtain more observed data sets to gain more experience with applying the calculations, design new metrics for cascading failure to summarize the information in the calculations, and make the calculations available to industry with software so that the calculations can be tested and improved with industry advice.

Part A

High Speed Extended Term Time Domain Simulation for Online Cascading Analysis of Power System

Faculty:

James McCalley
Iowa State University

Ph.D. Student:

Chuan Fu
Iowa State University

Information about Part A of this report

For information contact:

James D. McCalley
Iowa State University
Electrical and Computer Engineering Department
Ames, Iowa 50011
Phone: 515-294-4844
Fax: 515-294-4263
Email: jdm@iastate.edu

Power Systems Engineering Research Center

The Power Systems Engineering Research Center (PSERC) is a multi-university Center conducting research on challenges facing the electric power industry and educating the next generation of power engineers. More information about PSERC can be found at the Center's website: http://www.pserc.org.

Power Systems Engineering Research Center
Arizona State University
577 Engineering Research Center
Tempe, Arizona 85287-5706
Phone: 480-965-1643
Fax: 480-965-0745

Notice Concerning Copyright Material

PSERC members are given permission to copy without fee all or part of this publication for internal use if appropriate attribution is given to this document as the source material. This report is available for downloading from the PSERC website.

Table of Contents

Table of Contents
(continued)

List of Figures

List of Figures (continued)

List of Tables

1. Introduction

1.1 Overview

A cascading event is recognized as chronological sequence of multiple lower-order dependent events from [1]. Caused by an initial disturbance on power transmission system, cascading failure can propagate to become a widespread blackout. Although the most severe cascading events occur infrequently, however when they occur they can have significantly economic and social impacts. A typical example of a large cascading event is the North American blackout that happened in August 2003, which not only had direct impacts on residents and customers but also on the regulatory environment and perceptions of the entire industry. The vulnerability assessment of cascading events need to be evaluated and managed in power system operation and planning, and constructive information about the security of the power system, especially on the analysis of high consequence events, needs to be supplied to help operators in control center.

Extended-term time domain simulation is significantly beneficial for the analysis of power system security, especially when cascading events occur on the system and span over minutes or even hours. This is because the power system response to disturbances is decided not only by fast dynamics led by electronic elements, but also by the action of slow processes such as rotor of machines and load dynamic. The simulation time from minutes to hours is helpful to investigate the effect of a series of events on power system and unearth the power system's ability to withstand large disturbances over extended periods of time. Due to the fact that such information is not available today in control centers with dynamic simulation tools, it is meaningful and necessary to provide operators with more information about extended-term simulation, which makes it easy to see the impact brought by the potentially possible high consequence events

There are mainly two objectives for the extended-term time domain simulation of power system, 1) online monitoring and tracking of high consequence events, 2) providing preventive or corrective action strategies for mitigating the impact of the high consequences events. For online application of extended-term simulation, one feasible idea is to simulate a system with the well selected contingencies (N-1 and N-k) as fast as possible and then corresponding corrective or preventive actions are taken to decrease damage to the system. [1,2,3] The simulation process associated with contingency selection and corrective operation can be illustrated in Figure 1, which includes three parts, 1) contingency selection, 2) time domain simulation and 3) intelligent and automatic operation part.

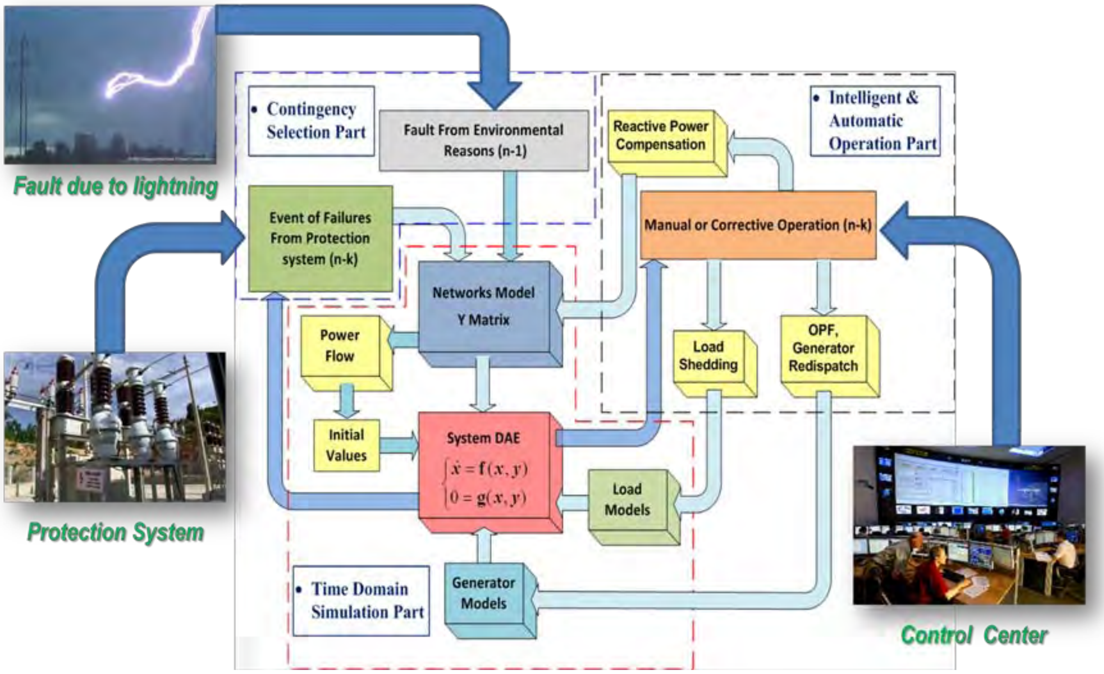

Fault due to lightning
Contingency Selection Part
Fault From Environmental Reasons (n-1)
Event of Failures From Protection system (n-k)
Networks Model Y Matrix
Power Flow
Initial Values
System DAE
$\begin{cases}\dot{x}=f(x,y)\\0=g(x,y)\end{cases}$
Load Models
Generator Models
Time Domain Simulation Part
Protection System
Reactive Power Compensation
Intelligent & Automatic Operation Part
Manual or Corrective Operation (n-k)
Load Shedding
OPF, Generator Redispatch
Control Center

Figure 1. Relationship between contingency selection, time domain simulation, and corrective operation

- *Contingency selection*

For a power system in normal operation, all state variables are in balanced state, which means that all state variables describing the system don’t vary with the time. The balance of the system can be broken by a potential event such as outage of transmission lines due to lightning and the impedance matrix Y of the system will be changed due to the outage of transmission lines. Contingency selection part in Figure 1 supplies the contingencies (n-1) from the networks topology processing of EMS (Energy Management System) or potential possible contingencies via historic statistic data. The cascading events are usually trigged by protection system, and with the analysis of protection system the potential high consequence event(n-k) can be supplied, in which a lot of research has been done [1].

- *Time domain simulation*

Because of the change of impedance matrix Y due to the potential events, some state equations describing the system will be not be balanced and dynamic process of many dynamic elements such as generators, excitation, governors and load will occur. The essence of analyzing the dynamics of power system is to solve a large set of differential algebraic equations (DAE). The initial values describing the differential equations can be acquired via computation of power flow when the system is in static state.

- *Intelligent and automatic operation part*

After the occurrence of a potential high consequence event, a stable case is one in which each variable in the DAE system describing the power system is able to achieve a new balanced point and each variable is within an acceptable bound. However, for the most of cases, the high consequence event may lead to the instability or oscillation of system, and also some variables may become out of the acceptable bounds. Corrective actions like load shedding, generation re-dispatch and reactive power compensation, can be taken to prevent the possible cascading outages. The information about these actions can be supplied by computations, such as optimal power flow, or the experience of operators.

During this simulation process, high computational efficiency plays a decisive role in the whole application, since the faster the simulation is, the less damage there will be. On the other hand, highly efficient time domain simulation is always welcomed for offline analysis of power system security, especially when the power system is of large scale. It usually happens that much longer time has to be taken to simulate a large scale system for very short period. High speed of extended-term time-domain simulation for a large scale system is required for both industrial usage and academic research.

1.2 High Speed Extended Term (HSET) Time Domain Simulator (TDS)

High-speed extended term (HSET) time domain simulator (TDS) is of a new functionality for control center security assessment. This functionality is motivated by the low-probability, high consequence events to which the power system is continuously exposed. Such events, usually comprised of multi-element (so-called “N-k”) outages, often causing additional cascading events spanning minutes or even hours, are typically perceived to be unlikely and therefore undeserving of preventive action and the associated increased costs due to off-economic dispatch typically involved in preventive action. Yet, such events do occur, and in today’s energy control centers, operational personnel have no decision-support function available to assist them in identifying effective corrective action, or even in becoming familiar with system performance under such events[4] .

HSET-TDS, intended to be a part of the energy management system (EMS), contains the following attributes:

Probability-based contingency selection: Contingencies are selected based on topological processing of node-breaker data based on user-specified probability order of magnitude.

Extended-term: Cascading sequences can play out over several hours, and so HSET-TDS has capability to simulate for this time frame.

Computational efficiency: Time-domain simulation, involving the solution of numerical integration, is computationally intense. Therefore, it has been extremely challenging in today’s control centers to implement associated functionality even for a short amount of simulated time, e.g., 10 seconds, for a limited number of contingencies. On-line simulation of minutes to hours for a very large number of contingencies requires computational efficiency several orders of magnitude greater than what is today’s state-of-the-art.

Fast and slow-dynamics: HSET-TDS must capture phenomena such as inertial instability affected by traditional fast dynamics (e.g., machine, excitation, speed-governing) but also phenomena such as voltage instability and cascading affected by slow dynamics (e.g., AGC, thermal changes in boilers, tap changing, and load variation).

Failure detection: HSET-TDS must detect, within the simulation, unacceptable system performance such as out of step conditions, voltage deterioration, and thermal overload.

Corrective action identification: Failure detection must be followed by, within the simulation, the identification of corrective actions such as redispatch, load shedding, network switching, or islanding.

Result storage: HSET-TDS may be helpful in a responsive mode where it is run following initiation of a severe disturbance. Alternatively, we envision that it will play a heavy role in an anticipatory mode, continuously computing responses to many contingencies and storing preparatory corrective actions that would be accessed by operational personnel should one of the contingencies occur. The goal is to cover as much as the event-probability space as possible within a particulate computing time, e.g., 1 hour. Results can be archived and re-used when similar conditions are met.

1.3 Motivation

In order to meet the requirement of on-line security assessment, we have been developing a new functionality that we refer to as high-speed extended term (HSET) time domain simulator (TDS). HSET-TDS, intended to be a part of the energy management system (EMS), contains the attributes, such as probability- based contingency selection, extended-term, computational efficiency, fast and slow dynamics, corrective action identification and result storage. An important attribute of HSET-TDS is very fast on-line computational capability to predict extended-term dynamic system response to disturbance. This work mainly focused on the acceleration of the extended term time domain simulation of power system from the following five perspectives, 1) hardware, 2) integration methods, 3) nonlinear solvers, 4) linear solver libraries.

As far as the hardware is concerned, HSET-TDS is being developed on two different platforms, sequential computing based on Windows and parallel computing based on Linux, which can be shown in Figure 2. The purpose of the sequential computing of HSET-TDS is to explore new numerical methods, nonlinear solvers and linear solvers which are suitable for time domain simulation of power system, and compare the HSET-TDS with currently available commercial software. Besides, for the small and medial scale power system, the sequential computing is more meaningful than parallel computing. The parallel computing version of HSET-TDS is based on Linux operation systems, which are currently available on parallel computing clusters of HPC-Class, and IBM Bluegene/L in Iowa State University. The goal of parallel computing of HSET-TDS is to explore the parallel computing algorithms suitable for extended term time domain simulation when the power system is large. Additionally, we intended to make use of high performance computer, IBM Bluegene/L, with large processors (1024), to experiment whether a high simulation speedup can be acquired. The parallel computing version of HSET-TDS is also intended to meet the requirement of high speed for online simulation of cascading on large power systems.

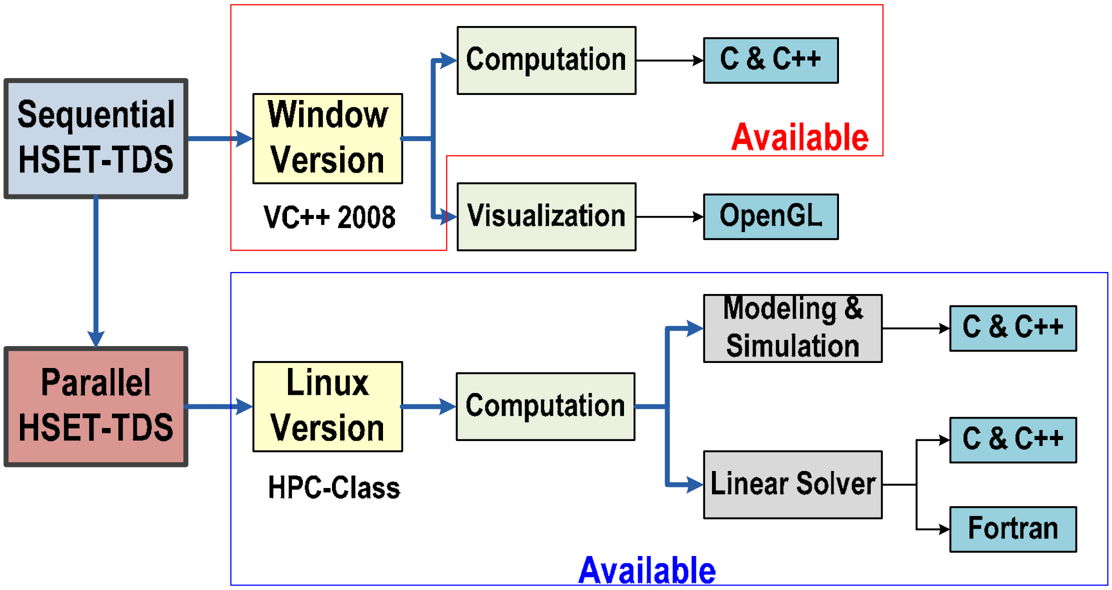


Figure 2. Hardware platforms of HSET-TDS

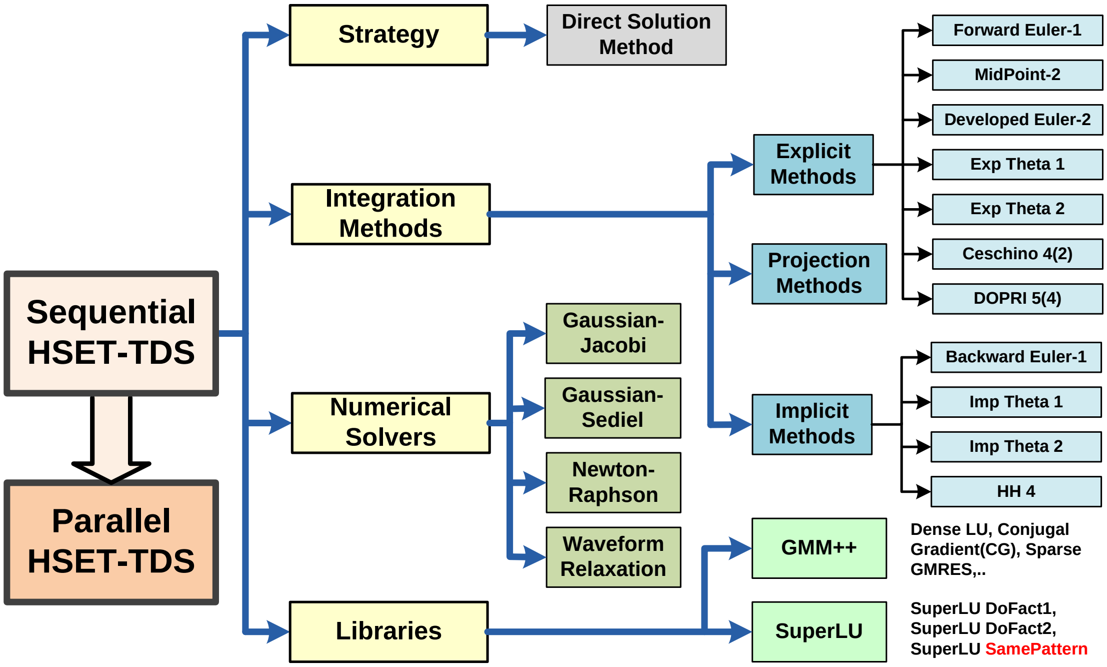


Figure 3. Structure of numerical methods in HSET-TDS

HSET-TDS has been developed to include numerical methods as many as possible to find the more suitable ones, and the structure of HSET-TDS related to numerical methods is shown in Figure 3. The motivation of exploring each categories can be described as follows:

- *Strategies.*

The classification of strategies is introduced in Chapter 2, and it includes alternating solution method and direct solution method. The motivation of this part is to find a suitable and efficient programming structure for HSET-TDS, and then to make the execution of integration methods, nonlinear solvers and linear solvers more flexible.

- *Integration methods*

The integration methods in HSET-TDS have been developed to include many methods, including explicit methods, implicit method and projection methods, which are controlled by users and are intended to be applied into simulation freely. The motivation of this part is to explore new integration method which is suitable for power system. An integration method, which is called Hammer-Hollingsworth 4 (introduced in Chapter 3), has been applied for the first time for time domain simulation in the HSET-TDS, and some satisfying simulation results have been achieved.

- *Nonlinear solvers*

Nonlinear solvers are indispensible part in the solution of differential algebraic equations. The motivation of this part is to explore some parallel algorithm (such as Waveform Relaxation), which is able to separate the algebraic equations into independent parts and make the parallel computing easily realized.

- *Linear solver libraries*

The process of solving Ax=b is necessary in the solution of algebraic equations when Newton method is used. Currently, there are many open source sparse linear solver libraries available in the world, such as GMM++, SuperLU, UMFpack, MUMPS. Some of these libraries are just for sequential computing, and some can be used for parallel computing. Besides, the performance of these libraries are different because of different methods and different programming approaches. Additionally, since the jacobian matrix in power system is high sparse, and the sparsity can impact the simulation performance tremendously. It is meaningful to include as many as linear solvers in HSET-TDS and compare the performances of these linear solvers.

Focusing on strategies, integration methods and linear solver libraries on how to enhance the efficiency of time-domain simulation, this work introduces a strategy of directive solution method, a new implicit integration method of Hammer-Hollingsworth 4(HH4), and serial & parallel SuperLU library of open source linear solver. Using small and large scale power system cases, all of proposed numerical methods are compared with traditional methods adopted in many commercial softwares.

2. Approaches for High Speed Execution for HSET-TDS

2.1 Problem Formulation

Time domain simulation of power system involves the solution of a large number of differential algebraic equations (DAEs), which are constructed based on the modeling of the power system electric elements and networks, such as generators, exciters, governors and other electronics devices. The general form of the DAEs can be described as follows.

$$\begin{cases} \dot{x} = \mathbf{f}(x, y) & (1a) \\ 0 = \mathbf{g}(x, y) & (1b) \end{cases} \qquad (2.1)$$

The ordinary differential equations (ODE) in (1*a*) describe the dynamic property of machines and the associated control systems correspondingly. The algebraic equations (AEs) in (1*b*) describe the network steady state equations and the machine algebraic equations. When a series of events such as faults initialization, faults clearing, lines outage, or operation on machines are investigated, the DAE system will change, and the changes are dependent on the variation of topology and the parameters of machines.

2.2 Approaches for Time Domain Simulation

Electric power system contains power networks and many different kinds of elements, such as generators, exciters, governors and other electronics devices, all of which can be describe as a set of differential algebraic equations (DAEs)[5,6]. Thus, the essence of time-domain simulation is to solve a set of DAEs on time domain. The methodologies for time-domain simulation of power system can described as a hierarchical pyramid shown in Figure 4. There are five categories of methodologies on programming during the process of solving DAE system in power system simulation, 1) hardwares for execute time domain simulation, 2) general methods for programming power systems and solving DAE, 3) numerical integration methods for discretization of differential equations, 4) numerical iterative solvers for solving nonlinear algebraic equations, 5) solver libraries for linear algebraic equations. In this report, we define following terminologies to describe each hierarchical category.

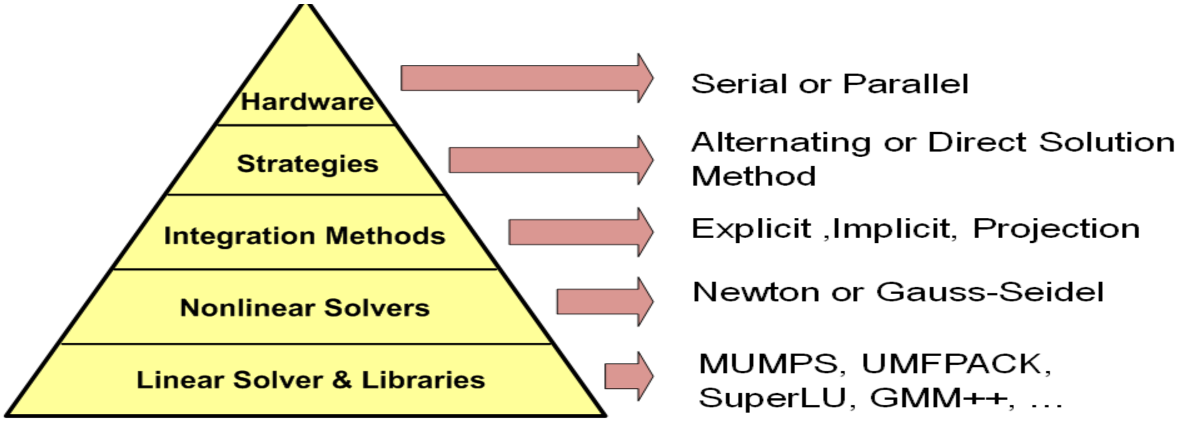

Hardware
Strategies
Integration Methods
Nonlinear Solvers
Linear Solver & Libraries
Serial or Parallel
Alternating or Direct Solution Method
Explicit ,Implicit, Projection
Newton or Gauss-Seidel
MUMPS, UMFPACK, SuperLU, GMM++, ...

Figure 4. Hierarchical pyramid for time-domain simulation

Hardware – the hardware where the time domain simulation is executed. It can be classified into platforms for sequential or parallel computing.

Strategy – the general methods for constructing the DAE system and programming the solution of the system. There are two strategies in this category, alternating solution method and direct solution method[6].

Integration method – the numerical integration methods for discretizing the ordinary differential equations (ODE) in the whole DAE system.

Nonlinear solvers – the numerical iterative solvers for solving nonlinear algebraic equations. The usual solver for this category is Newton method, which can be utilized to solve the value of next time step in the ODE system when implicit integration methods are used and solve the algebraic variables in the DAE system after the differential variable of the next time have been obtained.

Libraries – the various solvers for linear algebraic equations like Ax=b which are indispensible in the process of Newton method. There are many available open source libraries for this category, such as Gmm++, SuperLU.

During the process of time-domain simulation, the selection of different choices in each hierarchical category of the pyramid will lead to different programming scheme, different computational efficiency and different computational precision. Besides, the different choice in the top category has much influence on the bottom category, especially in the computational quantity. Currently, much information for each category in Figure 4 is reported in the technique papers of many kinds of commercial software for the time-domain simulation of power system, such as PSS/E [7], BPA [6], ETMSP [8], EXTAB [9], EUROSTAG [10,11]. The following sections will discuss each category respectively.

2.3 Hardware

Hardware is the basis for any calculation on computers. According to the numbers of processors used in the computation of time domain simulation, we classify the hardware into the platforms for sequential computing and parallel computing. For the sequential computing, the HSET-TDS is generally developed on the PC with windows; while for the parallel computing, the HSET-TDS is complied and built on Linux operation system which is adopted by high performance computers in Iowa State University. In this section we mainly introduce the hardware for parallel computing.

Table 1. High Performance Computing local recourses in Iowa State University

HPC-class Cluster	BlueGene/L	Sun
Front-end 3.06 GHz dual Intel Xeon server with 2 GB of error-correcting memory.	Front-end node, service node, and storage nodes	
16 compute nodes, 2.8 GHz dual Intel Xeons each with 2GB of ECC memory and 73GB of scratch disk.	**1024** compute nodes, dual-core PPC440 CPU, 700 Mhz, 512 MB RAM	400 nodes, dual processor, AMD quad core, **3200** cores, 3.2 TB memory,
	11 TB storage	96 TB storage
	5.7 TF peak compute power	27.6TF peak compute power
Handles small-medium jobs	Handles large jobs	Handles large jobs

Table 1 illustrates the high performance computers available in Iowa State University. HSET-TDS is mainly debugged on two platforms, HPC-class cluster and BlueGene/L. The IBM Blue Gene/L represents today's state of the art in computing. Compared to other computing solutions, it is small and very fast, has low power consumption, and is relatively inexpensive. These attributes are achieved by providing massive parallelism via thousands of processing nodes connected together and organized into a grid, mesh, torus or hypercube arrangement to allow each node to communicate with the other nodes. Iowa State University has recently purchased a Blue Gene/L consisting of 1024 chips, where each chip has two modified PowerPC® 440s running at 700 MHz. The structure of BlueGene/L is shown in Figure 5. These chips are connected by five networks having latency of about 4 microseconds and bandwidth of 350 Mb/sec. Each chip manages 64 compute nodes for a total of 65536 compute nodes. The system stands within a rack having power consumption of 28.14 kW. It has 512 GB of RAM and executes at 5.7 Tflops, in comparison to a 2.8 Ghz Pentium machine which typically has about 2 GB RAM and executes at 5.6 Gflops. A key feature of the Blue Gene/L is that its hardware is designed so that computational improvements are maximized when an algorithm is inherently parallelizable and implemented accordingly. We intend to achieve good parallelization results on Bluegene/L, and the preliminary experiments are made on HPC-Class.

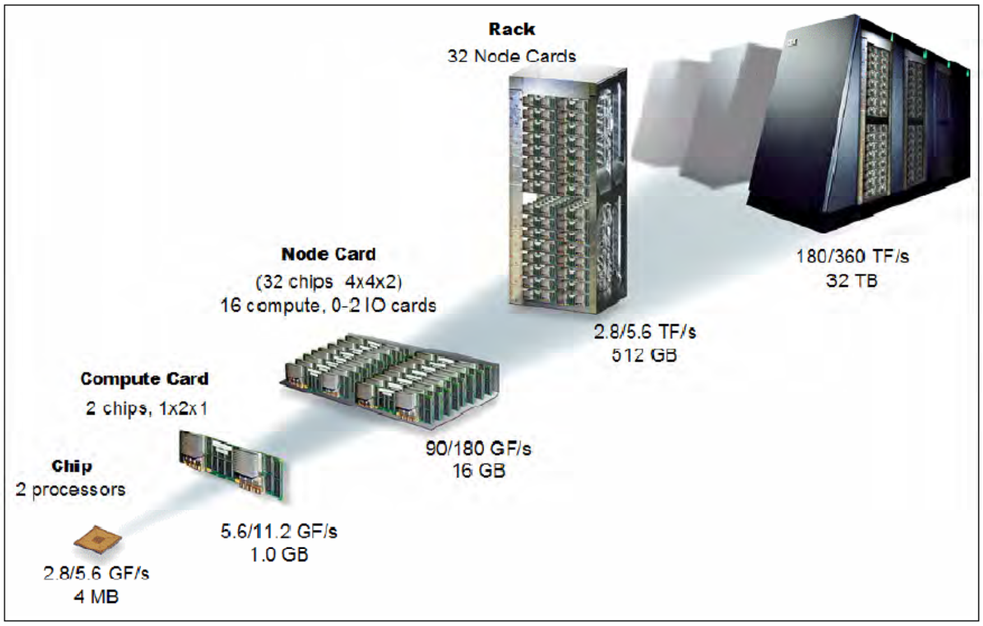


Figure 5. BlueGene/L at a glance

2.4 Strategies

Power system includes networks and various electric elements. Generator and load are the main parts for dynamic analysis. The Figure 6 illustrates the relationship of main variables between generator, load and power networks. The number of generators and loads is dependent on different cases, and different generators includes different control systems constructed by exciters, governors, PSS, etc. Therefore, we cannot use a static mode to construct and program the whole DAE system. There are two kinds of strategies of constructing and programming DAE system, alternating solution method and direct solution method.

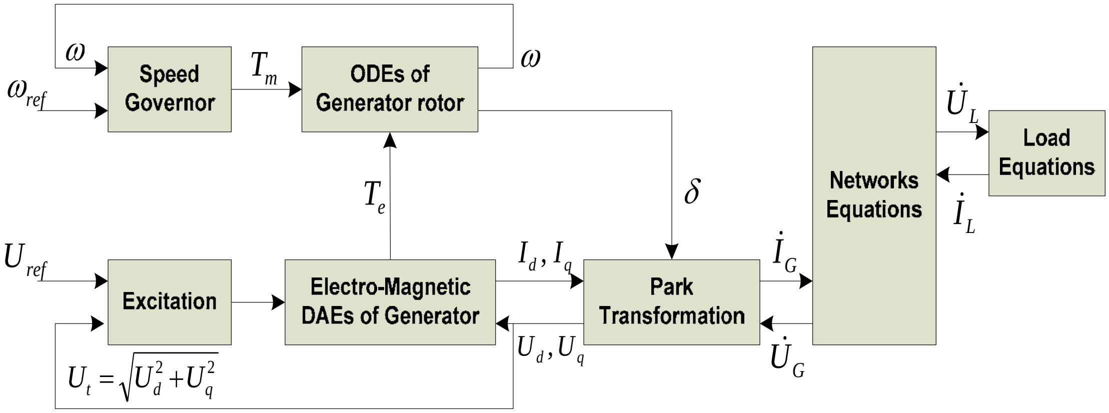


Figure 6. DAE construction of power system

2.4.1 Alternating Solution Method

The strategy of Alternating Solution Method (ASM)[6] is adopted in many kinds of software such as BPA and EXTAB. The main idea is to solve the injective current of each node using the voltage value of last step and then correct the voltage values by solving networks equations. The whole iterative process can be illustrated in Figure 7, where it can be seen that there are main iterative loop and sub iterative loops during the process of computing the next step value of DAE system. The main loop iterates bus voltage between networks, and generators and load of inductor motor, while the sub loops iterate the differential variables describing dynamics of generators and inductor motors. An advantage of this method is that the generators can be replaced by equivalent circuit with constant current sources and admittances, which can be also combined to network admittance matrix, and then it will be convenient to solve bus voltages by solving linear complex algebraic equations. However, much computation has to be added due to the necessary iterative processes from main iterative loop and sub iterative loops shown in Figure 7.

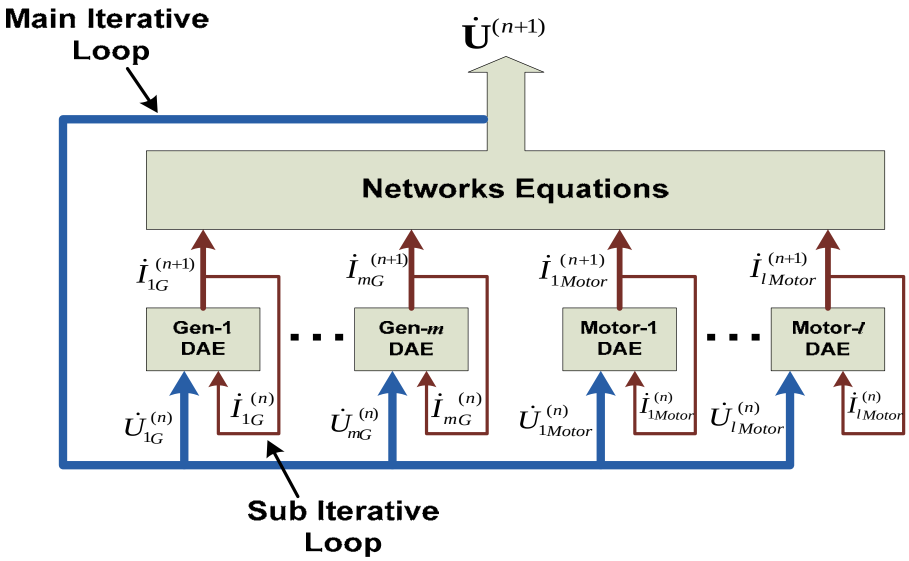


Figure 7. The iterative process of Alternating Solution method

2.4.2 Direct Solution Method

The main idea of direct solution method is first constructing all the ODEs and algebraic equations describing electric elements and power networks and then solving the whole system DAE with numerical methods. One difficulty of programming and realizing the direct solution method is how to organize differential and algebraic variables. A kind of organization structure, keeping sparsity, shown in Figure 8, is adopted in HSET-TDS for differential and algebraic variables in a DAE system of power system. It can be noticed that this structure includes all the details in the power system, and thus much memory is needed to construct the final DAE system. Besides, since the scale of DAE system becomes larger compared with that in alternating solution method, it will take much time to solve the bigger scale linear equations if sparsity is not considered. However, the whole structure in Figure 8 can hold the variable sparsity in each differential and algebraic equation, and this attribute will greatly improve the efficiency of time-domain simulation. Additionally, direct solution method is able to supply final expression of power system dynamics, and thus makes it easy to apply or change different integration methods to time-domain simulation.

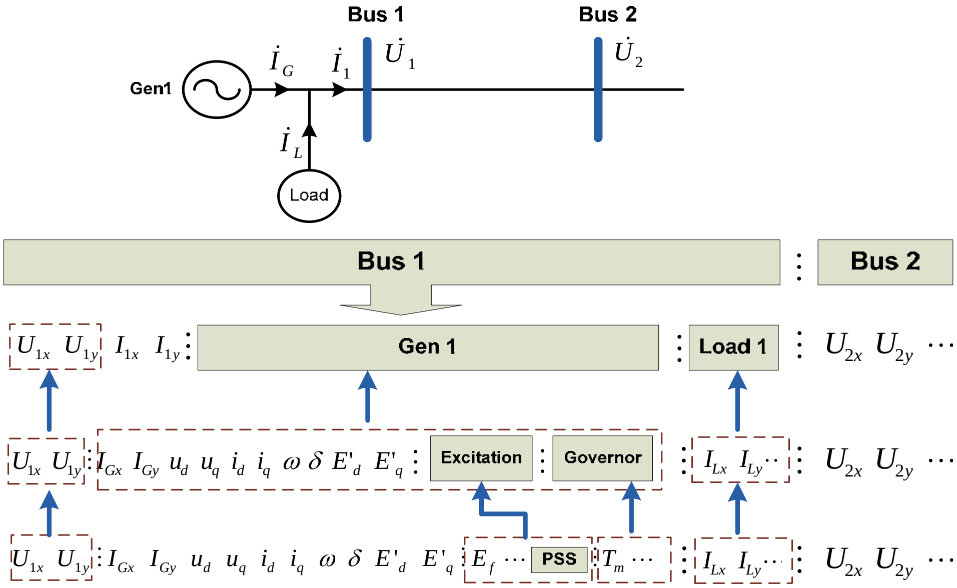

Bus 1
Bus 2
Gen1
$\dot{I}_G$
$\dot{I}_1$
$\dot{U}_1$
$\dot{U}_2$
$\dot{I}_L$
Load
Bus 1
Bus 2
U_{1x} U_{1y} I_{1x} I_{1y}
Gen 1
Load 1
U_{2x} U_{2y} $\cdots$
U_{1x} U_{1y} I_{Gx} I_{Gy} u_d u_q i_d i_q ω δ E'_d E'_q
Excitation
Governor
I_{Lx} I_{Ly}
U_{2x} U_{2y} $\cdots$
U_{1x} U_{1y} I_{Gx} I_{Gy} u_d u_q i_d i_q ω δ E'_d E'_q
E_f $\cdots$
PSS
T_m $\cdots$
I_{Lx} I_{Ly}
U_{2x} U_{2y} $\cdots$

Figure 8. Organization structure of differential and algebraic variables

(The generator shown is of 4th order model)

2.5 Integration Methods

Numerical integration methods is the core for the solution of DAE. All the listed commercial software utilize similar numerical integration methods, which is shown in Table 2. It is seen that trapezoidal rule algorithm is used by each of commercial software in different forms. For example, the θ-method formula [9] used by EXTAB is trapezoidal rule when $\theta = 0.5$, and 2nd order Adams used by EUROSTAG share the same formula with trapezoidal rule. The more details about the traditional numerical methods are discussed in Chapter 2. Additionally, a lot of investigation has been taken on the developed numerical methods, such as multi-rated method[12][13] and stiffness decoupled method[14,15,16] , to make the whole DAE solution believable because of numerical stability and stiffness problem. Most of the developed methods are based on the trapezoidal rule because of its attribute of A-stability.

In HSET-TDS, a new integrator named Hammer-Hollingsworth 4 [41] is adopted, and the details are discussed in Chapter 3. Hammer-Hollingsworth 4 is not only A-stable sharing the same stability domain as Trapezoidal rule, but it also is has the ability to compute the value of next point more precisely. The attribute of high precision make it possible to enlarge the integration step, and therefore the whole integration times can be substantially decreased.

Table 2. Integration methods adopted by several commercial software

Commercial Software	Integration Method	Step Technique
PSS/E	Trapezoidal rule	Fixed step
BPA	Trapezoidal rule	Fixed step
ETMSP	Trapezoidal rule	Variable step
EXTAB	θ-method ($_{C}$=0.47)	Variable step
EUROSTAG	Mixed Adams-BDF (2^{nd} order Adams)	Variable step

2.6 Nonlinear Algebraic Equations Solver

The process of solving nonlinear algebraic equations, which is the third stage in Figure 4, is indispensible for computing the next step values with the nonlinear equations generated by the third stage of integration methods. Also, nonlinear algebraic equations solver supplies the linear algebraic equations to the fifth stage in Figure 4, and then these linear algebraic equations will be solved by linear equations libraries .There are two main iterative methods for solving nonlinear algebraic equations, i) Newton-Raphson method, and ii) Gaussian-Seidel method.

2.6.1 Newton-Raphson Method

Newton-Raphson method is an efficient method to solve the nonlinear equations, and it has been used in power system for long time, such as the solution of power flow. The main attribute of Newton method is that the solution is solved from the linear optimum (slope) direction, and thus the iterative times is least compared to other methods. Figure 9 illustrate the flow chart on the relationship between the numerical integration method, Newton method and linear solver libraries. In time domain simulation, the linear algebraic equations from Newton method is usually of large dimension and high sparsity. To guarantee the computational efficiency, it is necessary to adopt sparse linear solver to solve the linear equations.

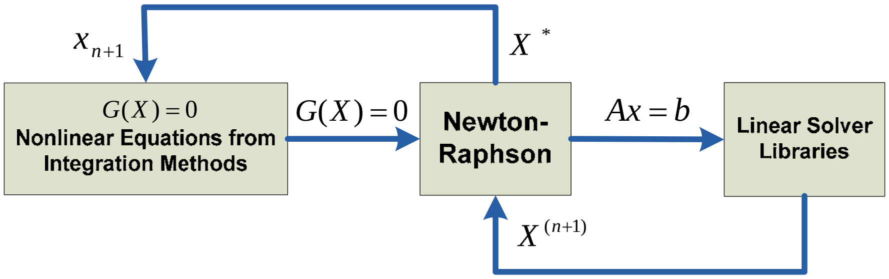


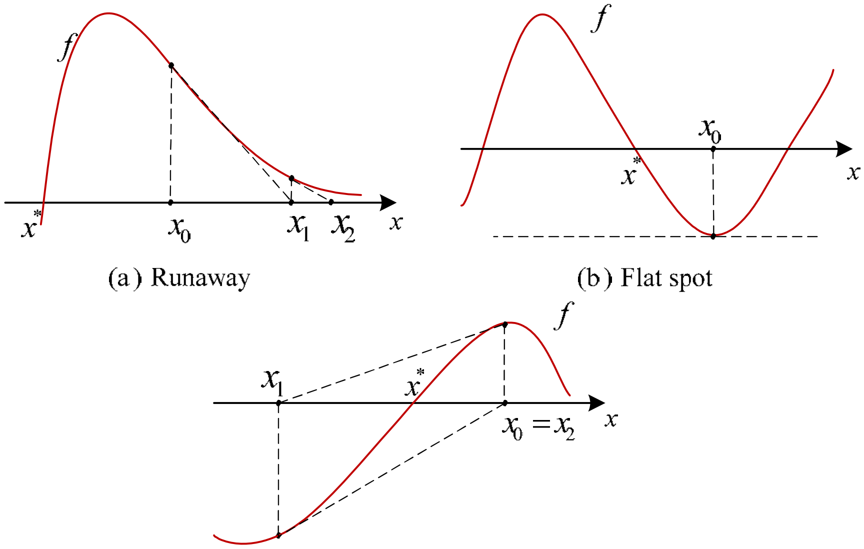
Figure 9. Newton method in time-domain simulation

Additionally, when the integration step becomes big, failure of Newton-Raphson method may occur because of bad starting points. Figure 10 from [17] shows three possible reasons for the failure of Newton method. Paper [9] introduced a modified Newton method where a deceleration factor is introduced. After the deceleration factor is involved, failure of Newton method due to "cycle" may be avoided, however the convergent direction is no longer the optimum, and thus iteration time will be increased. A strategy used in HSET-TDS to tackle with the failure of Newton method is the step control technique discussed in last section, and a threshold of iteration times of Newton method is considered be a criterion to vary integration step. For the nonlinear algebraic equations $G(X)=0$, the modified Newton-Raphson method can be described as follows.

$$\begin{cases} X^{(n+1)} = X^{(n)} - \alpha \cdot \Delta X \\ [J]\Delta X = [b] \end{cases} \tag{2.2}$$

where $[J]=\left[\frac{\partial G}{\partial X}\right]\Big|_{X=X^{(n)}}$ denotes Jacobian matrix; α is deceleration factor; $[b]=[G(X^{(n)})]$.

At the beginning the deceleration factor is set to be 1.0, and it will be decreased by a small value if the max norm of ΔX is constant or becomes larger. If the whole iteration times exceed a setting values, the Newton method is considered to be failure, and the integration step will be decreased so that the starting points for a new process of solving $G(X)=0$ with Newton method can become appropriate. The whole iteration of Newton method will be stopped until the ΔX is smaller than precision tolerance.



(a) Runaway

(b) Flat spot

(c) Cycle

Figure 10. Failure of Newton method

2.6.2 Gaussian-Seidel Method

Gaussian-Seidel method is a natural iterative method for solving nonlinear algebraic equations. The main idea of Gaussian-Seidel method is to solve the next step value of one variable explicitly by using the same values of last step values of other variables, or using the predicted value as initial values to iterate and correct. The predictor-corrector scheme is adopted by many commercial software of time-domain simulation, because it can make programming easy and conveniently deal with the non-linearity caused by non-linear elements (such as amplitude limiter) in power system dynamics. However, different from the Newton-Raphson method, Gaussian-Seidel method is not able make the convergent direction towards the optimum direction (slop), and therefore there will be more iteration steps and more computation in the whole convergent process.

One idea to develop the original Gaussian-Seidel method is to solve a small group of variables in one iterative step using the last step values of other variables. This idea is called Waveform Relaxation method based on Gauss-Jacobi relaxation, which has been discussed in many papers [18,19,20,21]. The main advantage of waveform relaxation method lies in that the nonlinear algebraic equations can be decoupled into several parts, where parallel computing can be implemented. Therefore, Waveform Relaxation may be competitive though the convergent direction of each variable is not optimal. Figure 11 shows the flow chart when Waveform Relaxation method is used in time domain simulation. HSET-TDS includes the Waveform Relaxation solver with parallel computing, which will be reported in future work.

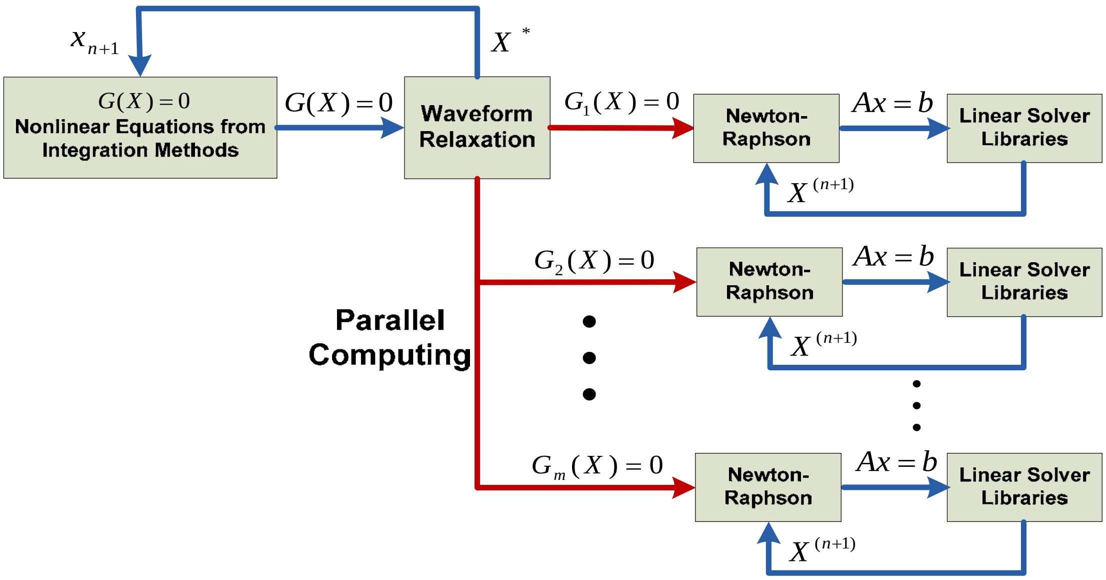


Figure 11. Waveform Relaxation method in time-domain simulation

2.7 Linear Solver Libraries

Gaussian Elimination is the basic method to solve a set of linear equations, and there are many available open source libraries, such as GMM++[22] , SuperLU[23] , for solving linear algebraic equations. Since much computation is involved during the process of solving linear equations in time-domain simulation, there are many papers on how to efficiently solve the linear equations to accelerate time-domain simulation by parallel computing, such Conjugate Gradient Method[24] [25] , or how to partition the system via linear equations, such as Block Bordered Diagonal Form(BBDF)[26] . The listed commercial software in Table 2 do not report what kind linear solver libraries they adopted to solve the linear equations. Chapter 6 will discuss the open source library of SuperLU, and corresponding comparison with other linear solver in GMM++ will be elaborated.

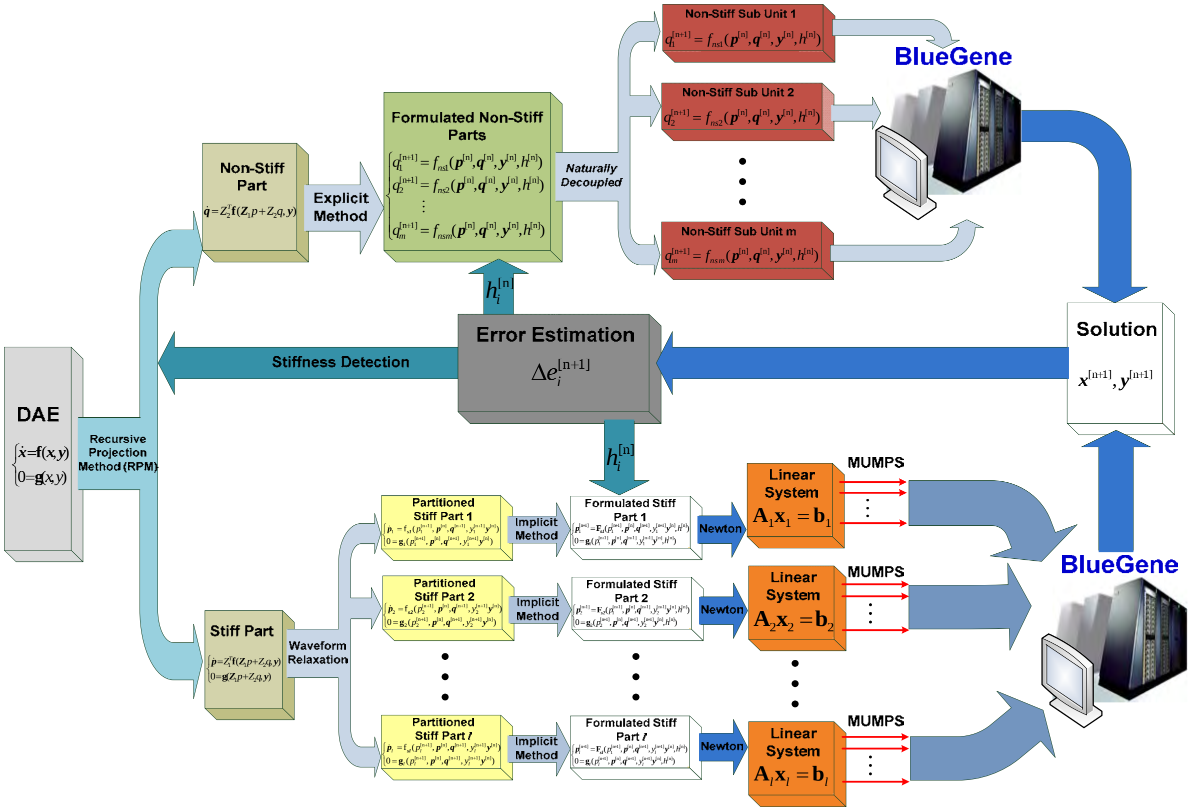


Figure 12. Parallel computing strategy of DAEs for Blue Gene/L

2.8 Overall Design for Parallel Computing On BlueGene/L

Figure 12 shows our overall design for high-speed execution of extended-term dynamic security assessment. The solution process divided into three distinct levels. The first level is the partition of DAEs into stiff part and non-stiff part by stiffness decoupling method, which will discussed in chapter 4. For the non-stiff part, the explicit method will be used to discretize the differential equations, which can be directly assigned into *m* processors since these formulated differential equations are naturally decoupled. The second level is the partition of the stiff part of the DAE where Waveform Relaxation Method (WRM) with Epsilon decomposition algorithm are used to continue to partition the stiff part of the DAE. Each partitioned part can be formulated by implicit method to ensure stability. The third level is the process of solution with a Newton-like method. Very dishonest Newton (VDHN) can be used to fix the Jacobian matrix constant for several steps, and Multi-frontal Massively Parallel sparse direct Solver (MUMPS) [27,28] will be adopted to solve the sparse linear equations. The final solution of one step will be integrated together, and the computational error will be estimated to control the stiffness detection and the step values for stiff part and non-stiff part.

There are 3 special cases of concern in the design of

Figure 12.

At some integration steps, the ODE parts may not be very stiff so that all differential equations can be solved by the explicit method, and thus the bottom branch of

Figure 12 need not be applied.

Even in the case where stiff parts are detected, it may be difficult to find an epsilon decomposition scheme that results in WRM convergence, and so in this case, the lower branch of

Figure 12 will be applied but without WRM; here, load balance can be maintained by appropriate parallelization via MUMPS.

Under the circumstance of failure in stiffness detection, the entire integration is done by WRM.

The ability to detect and respond to these special cases results in a design where the different partition methods and linear solution methods cooperate with each other to make the solution load of DAE at each processor balanced. A key feature of this design is that the DAE is divided into two different integration schemes, one for stiff and one for non-stiff parts, and then both parts are solved via a parallelized implementation. Numerical experiments will be reported in future work.

3. Generator Models and Validation of HSET

HSET-TDS is designed for fast dynamic simulation of power system, which involves with the numerical computation of a large set of differential equations shown in (2.1). With the given DAE system, the following chapters are going to introduce how to utilize different numerical technique to accelerate the solution of DAE system describing power system dynamics. One important issue before showing the simulation results is whether the basic DAE system is acceptable, and whether the simulation results is able to match with the results from commercial software such PSS/E when the same system is solved. In order to satisfy the requirement of the many different practical systems, the HSET-TDS has been developed to include the following ten different generator models[29, 30, 31] .

2nd order simplified Classic model, (just X_d')

2nd order Classic model (X_d' and X_q')

3rd order model (E_q')

4th order model (E_q' and E_d')

6th order model (E_q'' and E_d'')

6th order model considering generator speed dynamics in stator equations.

8th order model (modeling from perspective of flux, no simplification, and considering stator dynamics)

GENROU model ($E_d', E_q', \Psi_d', \Psi_q'$)

GENROU model ($E_d', E_q', \Psi_d', \Psi_q'$, and considering speed dynamics in stator equations)

GENROU model ($E_d', E_q', \Psi_d', \Psi_q'$, and considering stator dynamics)

However in this chapter, only two of the ten generator models above , namely the 6th order model and GENROU model are introduced. Simulation results are compared with those from PSS/E and presented..

3.1 Generator 6th order model in HSET-TDS (GEN6)

The basic Generator equations after Park transformation can be expressed as follows.

$$\begin{bmatrix} u_{dq0} \\ u_{fDQ} \end{bmatrix} = p \begin{bmatrix} \Psi_{dq0} \\ \Psi_{fDQ} \end{bmatrix} + \begin{bmatrix} S_{dq0} \\ 0 \end{bmatrix} + \begin{bmatrix} r_{dq0} & \\ & r_{fDQ} \end{bmatrix} \begin{bmatrix} -i_{dq0} \\ i_{fDQ} \end{bmatrix} \tag{3.1}$$

where $\boldsymbol{u_{dq0}} = \begin{bmatrix} u_d & u_q & u_0 \end{bmatrix}^T$ -- stator voltage after Park transformation

$\boldsymbol{u_{fDQ}} = \begin{bmatrix} u_f & u_D & u_Q \end{bmatrix}^T$ -- rotor voltage after Park transformation

$\boldsymbol{i}_{\boldsymbol{dq}0} = \left[\boldsymbol{i}_{\boldsymbol{d}} \quad \boldsymbol{i}_{\boldsymbol{q}} \quad \boldsymbol{i}_0\right]^{\boldsymbol{T}}$ -- stator current after Park transformation

$\boldsymbol{i}_{\boldsymbol{fDQ}} = \left[\boldsymbol{i}_{\boldsymbol{f}} \quad \boldsymbol{i}_{\boldsymbol{D}} \quad \boldsymbol{i}_{\boldsymbol{Q}}\right]^{\boldsymbol{T}}$ -- rotor current after Park transformation

$\Psi_{\boldsymbol{dq}0} = \left[\Psi_d \quad \Psi_q \quad \Psi_0\right]^T$ -- stator flux

$\Psi_{\boldsymbol{fDQ}} = \left[\Psi_d \quad \Psi_q \quad \Psi_0\right]^T$ -- rotor flux

$\boldsymbol{S}_{\boldsymbol{dq}0} = \left[-\omega\Psi_{\boldsymbol{q}} \quad \omega\Psi_{\boldsymbol{d}} \quad 0\right]^{\boldsymbol{T}}$

$\boldsymbol{r}_{\boldsymbol{dq}0} = diag\left[r_a \quad r_a \quad r_a\right]$

$\boldsymbol{r}_{\boldsymbol{fDQ}} = diag\left[r_f \quad r_D \quad r_Q\right]$

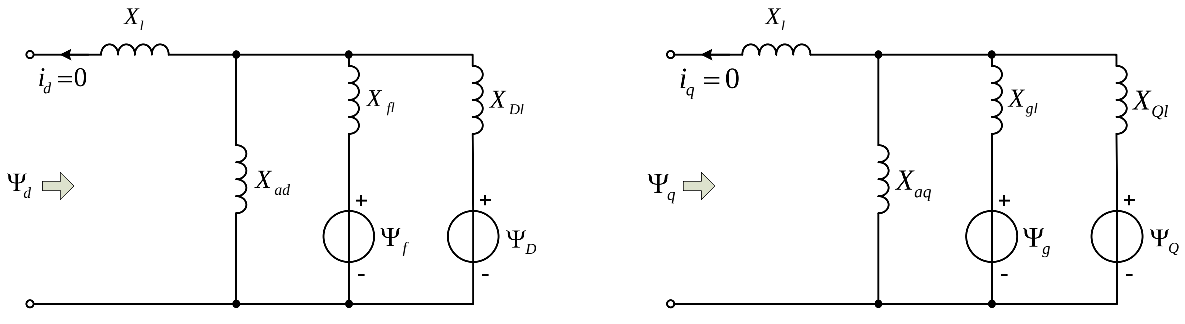
$p = \dfrac{d}{\omega_B dt(s)}$

The basic flux equations from d-axis and q-axis can be described as follows.

$$\begin{cases} \Psi_d = -X_d i_d + X_{ad} i_f + X_{ad} i_D \\ \Psi_f = -X_{ad} i_d + X_f i_f + X_{ad} i_D \\ \Psi_d = -X_{ad} i_d + X_{ad} i_f + X_D i_D \end{cases} \tag{3.2}$$

$$\begin{cases} \Psi_q = -X_Q i_Q + X_{aq} i_g + X_{aq} i_Q \\ \Psi_g = -X_{aq} i_Q + X_g i_g + X_{aq} i_Q \\ \Psi_Q = -X_{aq} i_Q + X_{aq} i_g + X_Q i_Q \end{cases} \tag{3.3}$$

When subtrasient process is considered in generator equations, E_q" and E_d" can be acquired according to the diagram shown in figure 13.



(a)Expression of Ψ_d for E_q" (b) Expression of Ψ_q for E_d"

Figure 13. Modeling Ψ_d and Ψ_q for E_q" and E_d"

We can define $E_q', E_d'\ , E_q'', E_d''$ according to [30] which can be shown as follows.

$$E_q' = \frac{X_f}{X_{ad}}\Psi_f \tag{3.4}$$

$$E_d' = -\frac{X_{aq}}{X_g}\Psi_g \tag{3.5}$$

$$E_q'' = \frac{X_{ad}}{X_f X_D - X_{ad}^2}\left(X_{Dl}\Psi_f + X_{fl}\Psi_D\right) \tag{3.6}$$

$$E_d'' = \frac{-X_{aq}}{X_g X_Q - X_{aq}^2}\left(X_{Ql}\Psi_g + X_{gl}\Psi_Q\right) \tag{3.7}$$

The time constant $T_{q0}', T_{d0}'\ , T_{q0}'', T_{d0}''$ can be expressed as follows.

$$T_{q0}' = \frac{X_g}{r_g} \tag{3.8}$$

$$T_{d0}' = \frac{X_f}{r_f} \tag{3.9}$$

$$T_{q0}'' = \frac{X_Q - \frac{X_{aq}^2}{X_g}}{r_Q} \tag{3.10}$$

$$T_{d0}'' = \frac{X_D - \frac{X_{ad}^2}{X_f}}{r_D} \tag{3.11}$$

The transient and sub transient inductance can be described as follows.

$$X_q' = X_l + X_{aq} \tag{3.12}$$

$$X_d' = X_l + X_{ad} \tag{3.13}$$

$$X_q' = X_l + X_{aq} \mathbin{//} X_{gl} \tag{3.14}$$

$$X_d' = X_l + X_{ad} \mathbin{//} X_{fl} \tag{3.15}$$

$$X_q'' = X_l + X_{aq} \mathbin{//} X_{Ql} \tag{3.16}$$

$$X_d'' = X_l + X_{ad} \mathbin{//} X_{fl} \mathbin{//} X_{Dl} \tag{3.17}$$

From the equations of flux on d-axis and q-axis shown in (3.6) and (3.7) and the rotor mechanical equations, we can get the 6 order differential equations and 2 stator voltage equations to describe synchronous generator.

$$\begin{cases} u_d = -\Psi_q - r_a i_d = E_d'' + X_q'' i_q - r_a i_d \\ u_q = \Psi_d - r_a i_q = E_q'' - X_d'' i_d - r_a i_q \end{cases} \tag{3.18}$$

$$T_{d0}'\, pE_q' = E_f - \frac{X_d - X_l}{X_d' - X_l}E_q' + \frac{X_d - X_d'}{X_d' - X_l}E_q'' - \frac{(X_d - X_d')(X_d'' - X_l)}{X_d' - X_l}i_d \tag{3.19}$$

$$T_{q0}'\, pE_d' = -\frac{X_q - X_l}{X_q' - X_l}E_d' + \frac{X_q - X_q'}{X_q' - X_l}E_d'' + \frac{(X_q - X_q')(X_q'' - X_l)}{X_q' - X_l}i_q \tag{3.20}$$

$$T_{d0}''\,pE_q'' = \frac{X_d''-X_l}{X_d'-X_l}T_{d0}''\,pE_q' - E_q'' + E_q' - (X_d' - X_d'')i_d \tag{3.21}$$

$$T_{q0}''\,pE_d'' = \frac{X_q''-X_l}{X_q'-X_l}T_{q0}''\,pE_d' - E_d'' + E_d' + (X_q' - X_q'')i_q \tag{3.22}$$

$$T_J\frac{d\omega}{dt} = T_m - \left[E_q''i_q + E_d''i_d - \left(X_d'' - X_q''\right)i_d i_q\right] - D(\omega - 1) \tag{3.23}$$

$$\frac{d\delta}{\omega_B dt} = \omega - 1 \tag{3.24}$$

where E_f and T_m are the voltage from excitation and the mechanical torque from prime turbine. In HSET-TDS, the available model for excitation and governor are IEEE1 excitation model as in PSS/E (or EXC-1A as in ETMSP) and GOV1 model as in PSS/E (or GOV-8 as in ETMSP) respectively [32] [33]. The description of these two model can be illustrate in Figure 14 and Figure 15. These two excitation and governor model can be embedded with all generator models in HSET-TDS if there are input of E_f and T_m. Besides, these two models are just for research grade use, which means that for the dynamic data from practical power system the dynamic data needs to be revised to match these two models. However, these two models are enough for the objective of comparison between HSET-TDS and commercial software such as PSS/E.

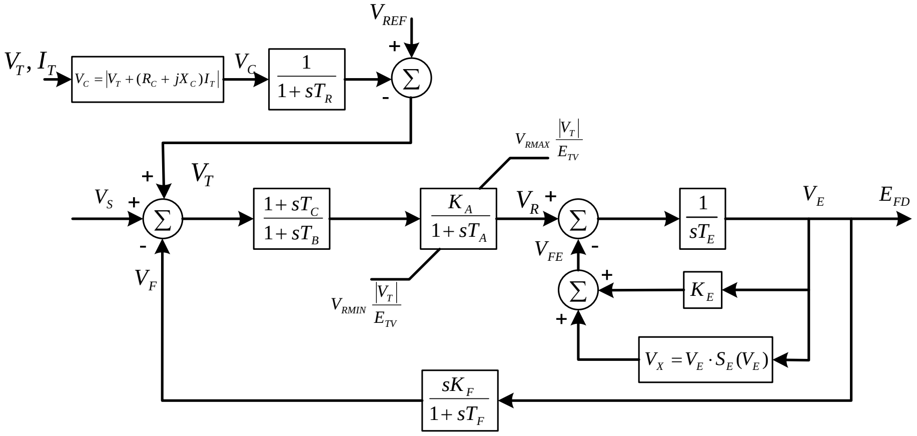


Figure 14. IEEE1 excitation model

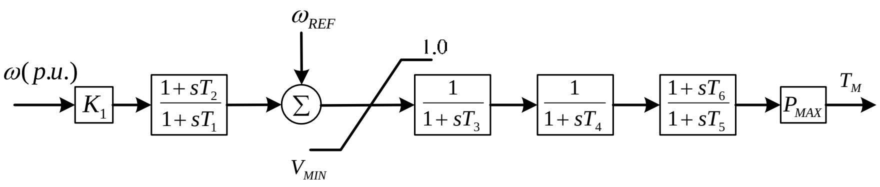
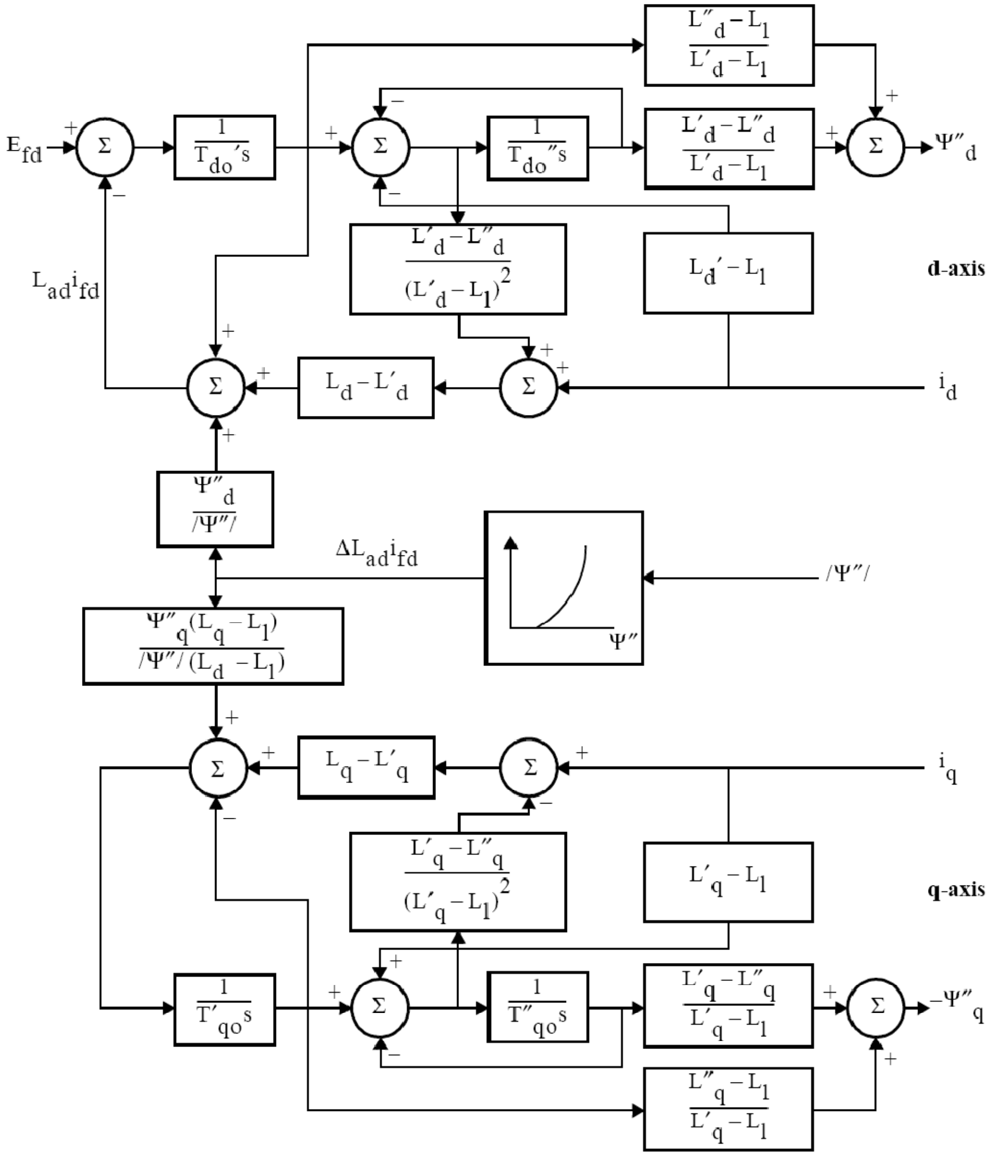


Figure 15. GOV1 excitation model

3.2 GENROU Model in HSET-TDS

GENROU [33] is a standard generator model, which is widely used in practical dynamic analysis of power system. According to the dynamics data from PJM in May of 2009 with around 10000 buses system, over 90% of the generators use GENROU model. The manual of PSS/E supplies the diagram about how to express Ψ_d" and Ψ_q" from d-axis and q-axis, which can be shown in Figure 16 from [33].



E_{fd}
Σ
$\frac{1}{T_{do}'s}$
$\frac{1}{T_{do}''s}$
$\frac{L''_d - L_l}{L'_d - L_l}$
$\frac{L'_d - L''_d}{L'_d - L_l}$
Ψ''_d
$\frac{L'_d - L''_d}{(L'_d - L_l)^2}$
$L_d' - L_l$
d-axis
$L_{ad}i_{fd}$
$L_d - L'_d$
i_d
$\frac{\Psi''_d}{/\Psi''/}$
$\Delta L_{ad}i_{fd}$
$/\Psi''/$
Ψ''
$\frac{\Psi''_q(L_q - L_l)}{/\Psi''/(L_d - L_l)}$
$L_q - L'_q$
i_q
$\frac{L'_q - L''_q}{(L'_q - L_l)^2}$
$L'_q - L_l$
q-axis
$\frac{1}{T'_{qo}s}$
$\frac{1}{T''_{qo}s}$
$\frac{L'_q - L''_q}{L'_q - L_l}$
$-\Psi''_q$
$\frac{L''_q - L_l}{L'_q - L_l}$
+
−

Figure 16. Electromagnetic model of Round Rotor Generator from PSS/E [33]

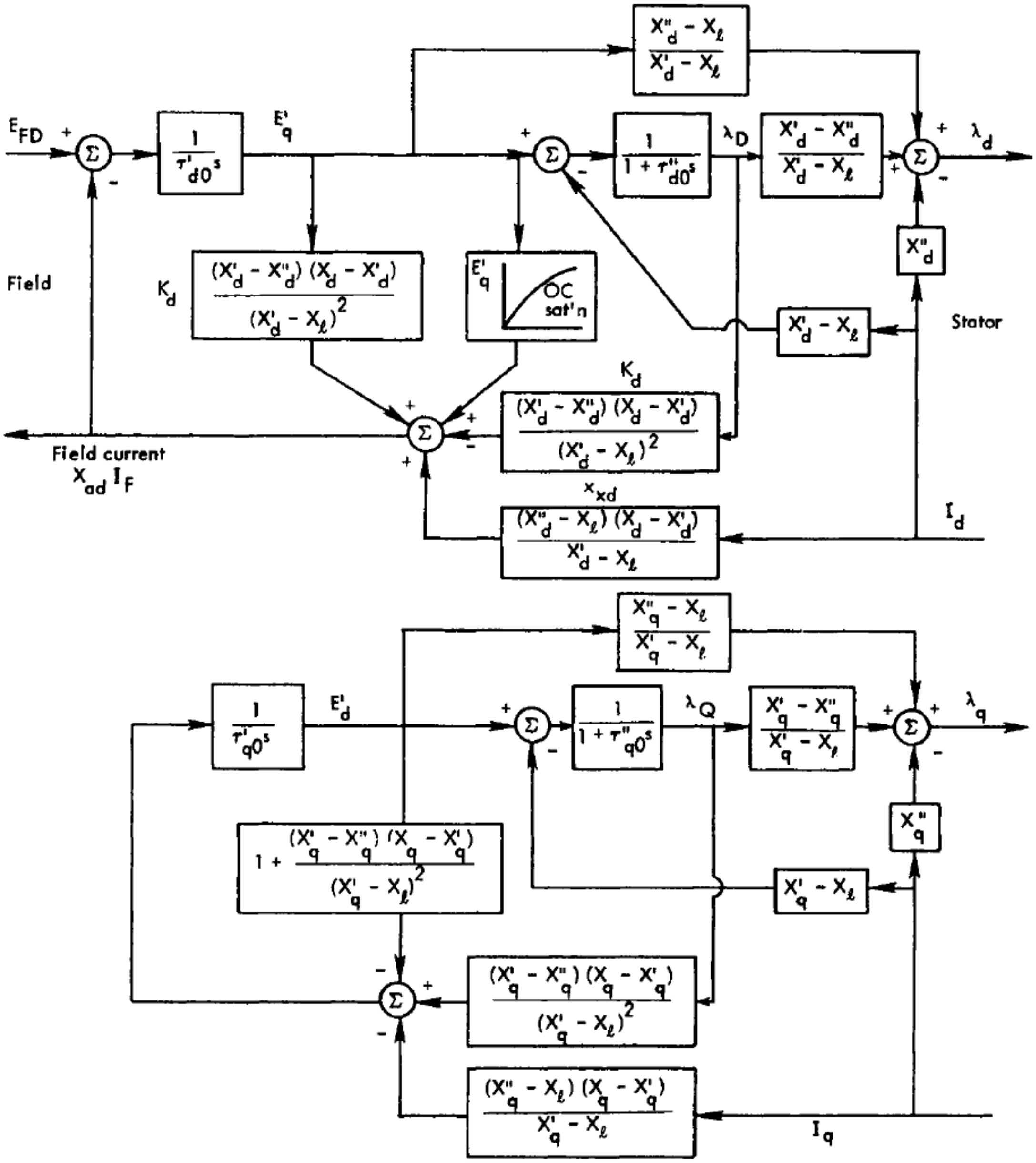

E_FD
Σ
1 / τ'_d0 s
E'_q
(X"_d - X_ℓ) / (X'_d - X_ℓ)
1 / (1 + τ"_d0 s)
λ_D
(X'_d - X"_d) / (X'_d - X_ℓ)
λ_d
X"_d
Field
K_d
(X'_d - X"_d) (X_d - X'_d) / (X'_d - X_ℓ)^2
E'_q
OC sat'n
X'_d - X_ℓ
Stator
K_d
(X'_d - X"_d) (X_d - X'_d) / (X'_d - X_ℓ)^2
Field current
X_ad I_F
x_xd
(X"_d - X_ℓ) (X_d - X'_d) / X'_d - X_ℓ
I_d
(X"_q - X_ℓ) / (X'_q - X_ℓ)
1 / τ'_q0 s
E'_d
1 / (1 + τ"_q0 s)
λ_Q
(X'_q - X"_q) / (X'_q - X_ℓ)
λ_q
1 + (X'_q - X"_q) (X_q - X'_q) / (X'_q - X_ℓ)^2
X"_q
X'_q - X_ℓ
(X'_q - X"_q) (X_q - X'_q) / (X'_q - X_ℓ)^2
(X"_q - X_ℓ) (X_q - X'_q) / X'_q - X_ℓ
I_q

Figure 17. Voltage behind subtransient reactance model [34]

[34] supplies similar diagram of GENROU which is shown in Figure 17. It can be found that these two diagram is basically the same with respect to the differential part but different with respect to saturation. In HSET-TDS, the GENROU model does not consider saturation currently, and the differential parts follow the diagram in Figure 17. The mathematical equations about E_q', E_d' , E_q'', E_d''can be described as follows:

$$\begin{aligned} T_{d0}'\frac{dE_q'}{dt} &= E_f + \frac{(X_d - X_d')(X_d' - X_d'')}{\left(X_d' - X_l\right)^2}E_q'' - \frac{(X_d - X_d')(X_d'' - X_l)}{\left(X_d' - X_l\right)}i_d \\ &\quad - \left[\frac{(X_d - X_d')(X_d - X_d')}{\left(X_d' - X_l\right)^2} + 1\right]E_q' \end{aligned} \tag{3.24}$$

$$\begin{aligned} T_{q0}'\frac{dE_d'}{dt} &= \frac{(X_q - X_q'')(X_q - X_q')}{\left(X_q' - X_l\right)^2}E_d'' - \frac{(X_d - X_d')(X_d'' - X_l)}{\left(X_q' - X_l\right)}i_q \\ &\quad - \left[\frac{(X_q - X_q')(X_q - X_q')}{\left(X_q' - X_l\right)^2} + 1\right]E_d' \end{aligned} \tag{3.25}$$

$$T_{d0}''\frac{dE_q''}{dt} = E_q' - (X_d' - X_l)i_d - E_q'' \tag{3.26}$$

$$T_{q0}''\frac{dE_d''}{dt} = E_d' + (X_q' - X_l)i_q - E_d'' \tag{3.27}$$

$$T_J\frac{d\omega}{dt} = T_m - \left(\Psi_d i_q - \Psi_q i_d\right) - D(\omega - 1) \tag{3.28}$$

$$\frac{d\delta}{\omega_B dt} = \omega - 1 \tag{3.29}$$

$$\Psi_d = \frac{X_d'' - X_l}{X_d' - X_l}E_q' + \frac{X_d' - X_d''}{X_d' - X_l}E_q'' - i_d X_d'' \tag{3.30}$$

$$\Psi_q = \frac{X_q'' - X_l}{X_q' - X_l}E_d' + \frac{X_q' - X_q''}{X_q' - X_l}E_d'' - i_q X_q'' \tag{3.31}$$

$$u_d = -r_a i_d - \Psi_q \tag{3.32}$$

$$u_q = -r_a i_q + \Psi_d \tag{3.33}$$

Similarly with the 6th order model, E_f and T_m in GENROU can be embedded with the IEEE1 or GOV1 model. Besides, these two values can be set to constant depending on different situation. As far as the stator equations in the GENROU model described above are concerned, it can be found that there are some difference compared with stator expression (3.1). There are two assumptions about the stator equations

The stator dynamics is not included, which means $\frac{d\Psi_d}{dt}$ and $\frac{d\Psi_q}{dt}$ are too small to be involved. In the process of disturbance, the speed of generator rotor is around its rated value, and then $\omega \approx 1$.

The first assumption is acceptable since $\left|\frac{d\Psi}{dt}\right|$ is much less than $|\omega\Psi|$ in practical case, and therefore the stator equation becomes to be linear equations. The second assumption is acceptable, since there are many control unit to render rotor speed around rating value such that the system frequency can be guaranteed around rated values. In HSET-TDS, there are auxiliary model about GENROU, which includes the dynamics of stator or variable rotor speed in stator equations. The comparison and analysis with report to this issue is going to be shown in the future report.

3.3 Validation of HSET-TDS by PSS/E

In this section three cases are adopted to make the comparison between HSET-TDS and PSS/E. The first case is the New England 39 buses and 10 generator system, and the second one is the expanded system of 39 buses system. In HSET-TDS, initial values of DAE system is calculated by power flow, which is based on Newton-Raphson method. There are differential equations, algebraic equations and variables. The integration methods used in the cased is variable-step Trapezoidal rule, which will be discussed in next chapter. The nonlinear solver is based on Newton method, and SuperLU library is chosen as linear solver.

3.3.1 Case of New England 39 buses, 10 gen system

IEEE New England 39 Bus 10 Gen system is a simplified system, which can be shown in Figure 18. In HSET-TDS, GEN6 and GENROU are adopted for simulation, while in PSS/E GENROU is adopted. E_f and T_m are assumed to be constant in each software. The event of the case is selected to be bus fault on bus 17 starting from 0.5s and lasting for 0.1s. The voltage of bus 37 and the speed of G8 will be monitored. The whole simulation lasts for 10 second.

Figure 20 shows the simulation results from HSET-TDS and PSS/E.

3.3.2 Case of expanded 3900 bus system

In order to verify the ability of HSET-TDS to solve large systems, a large system is constructed from the basic 39 bus system. The idea of expanding New England 39 buses system is to copy the systems for several times and then connect the buses within each system. To guarantee the stability of large system constructed, the transmission lines connecting each 39 buses system are of small impedance. This constructing approach is similar to the way of connecting large system nowadays utilizing ultra high voltage transmission lines.

In this case, the 39 buses system is going to be copied for 100 times according to the mesh shown in Figure 19. Bus 2, 9, 23, 29 are going to connected to adjacent systems. The new large system contains 3900 buses, 1000 generators and 4960 lines. In the DAE system, there are 6000 differential equations, 18000 algebraic equations and 24000 variables when Gen6 is adopted.

The event of the case is selected to be bus fault on bus 17 starting from 0.5s and lasting for 0.1s. The voltage of bus 37 and the speed of G8 will be monitored. The whole simulation lasts for 10 second.

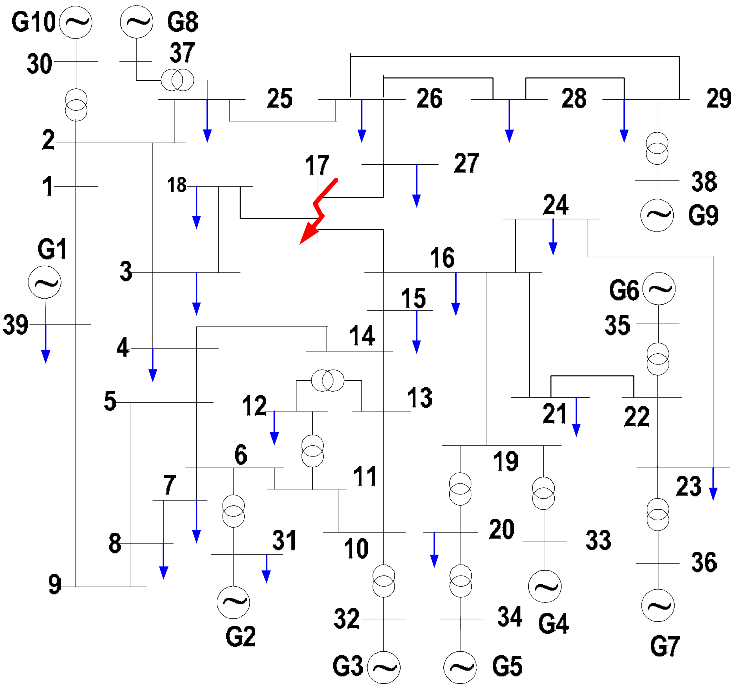
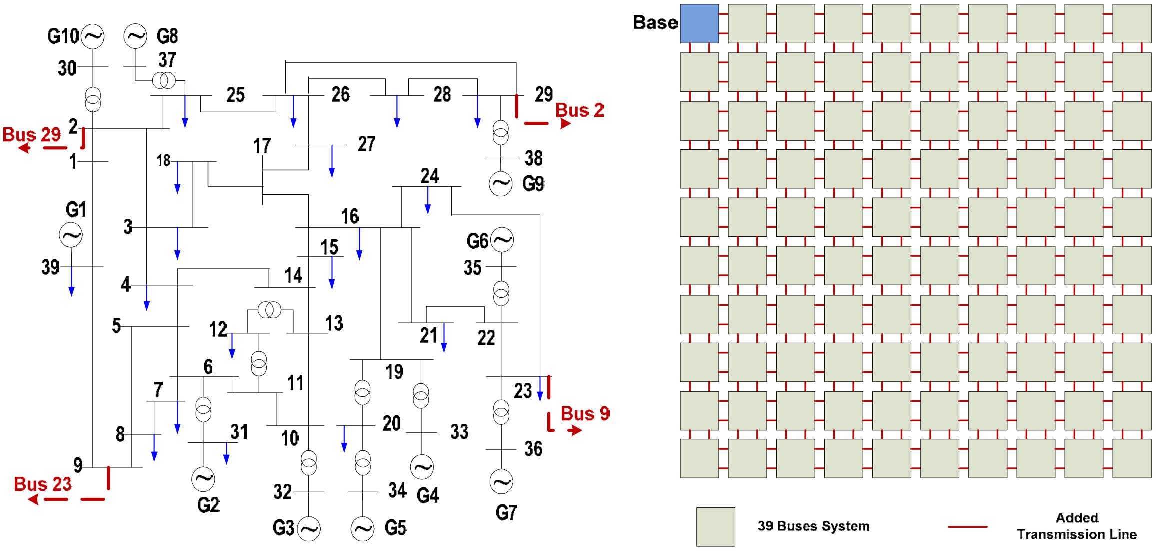


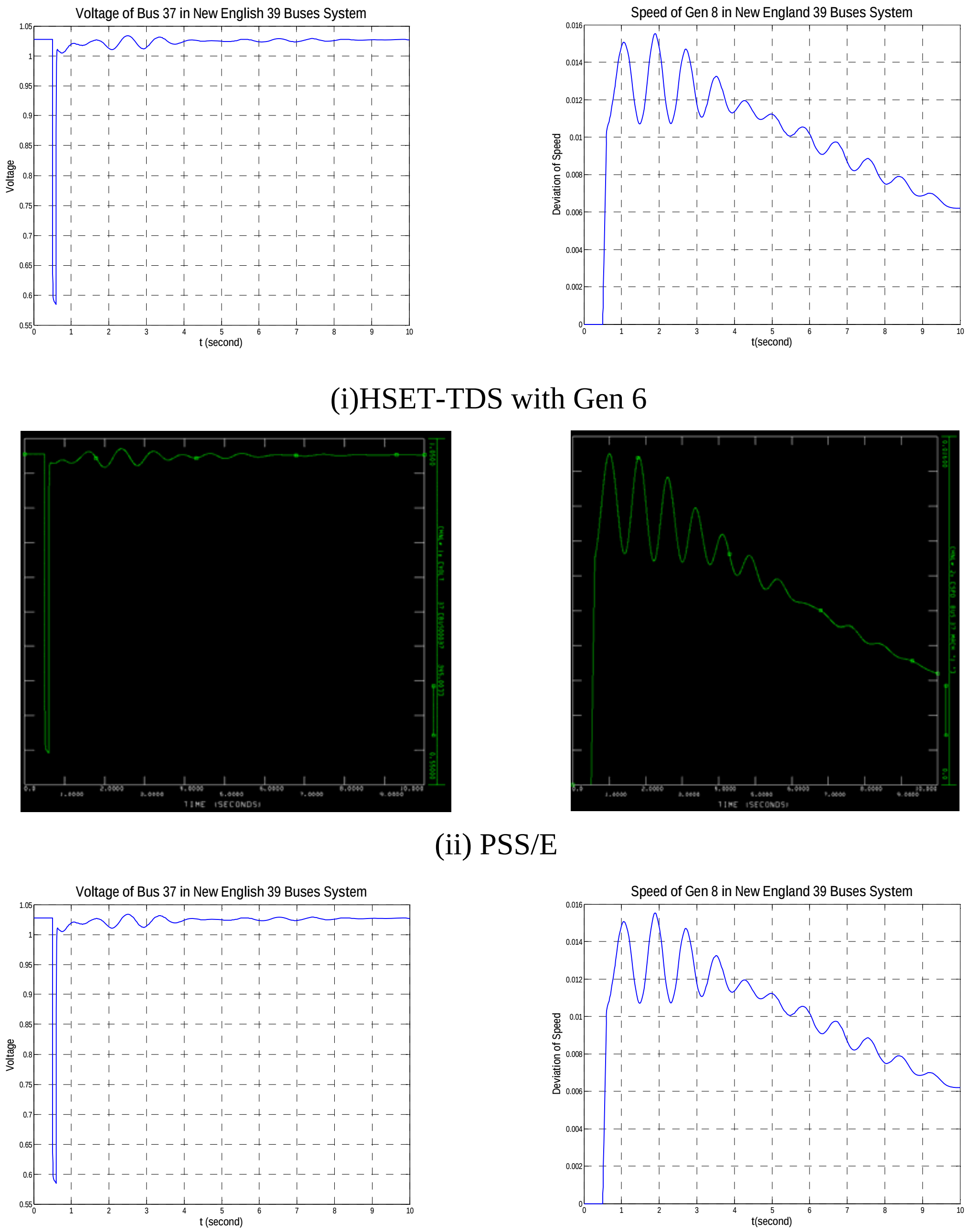
Figure 18. New England 39 buses



(a)Original 39 buses system

(b)Expanding mesh

Figure 19. Expanded system from New England 39 bus system (10×10)



(i)HSET-TDS with Gen 6

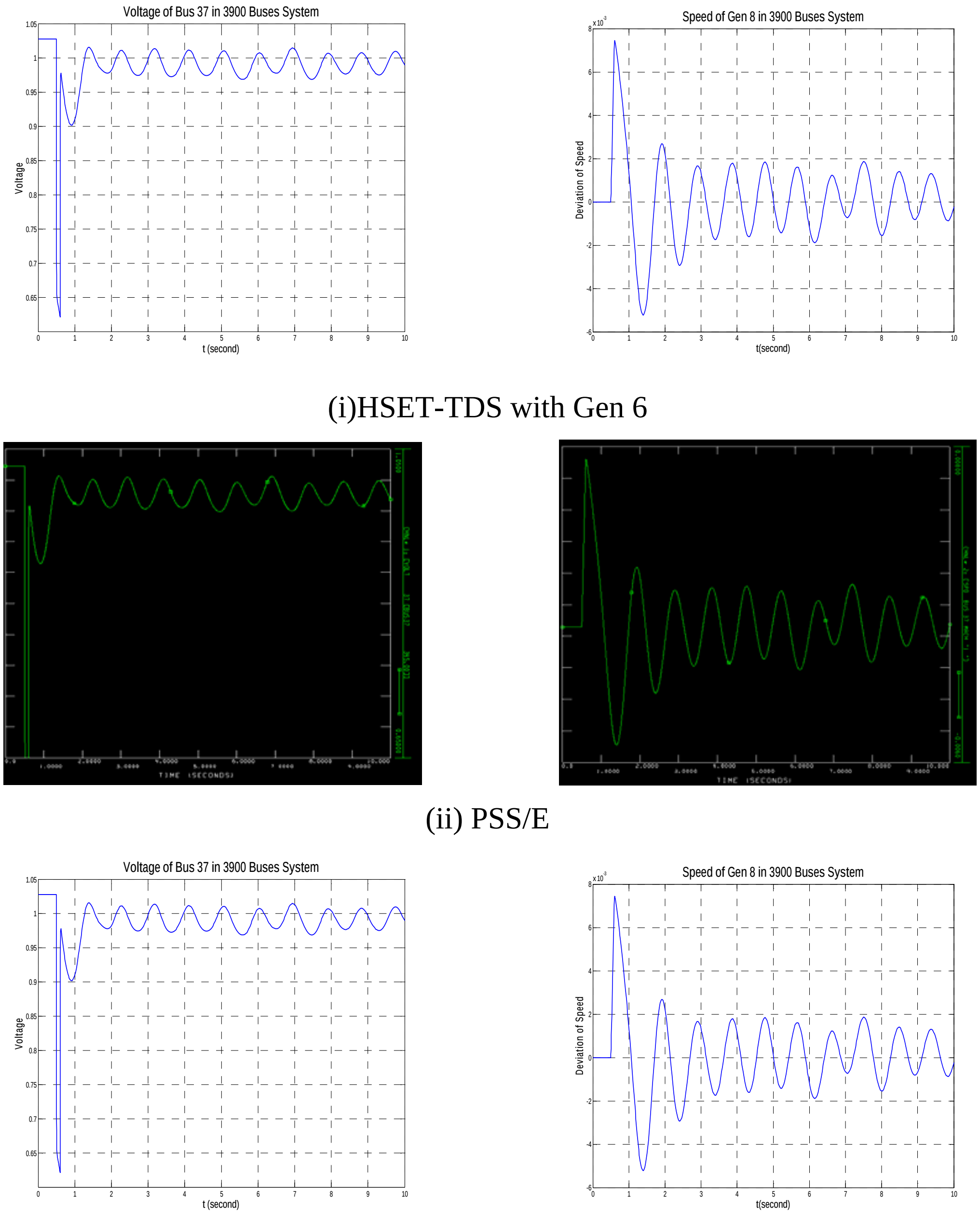
(ii) PSS/E

(iii) HSET-TDS with GENROU

(a) Voltage of Bus37

(b) Deviation of Speed on Gen8

Figure 20. Simulation results of New England 39 bus system



(i)HSET-TDS with Gen 6

(ii) PSS/E

(iii) HSET-TDS with GENROU

(a) Voltage of Bus37 (b) Deviation of Speed on Gen8

Figure 21. Simulation results of expanded 3900 bus system

Figure 21 shows the simulation results from HSET-TDS and PSS/E. From the simulation results in

Figure 20 and

Figure 21, it can be seen that the simulation results with the generator model of Gen6 and GENROU in HSET-TDS are totally the same. However, compared with GEN6 model since there is two more algebraic variables in GENROU which are Ψ_d and Ψ_q, there are more algebraic equations in DAE system when GENROU model are used. Furthermore, it can be seen that the simulation results from HSET-TDS are nearly the same as those from PSS/E. The validation of HSET-TDS is able to make the simulation results in the following chapters more convincing and trustable.

4. Hammer-Hollingsworth 4 (HH4) Formula

In this Chapter, a new integration method, Hammer-Hollingsworth 4 (HH4), will be introduced. The motivation of exploring the new integration method lies in that the traditional integration method adopted by most of commercial software is Trapezoidal rule, which is able to deal with not only stiffness problem but also hyper stability problem. However, due to that fact that the Trapezoidal rule is of second order precision, the integration step need to be small enough to guarantee the precision. We try to find a new integration method, which shares the same stability attribute with Trapezoidal rule but has advantage of higher precision than Trapezoidal rule.

4.1 Traditional Integration Methods for Power System Simulation

Discretization with numerical integration methods is a crucial process in solving any differential equations. A suitable integration methods for time-domain simulation of power system need to possess attributes which can cope with two problems, 1) stiffness problem and 2) hyper-stability problem. Since there are many various electric elements (generators, excitation, governor, PSS, ect.) with different time constants in power system, the DAE system describing power system can be stiff[14, 15, 16]. Stiffness problem can lead to large error in simulation or even make some integration methods like explicit methods not work [35]. On the other hand, [10] reported the phenomenon of hyper-stability of some numerical methods, which can make a really unstable system simulated as a stable one. Because the main goal of time-domain simulation is to check the stability of power system, the ability to deal with hyper-stability problem can be a criterion for a suitable integration method for time-domain simulation.

Since the analysis of stiffness problem and hyper-stability problem involves with numerical stability analysis of integration methods, we will focus on the stability analysis of several integration methods used in current commercial software in this section, Trapezoidal Rule, Theta-method, Adams method and BDF. Besides, the local truncation error estimation of these methods is analyzed.

4.1.1 Trapezoidal rule

Trapezoidal rule is a second order implicit integration method which is popularly adopted in many software such as BPA and PSS/E. For the ODE system

$$\begin{cases} \dot{x} = f(x) \\ x(t_0) = x_0 \end{cases} \tag{4.1}$$

the trapezoidal rule can be described as

$$x_{n+1} = x_n + \frac{h}{2}\left[f(x_n) + f(x_{n+1}) \right] \tag{4.2}$$

where h is integration step.

For the DAE system of (2.1), the Trapezoidal rule can be described as

$$\begin{cases} x_{n+1} = x_n + \frac{h}{2}\left[\mathbf{f}(x_n, y_n) + \mathbf{f}(x_{n+1}, y_{n+1})\right] \\ \quad 0 = \mathbf{g}(x_{n+1}, y_{n+1}) \end{cases} \tag{4.3}$$

We will discuss trapezoidal rule from two aspects, numerical stability and numerical precision.

4.1.2 Numerical stability of trapezoidal rule

Assume that the general form of a differential equation is $\dot{x} = f(t,x)$. We linearize f in its neighborhood as follows.

$$\dot{x} = f(t_n, x_n) + (t - t_n)\frac{\partial f}{\partial t}\bigg|_{(t_n, x_n)} + (x - x_n)\frac{\partial f}{\partial x}\bigg|_{(t_n, x_n)} + \cdots \tag{4.4}$$

The higher order sterms can be omitted. Let $\lambda = \frac{\partial f}{\partial x}\Big|_{(t_n, x_n)}$, and we can get

$$\dot{x} = \lambda x + f(t_n, x_n) + (t - t_n)\frac{\partial f}{\partial t}\bigg|_{(t_n, x_n)} - \lambda x_n \tag{4.5}$$

If $\lambda \neq 0$, we can do transformation $\overline{x} = x + \frac{1}{\lambda}\left[f(t_n, x_n) + (t - t_n + \frac{1}{\lambda})\frac{\partial f}{\partial t}\Big|_{(t_n, x_n)} - \lambda x_n \right]$,

Thus (1) can be transformed to

$$\dot{\overline{x}} = \lambda \overline{x} \tag{4.6}$$

Expression (3.6) is usually called test equation (see the definition as follows). For a set of differential equations, λ_i are the eigenvalues of Jacobian matrix.

Now we apply a numerical method to expression (3.6), for example Forward Euler method.

$$x_{n+1} = x_n + h\lambda x_n = (1 + h\lambda)x_n = R(h\lambda)x_n = R(z)x_n \tag{4.7}$$

where $z = h\lambda$. Assume that there is disturbance δ_n on x_n, and the resulting disturbance on x_{n+1} is δ_{n+1}.

Then, $\delta_{n+1} = R(z)\delta_n$. If we want $|\delta_{n+1}| \leq |\delta_n|$, we just need $|R(z)| \leq 1$.

Definition [35]: The function $R(z)$ is called the ***stability function*** of the method. It can be interpreted as the numerical solution after one step for

$$\dot{x} = \lambda x, \qquad \text{with } x_0 = 1, \quad z = h\lambda,$$

the famous Dahlquist test equation. The set

$$S = \left\{ z \in \mathbf{C}; \quad |R(z)| \leq 1 \right\}$$

is called the ***stability domain*** of the method.

It can be seen that the stability function of Forward Euler method is $R(z)=1+z$, which is a unit circle in complex plane shown in Figure 22. If we apply trapezoidal rule (3) into Dahlquist test equation, we can acquire the stability function of trapezoidal rule, which is

$$R(z)=\frac{1+z/2}{1-z/2} \tag{4.8}$$

The stability domain of trapezoidal rule can be shown in Figure 22, which is the whole left part of complex plane. Trapezoidal rule is able to deal with stiffness problem, since it possesses the attribute of A-Stability. A method whose stability domain includes the whole left plane is called A-Stable [35]. If an explicit method doesn't work for an ODE system and *z* is out of stability domain due to either too large integration step or too large eigenvalues (for example the point B in the Figure 22), trapezoidal rule is still able to make the numerical error convergent because *z* is still in the left part of complex plane. However, it does not mean that we can select any large integration step for simulation due to the fact that trapezoidal rule can make the numerical error convergent when *z* is in the left part of complex plane. The global truncation error is still dependent on the local truncation error. Additionally, if an eigenvalue is positive and *z* is on the right part of complex plane, the whole system is unstable and the numerical error will be divergent. In this case, the simulation results will depend on the numerical precision of the integration method. The point C in Figure 22 shows this scenario, and we can find that trapezoidal rule is able to give an unstable result since $|R(z)|>1$ and $|x_{n+1}|>|x_n|$.

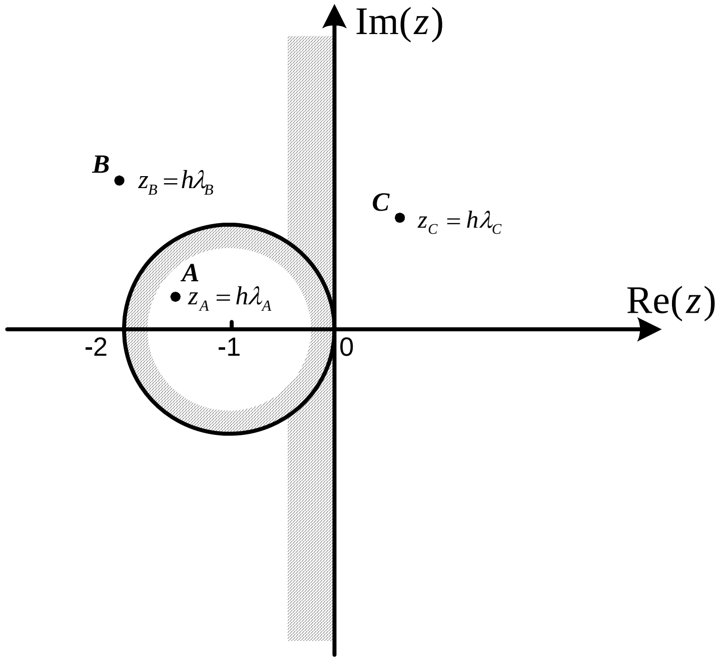


Figure 22. Stability domain of Trapezoidal Rule and Forward Euler

4.1.3 Numerical precision of trapezoidal rule

For the ODE system (3.1), the exact value of the next step can be expressed as follows according to Taylor expansion.

$$x_{n+1} = x(t_n + h) = x_n + hf(x_n) + \frac{h^2}{2!} f'(x_n) f(x_n) + \frac{h^3}{3!} \begin{bmatrix} f''(x_n) f^2(x_n) \\ + f^{2}{}'(x_n) f(x_n) \end{bmatrix} + O(h^4) \tag{4.9}$$

$f(x_{n+1})$ in expression (3.1) can be expanded by Tylor expansion, and the trapezoidal rule can be written as

$$x_{n+1} = x_n + hf(x_n) + \frac{h^2}{2!} f'(x_n) f(x_n) + \frac{h^3}{4}\left[f''(x_n) f^2(x_n) + f^{2}{}'(x_n) f(x_n)\right] + O(h^4) \tag{4.10}$$

Then the local truncation error of trapezoidal rule is

$$\left|E(x_{n+1})\right| = \left|\frac{h^3}{12}\left[f''(x_n) f^2(x_n) + f^{2}{}'(x_n) f(x_n)\right] + O(h^4)\right| \tag{4.11}$$

It can be seen that trapezoidal rule can guarantee the precision h^2 for simulation results.

4.1.4 Theta method

There are two kinds of forms of theta method. For the ODE system (3.1), the first form of theta method [36] can be described as

$$x_{n+1} = x_n + hf\left[(1-\theta)x_n + \theta\, x_{n+1}\right] \tag{4.12}$$

where θ is a parameter with $0 \le \theta \le 1$.

The second form of theta method [9, 37, 38, 39] can be described as

$$x_{n+1} = x_n + h\left[(1-\theta) f(x_n) + \theta\, f(x_{n+1})\right]. \tag{4.13}$$

where θ is a parameter with $0 \le \theta \le 1$.

The second form of theta method is adopted in EXTAB, and we will focus on the analysis of stability and precision of 2nd-form theta method.

4.1.5 Numerical stability of theta method

Apply the theta method (3.13) into Dahlquist test equation, and we can acquire its stability function, which is

$$R(z) = \frac{1+(1-\theta)z}{1-\theta z} \tag{4.14}$$

Thus, the stability domain of theta method (11) is $\boldsymbol{S}_{\boldsymbol{theta-M}} = \left\{ \boldsymbol{z} \in \mathbf{C}; \quad \left| \frac{1+(1-\theta)\boldsymbol{z}}{1-\theta\,\boldsymbol{z}} \right| \le 1 \right\}$

There are following special cases when $\theta = 0.5$, $\theta = 0$, $\theta = 1$.

If $\theta = 0.5$, the theta method becomes to be trapezoidal rule, and the stability domain is the whole left part of complex plain.

If $\theta = 0$, the theta method becomes to be Forward Euler method, and the stability domain is a unit circle in the left part of complex plain.

If $\theta = 1$, the theta method becomes to be Backward Euler method, and the stability domain is the whole complex plain except the unit circle in the right part of complex plain.

If $\theta \ne 0.5$, then the stability domain of theta method (11) can be written as

$$S_{theta-M} = \left\{ z \in \mathbf{C}; \quad \left| z + \frac{1}{1-2\theta} \right| \le \left| \frac{1}{1-2\theta} \right| \right\}$$

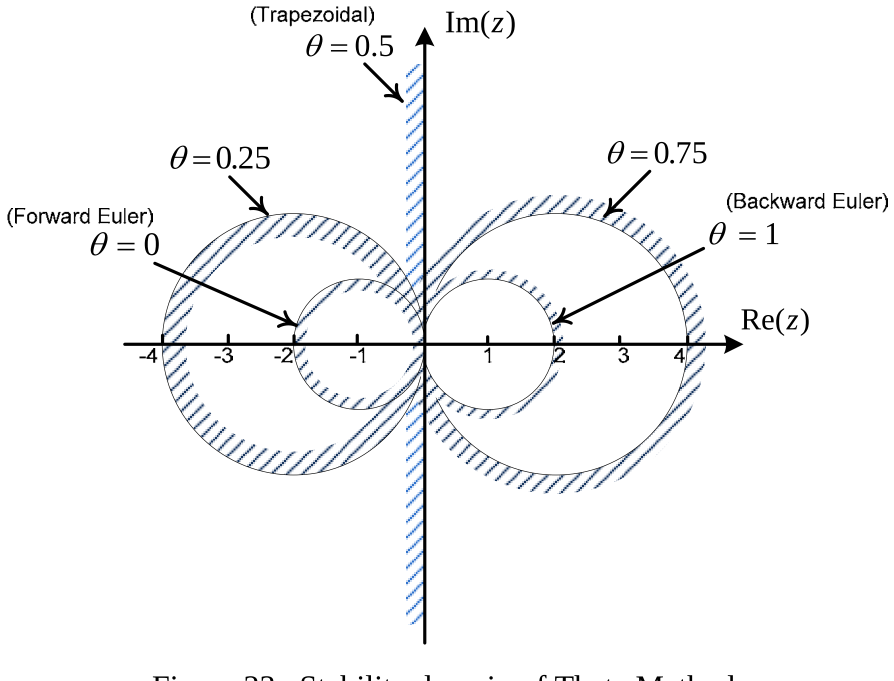


Figure 23. Stability domain of Theta Method

The Figure 23 shows the stability domain when θ is 0, 0.25, 0.5, 0.75, and 1. It can be found that the theta method lose A-stability when $\theta < 0.5$. Similar with the case of explicit method like Forward Euler method, the theta method ($\theta < 0.5$) is not suitable for time domain simulation because integration step has to small enough to make *z* within the stability domain if there is a big eigenvalue. When $\theta > 0.5$, it can be found that the stability domain contains the whole left part of complex plane and some area of right part of complex plane, thus the theta method ($\theta > 0.5$) is A-stable. However, the area of stability domain in right part of complex plane can lead to hyper-stability problem.

Hyper-stability problem is that the simulation result shows a stable one while actually the ODE system is unstable. An example can be shown to illustrate the phenomenon. Consider the following ODE system,

$$\begin{cases} \dot{x}_1 = 100x_1 - 400x_2 \\ \dot{x}_2 = 100x_1 + 100x_2 \end{cases} \qquad (4.15)$$
$$x_1(0) = 2,\ x_2(0) = 2$$

ODE system (3.15) is a linear system, with two eigenvalues $100 \pm 200j$. According to Lyapunov stability theory, system (3.15) is unstable. Figure 24 shows the simulation results of x_1 , x_2 by the numerical method of theta method ($\theta = 0$, also Backward Euler) , and the corresponding points of *z*. It can be found that when $h = 10^{-4}$, $z_1 = \lambda_1 h = 0.01 + 0.02j$, which is outside the stability domain, the simulation result is divergent and close to exact solution, which is simulated with $h = 10^{-5}$. However, when $h = 0.005$, $z_1 = \lambda_1 h = 0.5 + j$, which is within the stability domain, the simulation results turns out to be convergent.

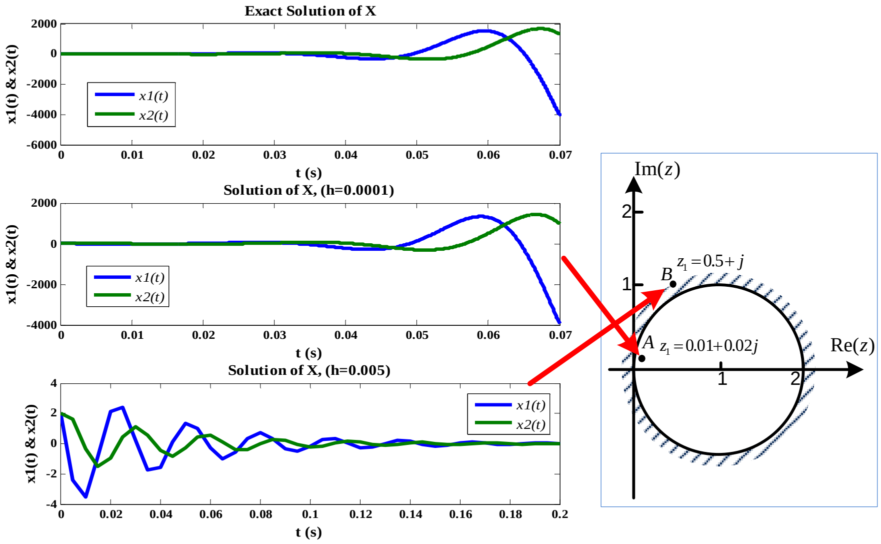


Figure 24. An example about Hyper-Stability problem

The hyper-stability problem is destructive for time-domain simulation, and possibly leads to large numerical error when simulating an unstable case due to numerical integration method with strong convergent ability, such as Backward Euler method. The reason of hyper-stability problem can be investigated from point of the stability function. If Backward Euler method is used to solve an unstable ODE system, we can find that $|R(z)| = \left|\frac{1}{1-z}\right| < 1$ if z is out of the unit circle in the right part of complex plane, and the method is able to make numerical error convergent such that $|\delta_{n+1}| \le |\delta_n|$. However, the method also makes $|x_{n+1}| \le |x_n|$, while actually for the exact solution, $|x_{n+1}| > |x_n|$ because the ODE system has positive eigenvalues. Paper [40] tried to improve Trapezoidal Rule with damping term, and the new trapezoidal rule is another expression of theta method. The damping term can make $\theta > 0.5$, and thus make the simulation avoid numerical oscillation due to the pure imaginative eigenvalues which can make $|R(z)| = 1$. Paper [9] provides a suggestive value of theta, $\theta = 0.53$ corresponding the formula (3.13), which have same effect of avoiding numerical oscillation. However, the Figure 25 shows that there are still some area in the right part of complex, which may lead to hyper-stability problem theoretically. Additionally, there will be precision changes when $\theta \ne 0.5$. The discussion on precision of theta method is shown below.

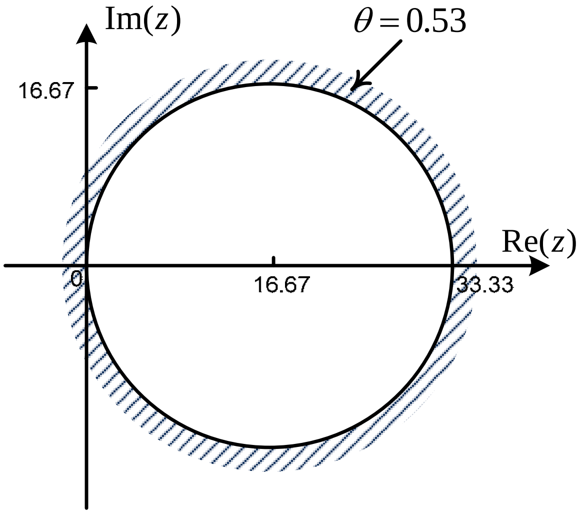


Figure 25. Stability domain of Theta method ($\theta = 0.53$)

4.1.6 Numerical precision of theta method

Similar with the analysis of Trapezoidal rule, $f(x_{n+1})$ in expression (11) can be expanded by Tylor expansion, and theta method (11) can be written as

$$x_{n+1} = x_n + hf(x_n) + \theta h^2 f'(x_n) f(x_n) + h^3 \begin{bmatrix} \frac{\theta}{2} f''(x_n) f^2(x_n) \\ + \frac{\theta}{2} f^2{}'(x_n) f(x_n) \end{bmatrix} + O(h^4) \tag{4.16}$$

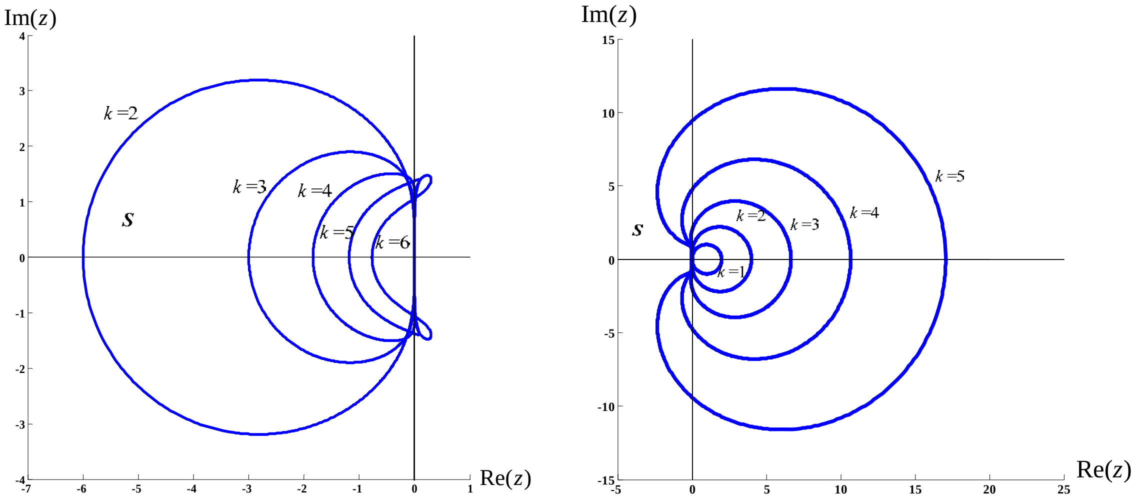
Compared with the exact x_{n+1} from (7), the local truncation error of theta method is

$$\left|E(x_{n+1})\right| = \left| \left(\frac{1}{2}-\theta\right)h^2 f'(x_n)f(x_n) + h^3 \left[\begin{array}{l} \left(\frac{1}{6}-\frac{\theta}{2}\right) f''(x_n)f^2(x_n) \\ +\left(\frac{1}{6}-\frac{\theta}{2}\right) f^{2}{}'(x_n)f(x_n) \end{array}\right] + O(h^4) \right| \tag{4.17}$$

From the local truncation error term (10) of theta method, if can be found that the theta method will lose second order precision if $\theta \neq 0.5$. From this point, the trapezoidal rule, which is the theta method with $\theta = 0.5$, is more precise.

4.1.7 Adams method and BDF

The methods of explicit Adams, implicit Adams and BDF are multi-steps integration methods for solving differential equations. EUROSTAG adopts a method combined with Adams and BDF for time domain simulation. From the formula of Adams and BDF methods, the first order of implicit Adams is Trapezoidal rule, which is the main formula used by EUROSTAG. [35] supplies the stability domain of implicit Adam and BDF, which is shown in Figure 26. It is be seen that the stability domain of Adams method becomes smaller and smaller as the order is increased, and therefore Adams method (order larger than 1) is not suitable for time domain simulation of power system. As far as BDF method is concerned, hyper-stability may possibly happen, and thus it is not suitable for power system either, which has been discussed in paper [10].



(a) Implicit Adam (b) BDF

Figure 26. Stability domain of Implicit Adam and BDF (*k* is order)

4.2 Hammer-Hollingsworth 4 (HH4) Formula

From the discussion on traditional integration methods in time domain simulation, trapezoidal rule is a suitable integration method, which is adopted by most of commercial software. The main advantage of trapezoidal rule lies in that it can effective deal with stiffness and hyper-stability problem, which means when the system is stable or unstable, the integration methods is able to provide correct stimulation results. In order to accelerate the efficiency of time domain simulation with trapezoidal rule or theta method (when parameter theta is around 0.5), a practical technique is variable step integration [35, 36], which can adjust the integration step from the changes of simulation results. When the simulation results change slowly, the integration step can be increased, and thus the simulation efficiency can be developed; while when the simulation results change fast, the integration step needs to be decreased, and thus the simulation precision can be guaranteed. However, the integration step for trapezoidal rule is always limited by its precision h^2. If we can make the local truncation error shrunk, the integration step can be continued to increase and simulation times can be decreased. Therefore, the whole efficiency of time domain simulation can be enhanced.

In this section, an integration method, Hammer-Hollingsworth 4 [36,41], based on Quadratic functions is discussed. Hammer-Hollingsworth 4, which is a fourth order implicit Runge-Kutta method, was first introduced in paper [41]. Of all different kinds of explicit and implicit Runge-Kutta methods, Hammer-Hollingsworth 4 posesses distinct attributes from other methods, 1) A-stability and same stability domain as trapezoidal rule, 2) higher precision than trapezoidal rule. HH4 is one of the implicit integration methods used in HSET-TDS, and it is expected to have better performance than trapezoidal rule.

4.2.1 Formula of HH4

For ODE system (2), [35] introduce an expression of the Hammer-Hollingsworth 4 integration method which can be described as

$$\begin{cases} K_1 = f\left(x_n + h\left[\frac{1}{4}K_1 + \left(\frac{1}{4} - \frac{\sqrt{3}}{6} \right) K_2 \right] \right) \\ K_2 = f\left(x_n + h\left[\left(\frac{1}{4} + \frac{\sqrt{3}}{6} \right) K_1 + \frac{1}{4}K_2 \right] \right) \\ x_{n+1} = x_n + \frac{h}{2}\left[K_1 + K_2 \right] \end{cases} \tag{4.18}$$

Let

$$\begin{cases} \xi = x_n + h\left[\frac{1}{4}K_1 + \left(\frac{1}{4} - \frac{\sqrt{3}}{6} \right) K_2 \right] \\ \eta = x_n + h\left[\left(\frac{1}{4} + \frac{\sqrt{3}}{6} \right) K_1 + \frac{1}{4}K_2 \right] \end{cases} \tag{4.19}$$

then we can acquire another expression of HH4.

$$\begin{cases}(2\sqrt{3}-3)(\eta - x_n)+3(\xi - x_n) = f\left(\xi\right)\\ 3(\eta - x_n)-(2\sqrt{3}+3)(\xi - x_n) = f\left(\eta\right)\\ x_{n+1} = x_n + \sqrt{3}(\eta-\xi)\end{cases} \tag{4.20}$$

It can be seen that during the solution of HH4 two points η, ξ are calculated by the first two equations in (3.20), and then the next step value x_{n+1} is calculated with values of points η, ξ (seen in Figure 27).

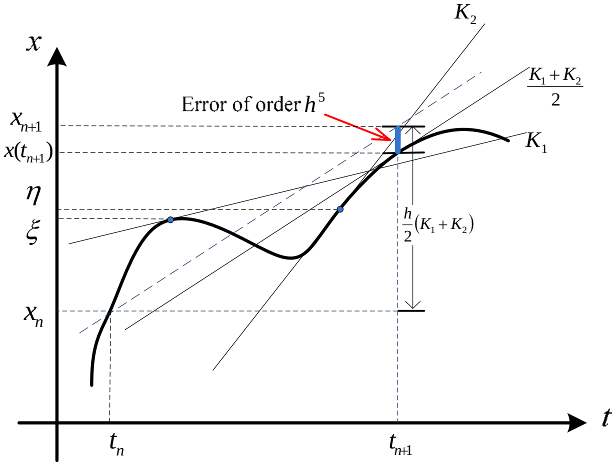


Figure 27. Integration method of Hammer-Hollingsworth 4

For the DAE system (2.1), the solution of x_{n+1} by HH4 integration method can be described as two stages, i) solution of vector η, ξ, and ii) solution of x_{n+1}. For the First stage, we need to solve the following non-linear algebraic equations.

$$\begin{cases}(2\sqrt{3}-3)(\eta - x_n)+3(\xi - x_n) = \mathbf{f}\left(\xi, y_\xi\right)\\ 0 = \mathbf{g}\left(\xi, y_\xi\right)\\ 3(\eta - x_n)-(2\sqrt{3}+3)(\xi - x_n) = \mathbf{f}\left(\eta, y_\eta\right)\\ 0 = \mathbf{g}\left(\eta, y_\eta\right)\end{cases} \tag{4.21}$$

For the second stage we need to solve

$$\begin{cases}x_{n+1} = x_n + \sqrt{3}(\eta-\xi)\\ 0 = \mathbf{g}(x_{n+1}, y_{n+1})\end{cases} \tag{4.22}$$

4.2.2 Numerical stability and precision of HH4

If we can apply the integration method of HH4 (3.18) to Dahlquist test equation, we can acquire the stability function of HH4, which can be described as

$$R(z) = \frac{1 + z/2 + z^2/12}{1 - z/2 + z^2/12} \tag{4.23}$$

The stability domain of HH4 is

$$S_{HH4} = \left\{ z \in \mathbf{C}; \quad \left| \frac{1 + z/2 + z^2\ /12}{1 - z/2 + z^2\ /12} \right| \le 1 \right\} \tag{4.24}$$

which can be illustrated in Figure 28. It can be found that the stability domain of HH4 is the whole left part of complex plain. Therefore, the integration method is able to deal with stiffness and hyper-stability problems.

Additionally, HH4 can guarantee the precision of h^4 with the local truncation error of $O(h^5)$ which can be proved by Tyler expression. The attribute of high precision of HH4 is able to enhance the efficiency of time-domain simulation since it can increase the integration step while guaranteeing the precision. For example, if we require the precision of integration method in the order of 10^{-4}, the maximum integration step for trapezoidal rule cannot be as large as 0.1 since its precision is in the order of h^2, while the integration method of HH4 can make the integration step as large as 0.1 due to its fourth order of precision. From the formula (3.20) and (3.21) it can be seen that HH4 need to solve a double-size non-linear algebraic equation compared with trapezoidal rule. If Newton method is used to solve the nonlinear algebraic equations, double-size linear equations with highly sparsity are need to be solved. With the sparse linear solver library, the integration method of HH4 is very competitive and superior to the method of trapezoidal rule. The comparison between HH4 and trapezoidal rule on time domain simulation of power system is shown in section 4.4.

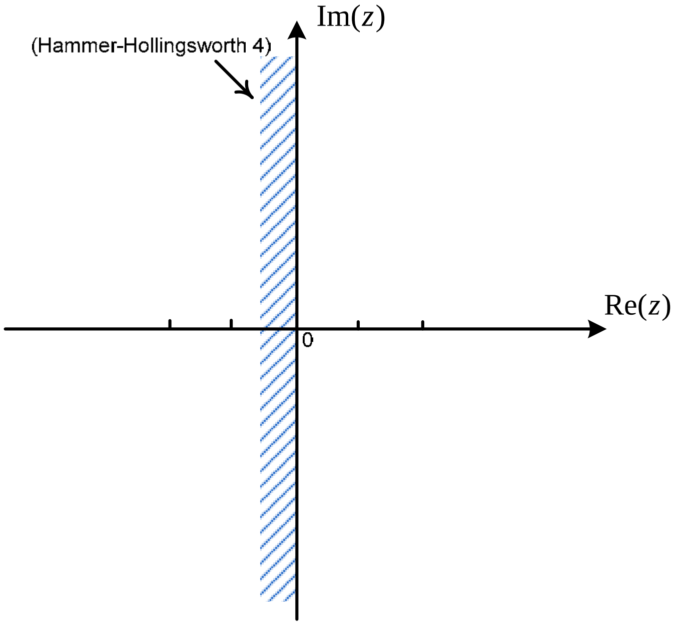


Figure 28. Stability domain of Hammer-Hollingsworth 4

4.3 Error Estimation and Time Step Control Techniques

Error estimation and time step control techniques are used in many commercial softwares of time domain simulation, such as EXTAB, EUROSTAG. Step control techniques is able to decrease or increase the integration step based on phenomena in simulation including the error estimation of integration methods, the number of iterations of Newton method and the changes in DAE system due to the switching events or faults.

4.3.1 Time step control criteria

In HSET-TDS, integration step is varied based on following criteria, i) the norm of the vector of error estimation is larger or smaller than a given threshold, ii) the iteration times of Newton method is more than a big value so that Newton iteration can be considered to be divergent, iii) DAE system is updated due to the switching events or faults and the error estimation cannot be estimated based on the previous values before updating. The first criterion can be described as follows [9].

$$h_{new} = h_{old}\left[tolerance / \left\| e \right\|_{\infty} \right]^{1/k} \tag{4.25}$$

where $\left\| e \right\|_{\infty}$ is the max norm of the estimated local truncation error, and k is the order of the integration method. HSET-TDS adopts a double tolerance, upper tolerance and lower tolerance, to control integration step. If the norm of error is larger than upper tolerance, the integration step will be decreased; while if the norm of error is smaller than lower tolerance, the integration step will be increased. The purpose of double tolerance is to make more cases of integration step usable, and thus the extra recalculation can be decreased.

4.3.2 Estimation of truncation error

The most important part in the time step control technique is how to estimate the truncation error. For an order *k* integration method, there are two methods to estimate its truncation error, i) extrapolation method, ii) another integration method with the same or larger order.

4.3.3 Extrapolation method

The main idea of extrapolation method is to make the integration step half, and then recalculated the value of next point, which is more precise than the next point value by original integration step. Then the main part of truncation error can be estimated by the value of next point of original integration step and half step. Assume that $x(t_{n+1})$ is the exact solution, $x_{n+1}^{[h]}$ is the numerical solution by an integration method of order *k* with integration step of *h*, and $x_{n+1}^{[h/2]}$ is the numerical solution by the same integration method with integration step of $h/2$. Thus, according to Gragg (1964) theory[36],

$$x_{n+1}^{[h]} = x(t_{n+1}) + \beta(x_{n+1})h^k + O(h^{k+1}) \tag{4.26}$$

$$x_{n+1}^{[h/2]} = x(t_{n+1}) + \beta(x_{n+1})\left(\frac{h}{2}\right)^k + O(h^{k+1}) \tag{4.27}$$

where $\beta(x_{n+1})$ is nothing to do with *h*. Expression (3.26) subtracts (3.27), and then

$$\left|\tilde{E}(x_{n+1})\right| = \left|\beta(x_{n+1})h^k\right| \approx \left|\frac{2^k(x_{n+1}^{[h]} - x_{n+1}^{[h/2]})}{2^k - 1}\right| \tag{4.28}$$

which can be considered the main part of the truncation error.

4.3.4 Error estimation by other integration methods

The extrapolation method is able to estimate the main part of truncation error; however, it can be found that another two steps of simulations are needed to calculate a more precise values of next step point. The extrapolation method is not economic for error estimation since it would need twice quantity of computation compared with the method whose truncation error is needed to be estimated. Another method to estimate truncation error is using another integration method with same order or larger order. In HSET-TDS, every implicit method has different integration methods to estimate its truncation error. The selection criteria of integrator for error estimation is based on, i) the integrator is explicit for sake of efficiency, ii) the integrator is allowed to use the last one step value. Table 3 shows the integration methods in HSET-TDS for error estimation for trapezoidal rule, theta method and HH4.

The implementation of Hammer-Hollingsworth 4 and its error estimation scheme require much computation at one integration step. With Newton method and sparse linear solver, it is more competitive than trapezoidal rule, and the whole integration times can be greatly decreased. The simulation comparison is shown in next section.

Table 3. Integration methods for error estimation

Integration Methods	***Integrator for Error Estimation***
Trapezoidal Rule $x_{n+1} = x_n + \frac{h}{2}\left[f(x_n) + f(x_{n+1})\right]$	**Explicit Adams 2** $\tilde{x}_{n+1} = x_n + h\left[\frac{3}{2} f(x_n) - \frac{1}{2} f(x_{n-1})\right]$
Theta Method $x_{n+1} = x_n + h\left[(1-\theta) f(x_n) + \theta\, f(x_{n+1})\right]$	**Explicit Theta Method [9]** $\tilde{x}_{n+1} = x_n + h\left(f(x_n) + \theta\left[f(x_n) - f(x_{n-1})\right]\right)$
Hammer-Hollingsworth 4 $\begin{cases} (2\sqrt{3}-3)(\eta - x_n) + 3(\xi - x_n) = f(\xi) \\ 3(\eta - x_n) - (2\sqrt{3}+3)(\xi - x_n) = f(\eta) \\ x_{n+1} = x_n + \sqrt{3}(\eta - \xi) \end{cases}$	**Explicit Runge-Kutta method, Ceschino 4[36]** $\begin{cases} K_1 = f(x_n) \\ K_2 = f(x_n + hK_1/4) \\ K_3 = f(x_n + hK_2/2) \\ K_4 = f\left[x_n + h(K_1 - 2K_2 + 2K_3)\right] \\ \tilde{x}_{n+1} = x_n + h\left(K_1/6 + 4K_3/6 + K_4/6\right) \end{cases}$

4.4 Simulation Results

IEEE New England system with 10 generators and 39 buses (shown in Figure 29) is utilized as a test system. Generator model for the system includes 4th order, 3rd order and classic 2nd order models, which are selected based on parameters. IEEE-1 standard model of excitation, shown in Figure 14, is used to control the voltage of generators, and governor model 8 in ETMSP manual [32] (shown in Figure 15) is adopted to control the input of mechanical torque. HSET-TDS utilizes direct solution method to construct and solve the DAE system, and the whole DAE system includes 105 ODEs and 190 algebraic equations. The initial values of the DAE system is computed by power flow program in HSET-TDS, which is based on Newton-Raphson method.

In order to compare the integration methods of Trapezoidal rule and HH4, an event of fault on bus 26 is selected, starting at 0.5s and lasting for 0.3 seconds. The whole transient process is simulated for 8 seconds, and the voltage and speed of generator 37 are monitored. The whole simulation comparison is implemented from two perspective, i) fixed integration step, and ii) variable integration step.

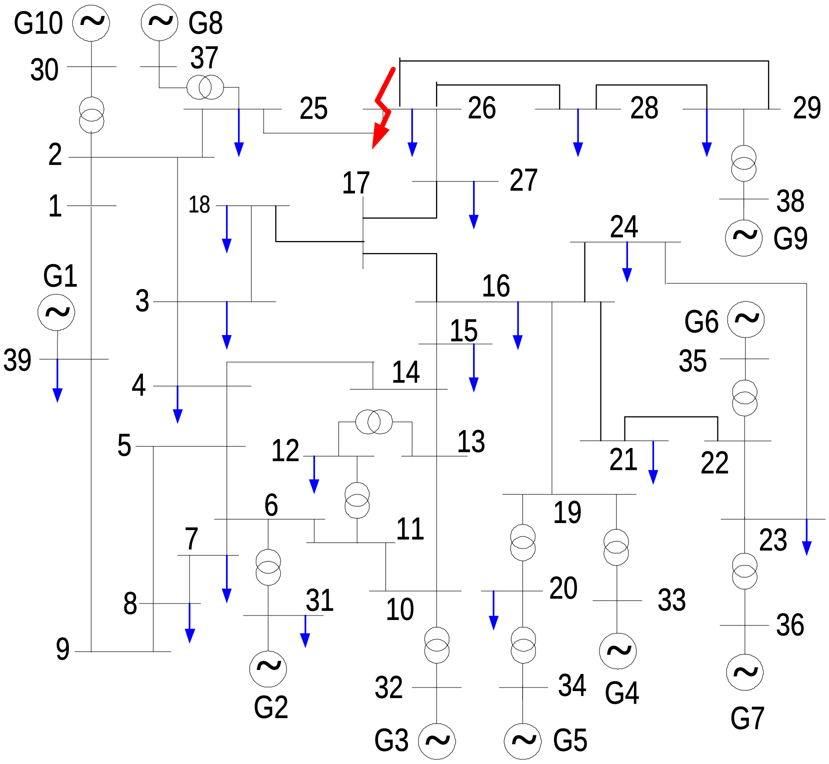


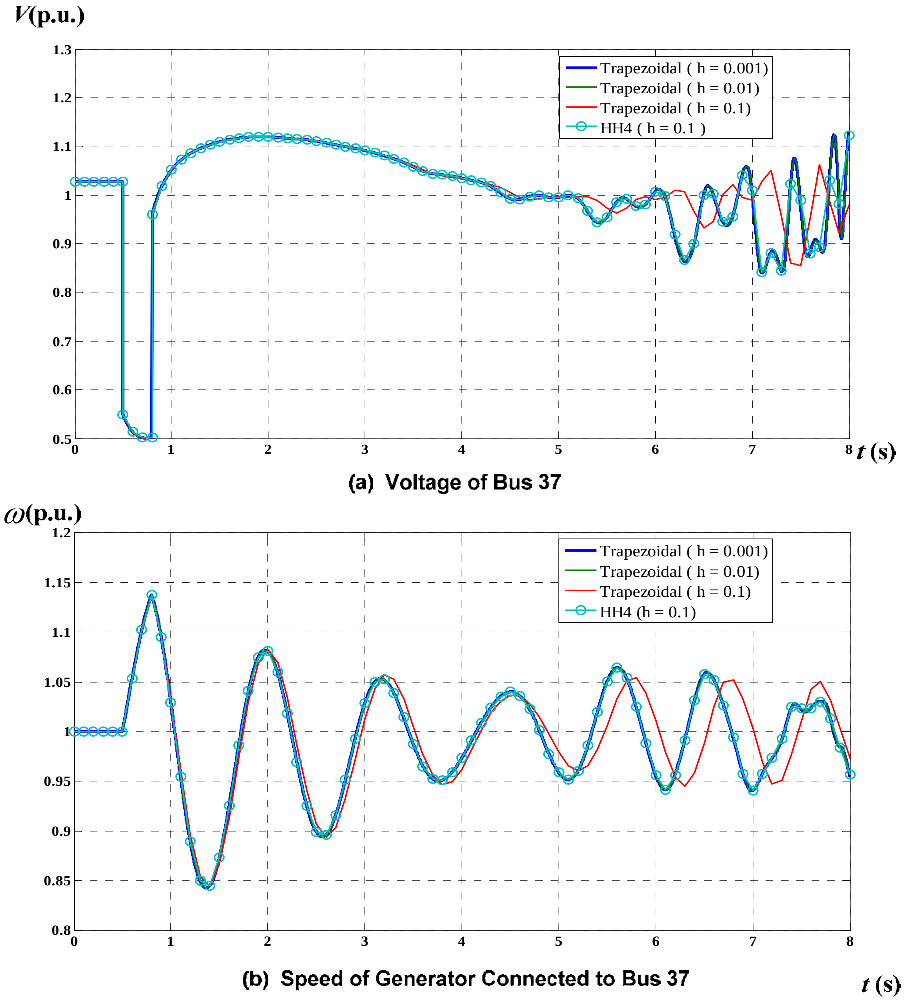
Figure 29. New England 39 bus system with fault on bus 26

4.4.1 Fixed Integration Step

The integration steps are selected as 0.001s, 0.01s and 0.1s for both trapezoidal rule and HH4. The simulation results by trapezoidal rule with h=0.001 is assumed to be the exact solution. The computational cost is shown in Table 4. From the results shown in Figure 30, it can be found that the simulation curves by trapezoidal rule with h=0.01 are still acceptable, and the curves by trapezoidal rule with h=0.01 deviates the exact solution; while points constructing the curves by HH4 with h=0.1 are well matched with the exact solution. Additionally, the computational time used by HH4 with h=0.1 is much shorter than the time used by trapezoidal rule with h=0.01.

Table 4. Simulation results with fixed integration step

Integration Methods	Integration Step (s)	Algebraic Solver	Linear Solver Library	Simulation Time (s)
Trapezoidal	0.001	Newton	SuperLU	300.573
Trapezoidal	0.01	Newton	SuperLU	33.234
Trapezoidal	0.1	Newton	SuperLU	6.875
HH4	0.1	Newton	SuperLU	9.328



(a) Voltage of Bus 37

(b) Speed of Generator Connected to Bus 37

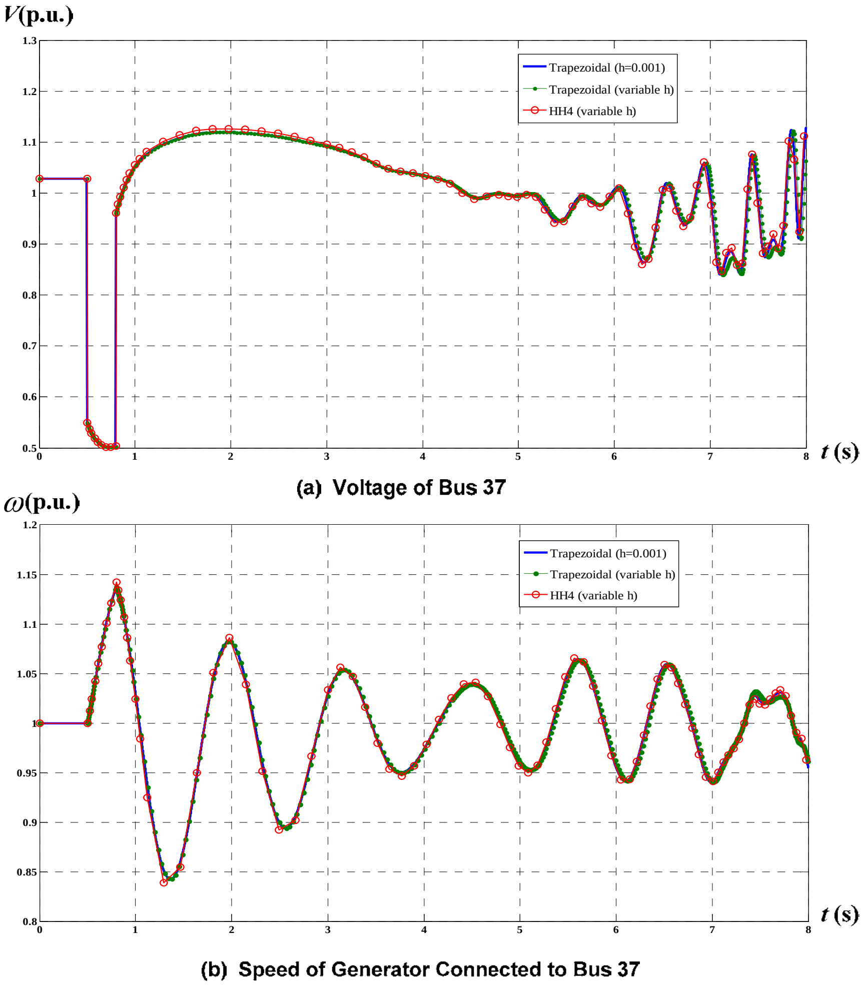
Figure 30. Simulation results of New England 39 bus system with fixed integration step

4.4.2 Variable Integration Step

HSET-TDS is developed with variable step technique, and the error estimation methods are listed in Table 3. Since HH4 possesses the attribute of high precision together with same stability domain of trapezoidal rule. We expect that the simulation by HH4 with variable step technique will use less integration times than the trapezoidal rule, and the error bounds for both methods are set to be the same. The maximum integration step will be recorded, and the simulation results are shown in Figure 31 and Table 5.

Table 5. Simulation results with variable integration step

Integration Methods	Maximum Step (s)	Computation Times	Algebraic Solver	Linear Solver Library	Simulation Time (s)
HH4	0.170859	90	Newton	SuperLU	19.046
Trapezoidal	0.036041	437	Newton	SuperLU	58.624



(a) Voltage of Bus 37

(b) Speed of Generator Connected to Bus 37

Figure 31. Simulation results of New England 39 bus system with variable integration step

5. Stiffness Detection and Decoupling Method

Step-by-step numerical integration is a basic way to solve differential algebraic equations (DAE). Usually numerical integration methods can be classified into two categories: explicit methods and implicit methods. In the explicit methods, the next step calculation uses only the solution information known, and the computation of each iteration points is efficient. However, it is reported that the explicit method involved with numerical stability problem when stiff problems are solved [5, 6, 35]. Stiff equations are problems for which explicit methods don't work [35]. It can lead to large error when fixed integration step technique is used or too small integration step when variable step technique is used. In the implicit method, the calculation uses the unknown solution information of next-step(s), and it needs to solve non-linear equations at each step. The implicit methods are slow but stable. Implicit methods are commonly used for solving power system dynamic simulation. It is meaningful to make use of the advantages of explicit method and implicit method to integrate these two of methods.

In this chapter, a decoupling method which can combine the explicit method and implicit method is going to be introduced. The main idea of this method is to decouple differential part of DAE into stiff part and non-stiff part. Explicit methods will be adopted to solve non-stiff part, while implicit methods will be used to solve stiff part. The critical technique for the decoupling method is how to detect the stiffness in DAE system and how to relatively correct partition the DAE into stiff part and non-stiff part. The following sections will introduce the techniques, and the practical technique used in HSET-TDS.

5.1 Automatic Stiffness Detection

Stiff equations are problems for which explicit methods don't work. There are two methods to detect the stiffness of ordinary differential equations [35].

5.1.1 Based on estimation of eigenvalues λ_i of the Jacobian matrix J

Assume that the ***stability function*** corresponding to an explicit method is $R(z)$, where $z = h\lambda$ and h is the step value. The ***stability domain*** of a method is the set $S = \{z \in C; |R(z) \le 1|\}$. If the eigenvalue λ_i satisfies the condition that $R(h\lambda_i) < 1$, then $h\lambda_i$ lies in the stability domain. The differential equations whose eigenvalues do not satisfy the condition of $R(h\lambda_i) < 1$ are stiff.

An example based on the test function $\dot{x} = \lambda x$ can demonstrate the stiffness phenomenon which is associated with eigenvalues and step. The explicit method is Forward Euler method (FEM), whose stability function is $R_{FEM} = (1 + h\lambda)$ .We assume that step $h = 0.001$, and λ is selected as $\lambda_1 = -100$, $\lambda_2 = -1000 + 1000i$, $\lambda_3 = -1500 + 1000i$,

When $\lambda_1 = -100$, it can be obtained that $|R(h\lambda_i)| < 1$, and the problem is non-stiff. From the Figure 32, it can be found that the simulation results is matched with the exact solution. When $\lambda_2 = -1000 + 1000i$, it can be obtained that $|R(h\lambda_i)| = 1$ where the critical phenomenon appears. The simulation result show that there is much oscillation but the result is not totally divergent. When $\lambda_3 = -1500 + 1000i$, it can be found that $|R(h\lambda_i)| > 1$, and the problem is stiff. The simulation results in figure become divergent.

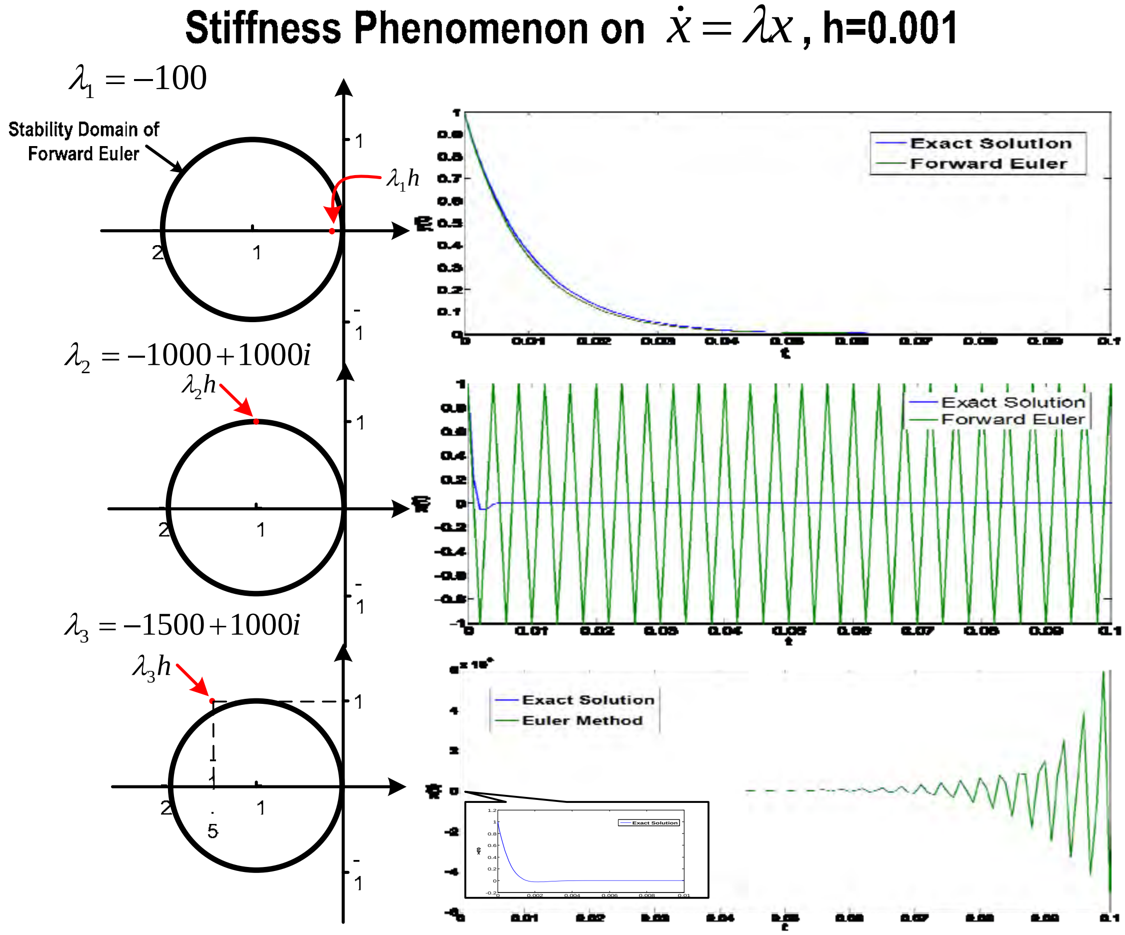


Figure 32. Stiffness phenomenon associated with eigenvalues

Therefore, during the process of one step integration, if all the eigenvalues of the Jacobian matrix from DAE system can be acquired, stiffness can be easily judged from the stability function of an explicit method. The differential equations whose eigenvalues are out of stability domain (for given explicit method) will be differentiated by implicit methods, while the explicit methods will be utilized to discretize differential equations whose eigenvalues are within stability domain. This method is not efficient since we need to calculate the eigenvalues of the Jacobian matrix by the numerical iterative methods such as QR decomposition or Arnoldi method, which are time-exhausting. Furthermore, since the eigenvalues are time-variable, the eigenvalues of Jacobian matrix needs to be updated after each integration step. This issue makes the stiffness detection by calculation of eigenvalues hard to meet the requirement of fast simulation.

5.1.2 Based on error estimation

The main idea of this method is to check the variation of global truncation error (GTE) of each step by explicit methods. The global truncation error can be efficiently estimated by Richardson extrapolation method. When $h\lambda_i$ lies within the stability domain, global truncation error is convergent and the values have decreasing tendency; while when $h\lambda_i$ lies outside the stability domain, global truncation error is divergent and the values have increasing tendency.

1) Illustration on a linear system

We continue the example linear system discussed in the last section. For convenience, we skip the process of Richardson extrapolation or error estimation techniques introduced in chapter 4 and use the exact solution to calculate global truncation error directly. The exact solution for the test function is $x(t) = e^{\lambda t}$ .The absolute values of GTE are shown in Figure 33.

From Figure 33, it can be found that when the differential equation is nonstiff for forward Euler method, the GTE is convergent finally. When the differential equations is in critical condition for forward Euler method, the GTE becomes a constant value. When the differential equations is stiff, the GTE becomes divergent.

2) Illustration on a nonlinear system

The next example shows whether the automatic stiffness detection based on error estimation works for a nonlinear system when the eigenvalues will change time to time and the system varies from stiff to non-stiff for the method of forward Euler.

Analyze the ODE $\begin{cases} \dot{x} = x^2 \\ x(0) = x_0 \end{cases}$ The exact solution is $x(t) = -\dfrac{1}{t + {}^{1}\!/\!_{x_0}}$ We assume that step $h = 0.001$. We still use Forward Euler method to calculate the system. In order to make the eigenvalues of the ODE system variable starting from different point, we set the initial values for three cases $x_0 = 400$, $x_0 = 1000$ and $x_0 = 1100$. The system is simulated for 0.2 second. We record the following curves, i) the trajectory of eigenvalues, ii) the simulation solution by forward Euler method with step *h* and *h*/2, iii) global truncation error by exact solution and extrapolation. The simulation results are shown in Figure 34 and the analysis is shown as follows.

i) When $x_0 = 400$, *z* starts from the inside of the stability domain. It can be found that both simulation results are able to match with the exact solution. Also, the GTE from both exact solution and extrapolation becomes smaller and smaller following the change of *z*.

ii) When $x_0 = 1000$, *z* starts from the boundary of the stability domain. We can found that the simulation result with *h*/2 is able to match with the exact solution while the simulation result with *h* becomes bad and not totally divergent. Besides, the GTE from both exact solution and extrapolation becomes smaller and smaller following the change of *z*, but the error at the beginning is larger than that when $x_0 = 400$.

iii) When $x_0 = 1100$, *z* starts from the outside of the stability domain. It can be found that both simulation results are not able to match with the exact solution. Also, the GTE from both exact solution and extrapolation becomes larger and larger following the change of *z*.

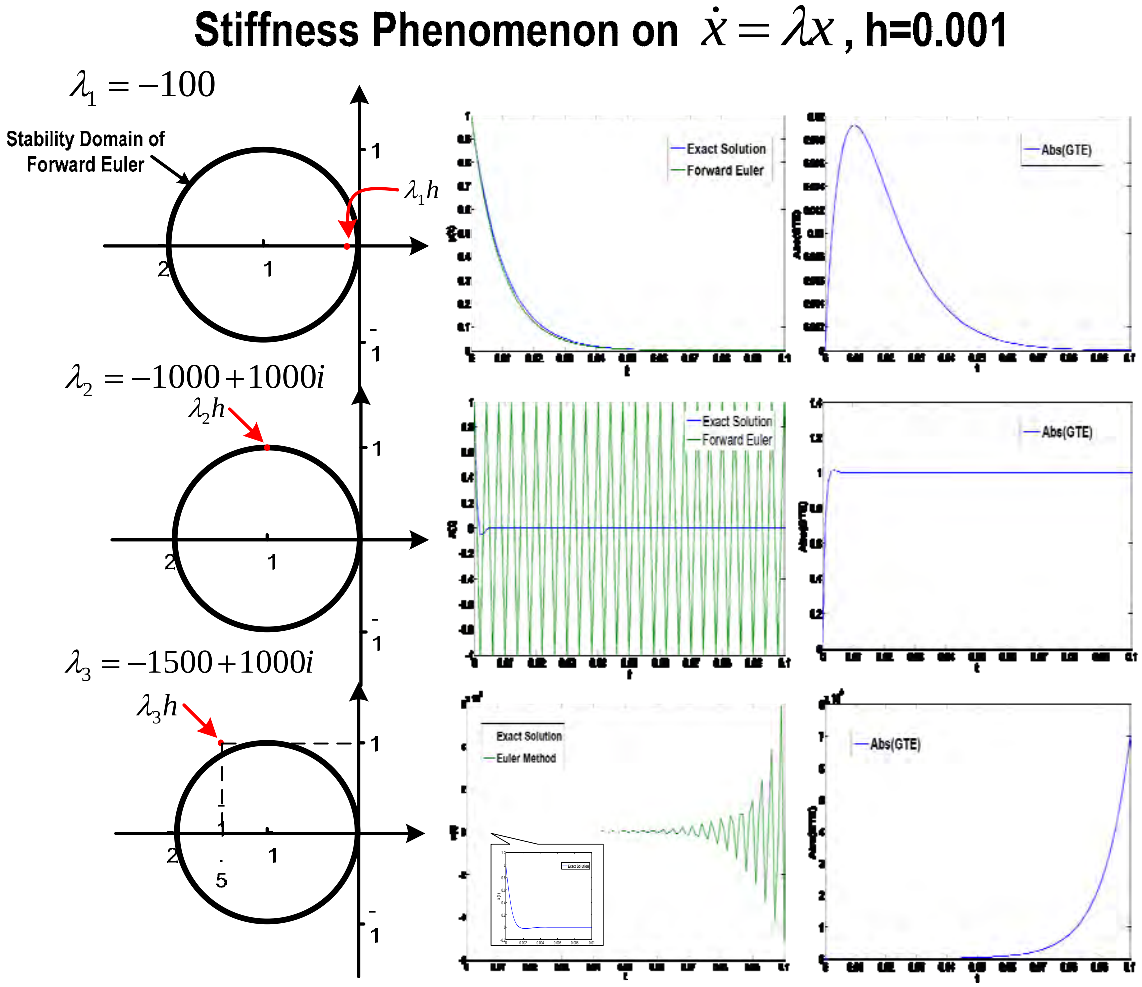


Figure 33. Stiffness phenomenon associated with GTE on test function

3) Illustration on a nonlinear system

The next example shows whether the automatic stiffness detection based on error estimation works for a nonlinear system when the eigenvalues will change time to time and the system varies from stiff to non-stiff for the method of forward Euler.

Analyze the ODE $\begin{cases} \dot{x} = x^2 \\ x(0) = x_0 \end{cases}$ The exact solution is $x(t) = -\dfrac{1}{t + {}^{1}\!/_{x_0}}$ We assume that step $h = 0.001$. We still use Forward Euler method to calculate the system. In order to make the eigenvalues of the ODE system variable starting from different point, we set the initial values for three cases $x_0 = 400$, $x_0 = 1000$ and $x_0 = 1100$. The system is simulated for 0.2 second. We record the following curves, i) the trajectory of eigenvalues, ii) the simulation solution by forward Euler method with step *h* and *h*/2, iii) global truncation error by exact solution and extrapolation. The simulation results are shown in Figure 34 and the analysis is shown as follows.

iv) When $x_0 = 400$, *z* starts from the inside of the stability domain. It can be found that both simulation results are able to match with the exact solution. Also, the GTE from both exact solution and extrapolation becomes smaller and smaller following the change of *z*.

v) When $x_0 = 1000$, *z* starts from the boundary of the stability domain. We can found that the simulation result with *h*/2 is able to match with the exact solution while the simulation result with *h* becomes bad and not totally divergent. Besides, the GTE from both exact solution and extrapolation becomes smaller and smaller following the change of *z*, but the error at the beginning is larger than that when $x_0 = 400$.

vi) When $x_0 = 1100$, *z* starts from the outside of the stability domain. It can be found that both simulation results are not able to match with the exact solution. Also, the GTE from both exact solution and extrapolation becomes larger and larger following the change of *z*.

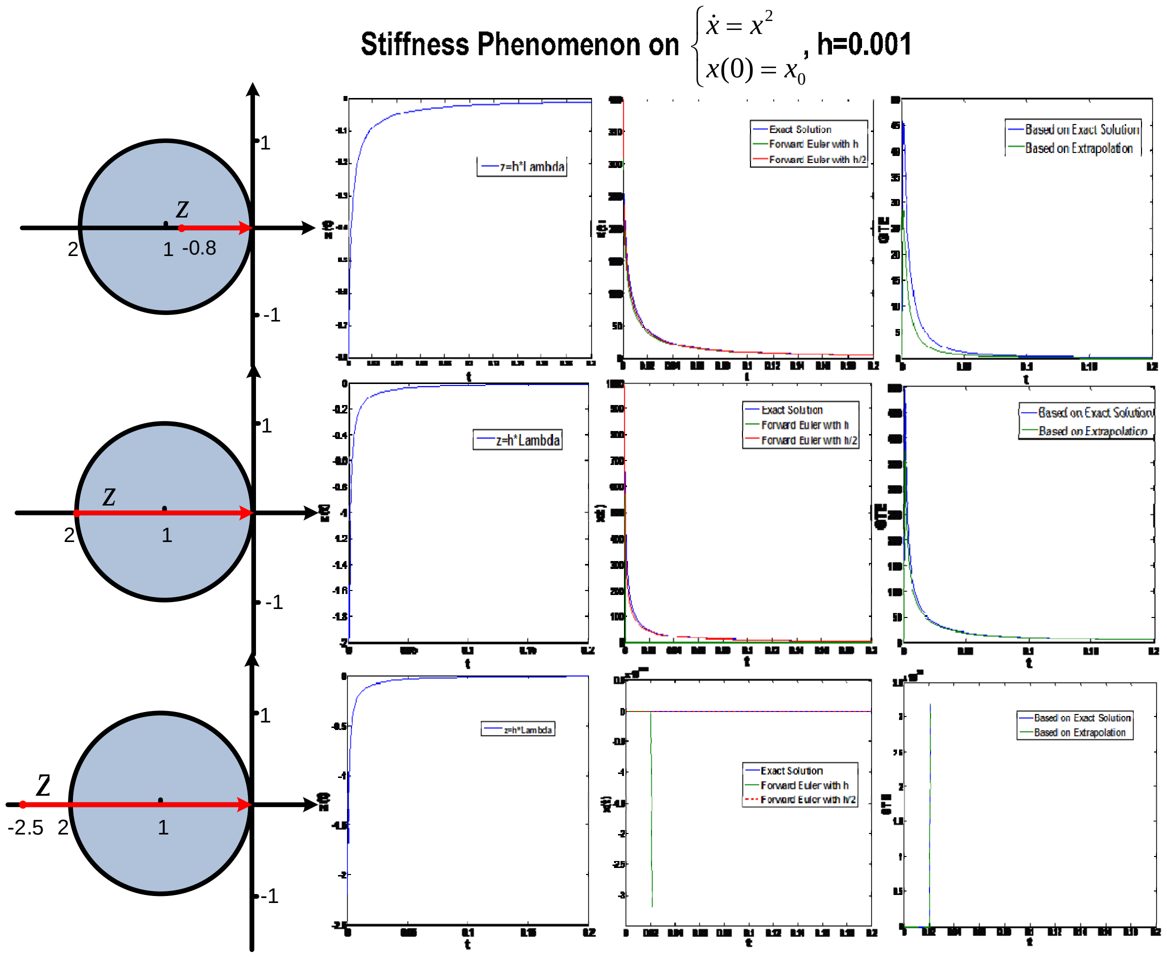

Stiffness Phenomenon on $\begin{cases}\dot{x} = x^2 \\ x(0) = x_0\end{cases}$, h=0.001

Figure 34. Stiffness phenomenon associated with GTE on a nonlinear system

The automatic stiffness detection based on error estimation can efficiently detect the stiffness of ODE system. The computational cost is just basic global error estimation. The potential problem about this stiffness detection method is how to detect the increasing or decreasing tendency of the GTE. It's possible that the result of detection is conservative because probably there are many cycles of oscillation until the GTE decrease. There are two approaches to deal with this problem.

- [35] introduces an idea, which is to check the error in succession (say 15 times). Then final conclusion of stiffness or non-stiffness can be drawn.
- Another idea is to give a threshold for global truncation error. If the GTE is over the threshold, the corresponding equation is considered to be stiff. The stiffness threshold for an explicit method can be fixed before the simulation. It's a good idea to set the threshold with reference to the critical case. Besides, concerning that the absolute GTE is unfair for each equation of ODE system, both relative and absolute GTE can be used to check the stiffness.

5.2 Stiffness Decoupling Method

5.2.1 Recursive Projection Method

The main idea behind achieving computational gain is to take advantage of both the explicit and the implicit methods and simultaneously achieve as much parallelism as possible. This is made possible through the division of the ODE part of the DAE into stiff and non-stiff parts through a partition algorithm called recursive projection method (RPM) [42,43], an invariant subspace method. Consider the ODE system described in (5.1).

$$\begin{cases} \dot{x} = \mathbf{f}(x) \\ x(0) = x_0 \end{cases} \tag{5.1}$$

where ***x*** is an *n*-dimensional vector, and **f(*x*)** is an *n*-dimensional vector function described in expression (3)

$$\begin{cases} \dot{\boldsymbol{x}}_{\boldsymbol{s}} = \boldsymbol{f}_{\boldsymbol{s}}(\boldsymbol{x}_{\boldsymbol{s}}, \boldsymbol{x}_{\boldsymbol{ns}}) \\ \dot{\boldsymbol{x}}_{\boldsymbol{ns}} = \boldsymbol{f}_{\boldsymbol{ns}}(\boldsymbol{x}_{\boldsymbol{s}}, \boldsymbol{x}_{\boldsymbol{ns}}) \end{cases} \tag{5.2}$$

where $\boldsymbol{x_s}$ and $\boldsymbol{x_{ns}}$ are stiff and non-stiff variables, and $\boldsymbol{f_s}$ and $\boldsymbol{f_{ns}}$ are stiff and non-stiff equations. For the non-stiff part of ODEs, explicit methods can be used to efficiently compute the values of next point with numerical stability guaranteed; while implicit methods deal with the stability problems for the stiff parts using iterative computing.

The solution space $\mathbf{R}^n$ of ODEs (2) can be written as a direct sum of the span of the stiff eigenspace (say, invariant subspace **P**) and it's orthogonal complement (say, invariant subspace **Q),** which is the non-stiff eigenspace. The original *n*-dimensional space can be split into two subsystems:

$$\mathbf{f}^{\boldsymbol{P}}(\boldsymbol{p},\boldsymbol{q}) = \boldsymbol{Z}_1^{\boldsymbol{T}}\mathbf{f}(\boldsymbol{Z}_1\boldsymbol{p} + \boldsymbol{Z}_2\boldsymbol{q}) \tag{5.3}$$

$$\mathbf{f}^{\boldsymbol{Q}}(\boldsymbol{p},\boldsymbol{q}) = \boldsymbol{Z}_2^{\boldsymbol{T}}\mathbf{f}(\boldsymbol{Z}_1\boldsymbol{p} + \boldsymbol{Z}_2\boldsymbol{q}) \tag{5.4}$$

where $\boldsymbol{Z_1}$ and $\boldsymbol{Z_2}$ are basis of invariant subspace **P** and **Q**, respectively, **Q** is the orthogonal complement of **P** such that $\mathbf{Q} = \mathbf{P}^{\perp}$, and $\boldsymbol{p} = \boldsymbol{Z}_1^{\boldsymbol{T}}\boldsymbol{x}$, $\boldsymbol{q} = \boldsymbol{Z}_2^{\boldsymbol{T}}\boldsymbol{x}$. Thus, the ODEs system equations can be decoupled into two subsystems:

$$\begin{cases} \dot{\boldsymbol{p}} = \mathbf{f}^{\boldsymbol{P}}(\boldsymbol{p},\boldsymbol{q}) = \boldsymbol{Z}_1^{\boldsymbol{T}}\mathbf{f}(\boldsymbol{Z}_1\boldsymbol{p} + \boldsymbol{Z}_2\boldsymbol{q}) \\ \dot{\boldsymbol{q}} = \mathbf{f}^{\boldsymbol{Q}}(\boldsymbol{p},\boldsymbol{q}) = \boldsymbol{Z}_2^{\boldsymbol{T}}\mathbf{f}(\boldsymbol{Z}_1\boldsymbol{p} + \boldsymbol{Z}_2\boldsymbol{q}) \end{cases} \tag{5.5}$$

By solving the above decoupled equations, ***p*** and ***q*** can be calculated separately, and the original states $\boldsymbol{x} = \boldsymbol{Z}_1\boldsymbol{p} + \boldsymbol{Z}_2\boldsymbol{q}$.

The critical problem of the recursive projection method is how to detect the stiffness, and how to fix the basis $\boldsymbol{Z_1}$ and $\boldsymbol{Z_2}$ of invariant subspace **P** and **Q.** As the Jacobian matrix of the original ODE systems is updated, the basis of invariant subspace P and **Q** changes correspondingly, meaning the basis $\boldsymbol{Z_1}$and $\boldsymbol{Z_2}$ needs to be updated as well. There are following several reported methods about estimating eigenvalues to detect stiffness. Reference [42] describes a method which approximates some vectors of the Jacobian matrix and uses QR decomposition to detect the stiffness and fix the basis. Reference [43] utilizes the Cayley transform of the Jacobian matrix and an iterative process to construct the basis $\boldsymbol{Z}_1$. The Arnoldi method is adopted in [44] to identify eigenvalues outside the stability domain of an explicit method. The stiffness detection based on the eigenvalues solution of Jacobian matrix is precise but computational. The other two methods, QR decomposition and Cayley, also involve obtaining eigenvalues, but the strategy based on QR decomposition may be more efficient since just one step of QR decomposition is implemented in a one step stiffness detection.

5.2.2 Stiffness Decoupling Method used in HSET-TDS

The recursive projection method introduced in [42] is able to roughly detect the stiffness in the differential part of DAE system. The computational cost the method is the one step QR decomposition of Jacobian matrix. There are mainly two disadvantages in the method, 1) one step QR decomposition can be much time-exhausting for large DAE system, 2) the roughness of stiffness detection may lead to failure of one step integration.

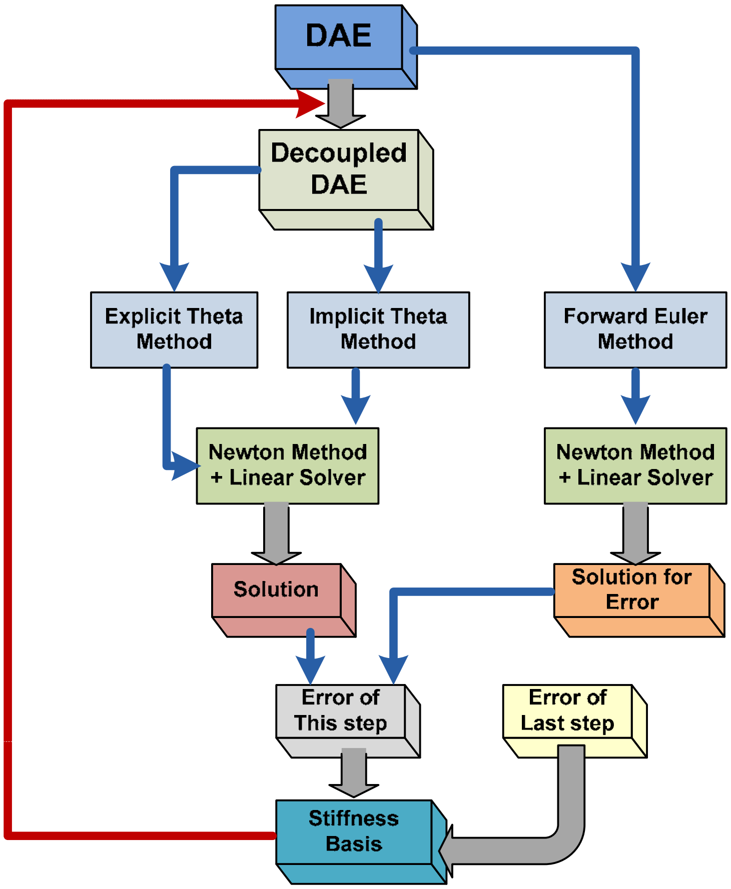


Figure 35. Stiffness Decoupling technique used in HSET-TDS

In HSET-TDS, stiffness decoupling based on error estimation is utilized. The technique process can be illustrated in Figure 35. Since the stiffness detection needs the error estimation, the stiffness decoupling defaults to use variable-step techniques, which will change the integration step and stiffness basis during the process of each integration step computation. The figure shows the example of explicit theta method and implicit theta method for non-stiff and stiff parts, and forward Euler method for error estimation. The main process of stiffness decoupling method associated with explicit theta and implicit theta method can be described as follows.

1) The stiffness basis is set to be zero set at the beginning.
2) The differential part of original DAE system is separated into stiff part and non-stiff part according to the stiffness basis.
3) Explicit theta method is utilized to differentiate the non-stiff part, and implicit theta method is adopted to differentiate the stiff part. Therefore, the differential part of DAE system is transformed to be a set of algebraic equations.
4) Newton method is used to solve the algebraic equations from differential part and algebraic part of DAE system, and then the solution by stiffness decoupling method can be acquired.
5) Forward Euler method is utilized to differentiate the differential part of original DAE system. And the solution by forward Euler can be acquired after the Newton method.
6) Error of this step can be obtain by error estimation formula specifically for theta method, and the error needs to be saved for next step calculation.
7) Stiffness basis can be updated with the error of this step and that of the last step. For one differential equation in DAE system, it will be updated to be stiff if the equation is non-stiff from the stiffness basis of last step and the error now of the variable from this equation is larger than error from last step. Similarly, it will be update to be non-stiff if the equation is stiff from the stiffness basis of last step and the error now is less than last step error or within a certain interval.
8) Continue the variable step technique.

During the process discussed above, two important issues needs to be clarified.

1) The error estimation technique for theta method by forward Euler is feasible, since the estimation part will the part of h^2. This does not work if the theta is zero since the theta method becomes explicit Euler, and the error will be always zero. If the theta is 0.5, $x_{n+1}^{[h]}$ by theta method can be formularized as (5.6) where $x(t_{n+1})$ is the real value of next step. $\bar{x}_{n+1}^{[h]}$ by Forward Euler method can be formularized as (5.7). The error part can be shown in (5.8). Since the estimation will maximized the real error, the estimated error will be multiplied by h and a coefficient.

$$x_{n+1}^{[h]} = x(t_{n+1}) + \alpha(x_{n+1})h^3 + O(h^4) \tag{5.6}$$

$$\bar{x}_{n+1}^{[h]} = x(t_{n+1}) + \beta(x_{n+1})h^2 + O(h^3) \tag{5.7}$$

$$\bar{x}_{n+1}^{[h]} - x_{n+1}^{[h]} = \beta(x_{n+1})h^2 + O(h^3) \tag{5.8}$$

2) The criterion for updating stiff part of equations to be non-stiff part is different from that for updating non-stiff part to stiff part. Assume that a differential equation is non-stiff from the last step stiffness list, which means the error of last step is calculated based on explicit theta method. According to the automatic stiffness detection in 5.1.2, the eigenvalue corresponding to the differential variable of this equation will be outside stability domain if error of this step is larger than the error of the last step. While, if a differential equation is stiff from the last step stiffness list, which means the error of last step is calculated based on implicit theta method, the eigenvalue corresponding to the differential variable of this equation will still be outside stability domain if the error of this step is smaller than the error of the last step. For example, when theta is 0.5, stability function of implicit theta method (Trapezoidal rule) can be expressed as (5.9), and the relationship between error and stability function can be described as (5.10).

$$R(z) = \frac{2+z}{2-z} \tag{5.9}$$

$$\frac{\delta_{n+1}}{\delta_n} = R(z) \tag{5.10}$$

If we want z is within the stability domain of explicit method(such as Forward Euler), the interval for z is $|1+z|<1$. From (5.10), we can get the interval $[\frac{1}{3}, 1]$ for $\frac{\delta_{n+1}}{\delta_n}$, where δ_n and δ_{n+1} are the errors from this step and last step by implicit theta method respectively.

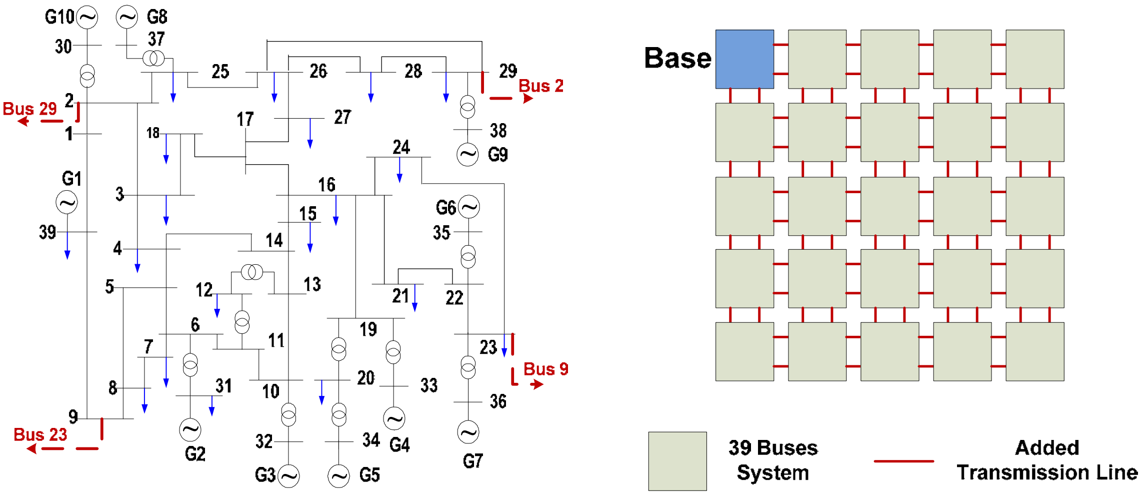
The stiffness decoupling method embedded with variable step technique can more efficient than both variable step implicit method and variable step explicit method. The computational cost is just the updating stiffness basis in each integration step by utilizing the error of this step and last step. Because of double use of error for stiffness detection and integration step resetting, the stiffness decoupling method is of certain practical significance.

5.3 Simulation Results

The expanded 975 buses system expanded from New England 39 buses system by 25 times is adopted as a test system. The construction of the system is similarly to the 3900 buses system introduced in 3.3.2. The bus 2, 9, 23, 29 are reconnected, and the structure of construction follows the mesh shown in Figure 36.

The event of the case is selected to be bus fault on bus 17 starting from 0.5s and lasting for 0.1s. The voltage of bus 37 and the speed of G8 is monitored as shown in Figure 37. The whole simulation lasts for 10 second.

Three integration methods are going to be compared, stiffness decoupling method based on implicit theta and explicit theta methods, variable-step implicit theta method and variable-step explicit theta method. The theta is set to be 0.5, and Forward Euler method is used for error estimation. The upper and lower boundary for variable step technique is 0.001 and 0.0005. The nonlinear solver for these three methods is Newton based method, and linear solver is SuperLU. The simulation results are shown in Table 6 and Figure 37. It can be seen that the stiffness decoupling method is relatively more efficient than other two methods. The explicit method takes more time, since its stability domain is limited and it need smaller integration step to guarantee the precision.

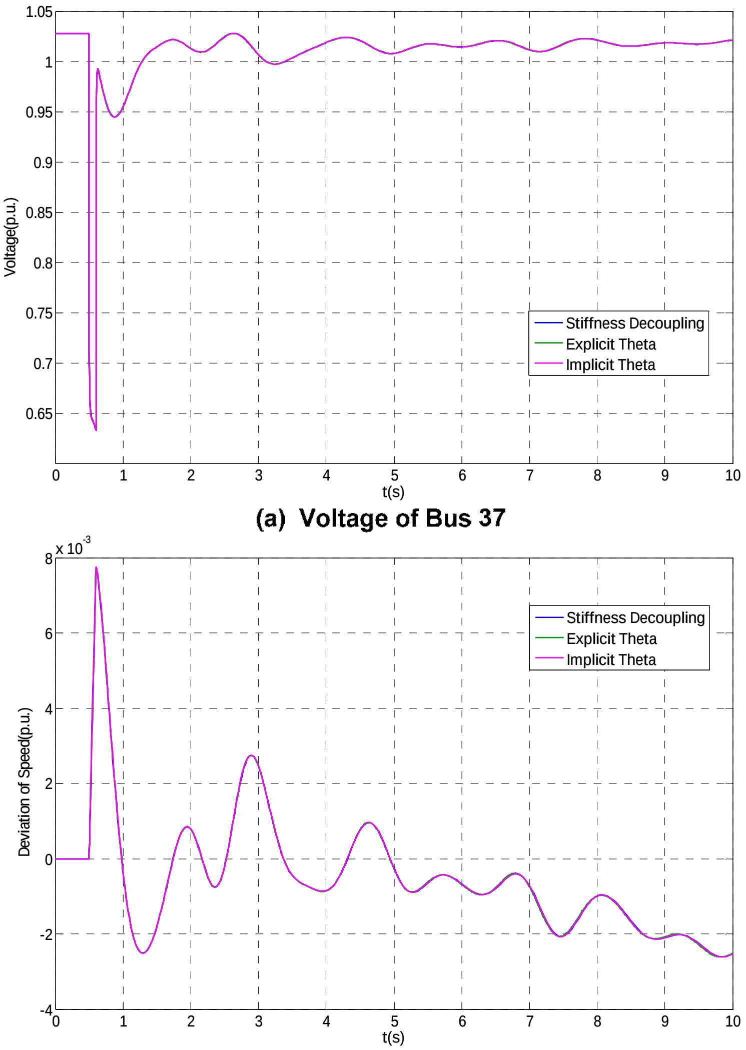


(a) Original 39 Buses System (b) Expanding Mesh

Figure 36. Expanded system from New England 39 bus system (5×5)

Table 6. Simulation results of Stiffness Decoupling, Explicit Theta and Implicit Theta

Integration Methods	Maximum Step (s)	Computation Times	Algebraic Solver	Linear Solver Library	Simulation Time (s)
Stiffness Decoupling	0.050096	327	Newton	SuperLU	82.264
Explicit Theta	0.006050	1674	Newton	SuperLU	223.632
Implicit Theta	0.0417	405	Newton	SuperLU	99.871



(a) Voltage of Bus 37

(b) Speed Deviation of Generator 8

Figure 37. Simulation results of expanded 975 bus system

6. Sequential and Parallel Library of SuperLU Solver

6.1 Introduction

The process of solving linear algebraic equations is important in time domain simulation when Newton-Raphson method is used. In Figure 4, it can be seen that the solution of solving linear equations is the bottom stage of all computational process, where usually most of computation is carried on during the whole process of time domain simulation. Due to the fact that coefficient matrices describing DAE system of power system and all Jacobian matrices of DAE system and nonlinear equations are highly sparse, it is indispensible to adopt a sparse solver of linear equations to improve the efficiency of time domain simulation. In this section, SuplerLU[23, 45, 46, 47, 48, 49], an open source library developed by computer science division, university of California, Berkeley, CA, is introduced, and this library is one of sparse solvers included in HSET-TDS.

The library of SuperLU contains three sub-libraries for both sequential and parallel computing, the status of which is summarized in Table 7 from [45]. The sub-library of sequential SuperLU is designed for sequential processors; multithreaded SuperLU (SuperLU_MT) is developed for shared memory parallel processors with shared memory routines of Pthreads or OpenMP; distributed SuperLU (SuperLU_DIST) is parallel linear solver for distributed memory parallel processors with MPI for interprocess communication. In the sequential computing version of HSET-TDS, the library of sequential SuperLU is included, while the library of distributed SuperLU is adopted to further develop the efficiency of time domain simulation in the parallel computing version of HSET-TDS.

Table 7. Status of SuperLU library

	Sequential SuperLU	SuperLU_MT	SuperLU_DIST
Platform	Serial	Shared-memory	Districuted-memory
Language	C	C + OpenMP (or Pthreads)	C + MPI
Data Type	real / complex single / double	real / complex single / double	real / complex double

[45] introduces the overall algorithm and him of sparse Gaussian elimination adopted in SuperLU, which can be described as following two steps.

a) Compute a triangular factorization $P_r D_r A D_c P_c = LU$, where D_r and D_c are diagonal matrices to equilibrate the system, and P_r and P_c are permutation matrices to reorder the rows and columns of A . L is a unit lower triangular matrix ($L_{ii} = 1$) and U is an upper triangular matrix.

b) Solve $AX = B$ by evaluating

$$X = A^{-1}B = (D_r^{-1}P_r^{-1}LUP_c^{-1}D_c^{-1})^{-1}B = D_c(P_c(U^{-1}(L^{-1}(P_r(D_rB)))))$$

SuperLU supplies several different routines to operate the solution, and HSET-TDS adopts some routines which are appropriate for time domain simulation of power system.

6.2 Sequential SuperLU

The computational routines of sequential SuperLU can be described in Figure 38. SuperLU supplies two main driver routines, which are convenient to call SuperLU library, i) simple driver dgssv(), and ii) expert driver dgssvx(). From the option configuration given by users, both driver routines implement following operations shown in Figure 38, factorization, triangular solving, estimating condition number, equilibrating and refining solution. In the expert driver routine, more options are provided to supply, and factorization type of the same pattern sparsity is a good option which can further develop the efficiency of time domain simulation. In the process of time domain simulation, the sparsity of Jacobian matrix is updated only if the topology of networks will be changed, and for most of simulation process the sparsity of Jacobian is fixed. Therefore, the diagonal matrices (D_r and D_c) and permutation matrices (P_r and P_c) can be used repeatedly, and the computational efficiency of solving linear equations can be improved.

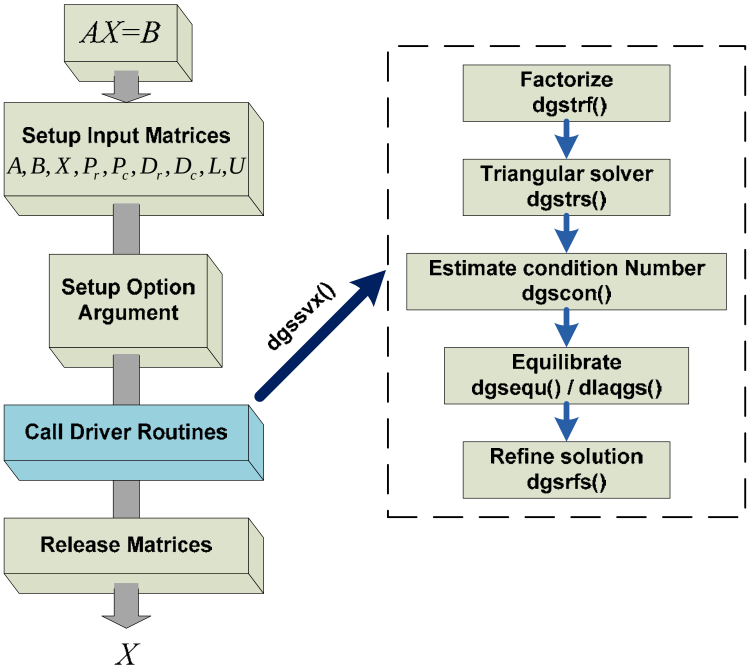
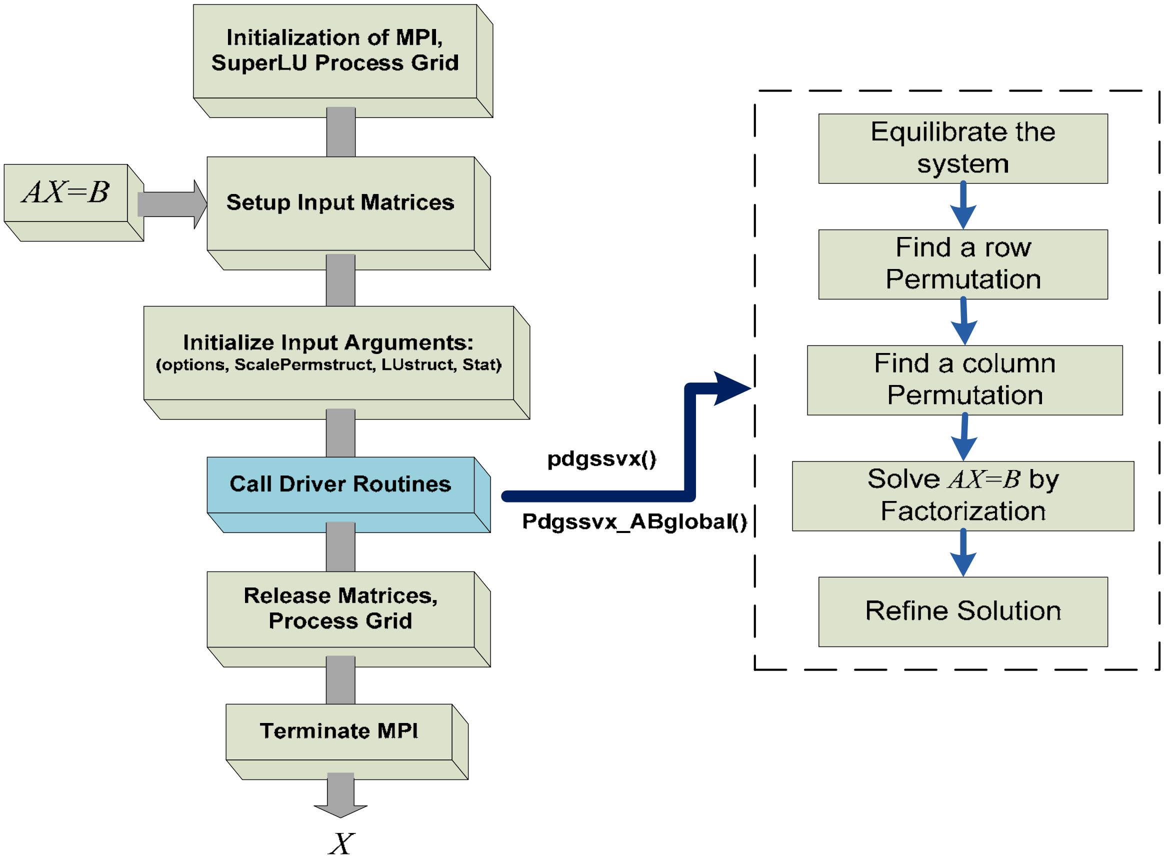


Figure 38. Computational routines of sequential SuperLU

6.3 Distributed-Memory Parallel SuperLU

Figure 39 illustrates the basic routines for parallel SuperLU with MPI. There are mainly two driver routines to solve systems of linear equations, pdgssvx_ABglobal for the global input interface, and pdgssvx for the distributed interface. From the introduction in [45], both of the routines implement following functions, i) equilibrating the system if *A* is poorly scaled, ii) finding a row permutation that makes diagonal of *A* large relative to the off-diagonal, iii)finding a column permutation that preserves the sparsity of the *L* and *U* factors, iv) solving the system $AX = B$ for *X* by factoring A followed by forward and back substitutions, v) refining the solution. The difference between the sequential SuperLU and distributed-memory parallel SuperLU is that the matrices *L* and *U* in parallel SuperLU are distributed in a two-dimensional and him and him and him and him block-cyclic fashion so that the linear equations can be solved in parallel. The configuration on parallel computing is initialized by SuperLU process grid. More details about the data structure used in parallel SuperLU are elaborated in paper [45].



Initialization of MPI, SuperLU Process Grid
AX=B
Setup Input Matrices
Initialize Input Arguments:
(options, ScalePermstruct, LUstruct, Stat)
Call Driver Routines
pdgssvx()
Pdgssvx_ABglobal()
Release Matrices, Process Grid
Terminate MPI
X
Equilibrate the system
Find a row Permutation
Find a column Permutation
Solve AX=B by Factorization
Refine Solution

Figure 39. Basic routines of distributed-memory parallel SuperLU

6.4 Simulation Results About SuperLU Performance In HSET-TDS

The performance of a linear solver has great impact on time-domain simulation, since the solution of linear equations constructs the most part of Newton-Raphson method. HSET-TDS includes many linear solver libraries, and the performance of SuperLU can be very efficient. Table 8 and Figure 40 illustrate the performance of different linear solver in HSET-TDS, which are simulated on the New England system described in the section 3.4 with same contingency and simulation time. Other linear solvers which are compared with SuperLU include dense LU factorization and sparse generalized minimum residual method (GMRES) from GMM++.

The numerical method used in Sparse GMRES solver is an iterative method, which can deal with the ill-conditioned matrix A in solving AX=B. The ill-conditioning of matrix A is possibly lead by the parameters of different degree level in some electric elements. Besides, when the solution vector from Newton method is close to the final solution, the some items of the Jacobian matrix can be very small, and under this circumstance the matrix A may be ill-conditioning. The iterative method in the solution of linear equations can be more effective to deal with ill-conditioning problem compared with direct solution.

Table 8. Comparison between SuperLU, Dense LU and Sparse GMRES

Integration Methods		Dense LU	Sparse GMRES	SuperLU
Trapezoidal Rule (h = 0.01s)	Time(s)	5275.368	1542.814	33.234
	Speed-up	1	3.4193	158.73
Trapezoidal Rule (variable step)	Time(s)	3942.143	1202.29	58.624
	Speed-up	1	3.28	67.24
HH4 (h = 0.1s)	Time(s)	4001.47	707.955	9.328
	Speed-up	1	5.65	428.97
HH4 (variable step)	Time(s)	6572.406	1366.27	19.046
	Speed-up	1	4.81	345.08

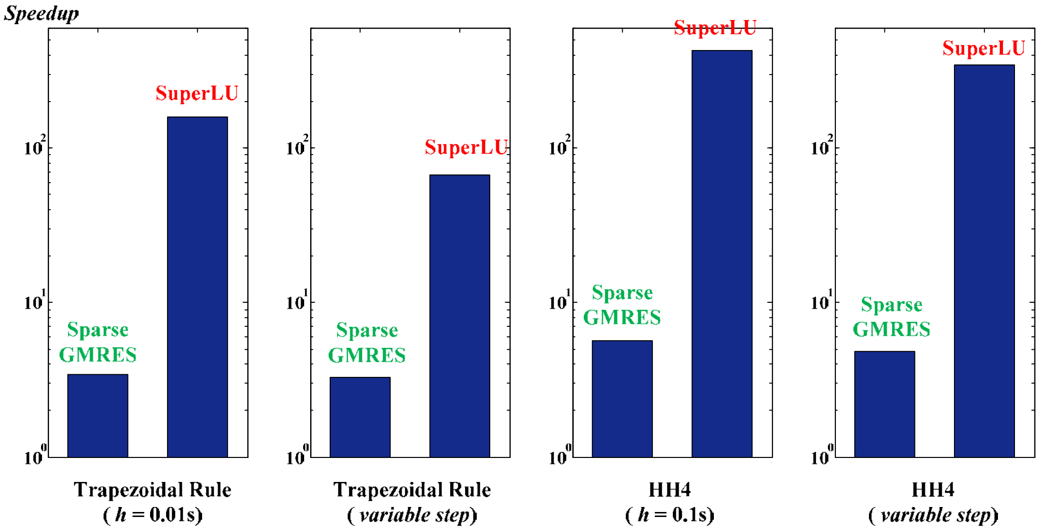


Figure 40. Speedup of SuperLU, Sparse GMRES to Dense LU

From the simulation results in Table 8 and Figure 40, we can find that SuperLU is highly more efficient than the dense LU factorization and sparse GMRES. Since the SuperLU linear solver library fully considers the sparsity of the matrix A and the Jacobian matrix in power system problems are usually highly sparse, it is natural that the efficiency of SuperLU will be much higher than the dense LU factorization. For GMRES, there are two possible reasons why SuperLU is faster. The first is that the algorithm used in GMRES needs more computation because of the attribute of iteration, while SuperLU is more like direct solution, and iteration will also be needed when too large error is produced. The second reason is about different programming approaches, such as how to make use memory, which may lead to different performance.

7. Conclusion

7.1 Summary

A new control center functionality called high-speed extended term (HSET) time-domain simulation (TDS) was proposed for the online analysis of cascading. Key to this functionality is computational speed. The work reported uses state-of-art algorithms to identify maximum on-line computational speed for HSET-TDS. The computational speed is intended to be enhanced from five hierarchical aspects, hardware, strategies, integration methods, nonlinear solvers and linear solver libraries. The work elaborated in this report can be generalized as follows.

Hammer-Hollingsworth 4 (HH4), different from the traditional integration method Trapezoidal rule, which is adopted by many commercial software, is an important integration method used in HSET-TDS. HH4 not only shares the same advantageous attribute of Trapezoidal rule, symmetrical A-stability, which can cope with the stiffness and hyper-stability problems, but also possesses the attribute of higher precision (h^4). The higher precision make it possible to enlarge the integration step, and therefore the computational efficiency can be improved. The simulation results show that HH4 is a suitable integrator for time domain simulation of power system, and is superior to Trapezoidal rule.

Linear solver library is an indispensible part when solving non-linear equations by Newton method. SuperLU, an open source linear solver library from Berkeley, is introduced and applied in HSET-TDS. The goal of the application of SuperLU in HSET-TDS is to expand the linear solver libraries in HSET-TDS, especially for the parallel version. The simulation results of sequential computing shows that SuperLU is competitive and suitable for time domain simulation of power system.

7.2 Contributions

The contribution of this work achieved and expected is summarized as follows.

A new control center software called HSET-TDS has been developed. HSET-TDS is of two versions, sequential and parallel version. And in each version, it includes many different numerical methods which are intended to fully enhance the computational speed. The attribute of high speed computation of HSET-TDS make it possible and practical to analysis high consequence event such as cascading, which will be beneficial for the operation of control center.

A new integration method, Hammer-Hollingswoth 4 (HH4), was first applied in the field of power system. Comparing with the traditional integration method, Trapezoidal rule, which is adopted by many commercial software, HH4 is not only symmetrical A-stable, but also highly precise, and therefore the integration step can be enlarged. HH4 can be a very efficient integrator for time domain simulation of power system.

An new algorithm design for deployment on BlueGene/L was proposed, which is intended to enhance the computational speed with the following techniques i) projection method for stiffness detection and decoupling, ii) waveform relaxation to decouple stiff part, iii) integration methods to decrease integration step, and iv) parallel linear solver.

An new efficient stiffness decoupling strategy has been explored. The main idea of the scheme is to detect the stiffness of ODE by the numerical error, and this technique is more efficient than the stiffness detection strategy by seeking eigenvalues of Jacobian matrix. The stiffness decoupling strategy makes it possible to solve the stiff part of ODE by implicit methods and solve the non-stiff part of ODE by explicit methods. Since explicit methods are much more efficient than implicit method, the computation speed of time domain simulation can be further developed.

7.3 Future Work

We have reported on an algorithmic design for deployment on high-performance parallelized computer architectures shown in section 2.8.

- Implementation of the proposed design will be taken as the next step.
- The exploration of sequential and parallel linear solver libraries, such as SuperLU, UMFpack, GMM++, MUMPS, LAPACK will be carried on.
- The comparison of these available linear solver libraries will be based on HSET-TDS. The simulation results can supply helpful information for industry regarding which sparse linear solvers are suitable for the time domain simulation of power system.

References

[1] Q. Chen and J. McCalley, “Identifying High-Risk N-k Contingencies for On-line Security Assessment,” IEEE Trans. on Power Systems, Vol. 20, Issue 2, May 2005 pp. 823 – 834.

[2] Q. Chen, “The probability, identification and prevention of rare events in power system” PhD Thesis, ECE, Iowa State University, 2004.

[3] S. K. Khaitan, “On-line cascading event tracking and avoidance decision support tool” PhD Thesis, ECE, Iowa State University, 2008.

[4] S. Khaitan, C. Fu and J. McCalley, “Fast Parallelized Algorithms for On-Line Extended-Term Dynamic Cascading Analysis”, invited paper at IEEE PES Power Systems Conference and Exposition, Seattle WA, March 15-18, 2009.

[5] B.Stott. “ Power system dynamic response calculations,” Proc, IEEE, vol. 67, pp. 219-240, Feb. 1979.

[6] Taoka H., Abe S., Takeda S., “Fast Transient Stability Solution Using An Array Processor”, IEEE Transactions on Power Apparatus and Systems Volume PAS-102, Issue 12, Dec. 1983 Page(s):3835 – 3841.

[7] De Mello F.P., Feltes J.W., Laskowski T.F., Oppel L.J., “Simulating fast and slow dynamic effects in power systems”, Computer Applications in Power, IEEE, Volume 5, Issue 3, July 1992 Page(s):33 -38.

[8] EPRI EL 4610, “Extended Transient Midterm Stability Program”, Jan, 1987.

[9] J. J. Sanchez-Gasca, R. D’Aquila, W.W. Price, J.J. Paserba, “Variable time step, implicit integration for extended-term power system dynamic simulation”, Power Industry Computer Application Conference, 1995. Conference Proceedings., 1995 IEEE, 7-12 May 1995 Page(s):183 – 189.

[10] Astic J.Y., Bihain A., Jerosolimski M., “The mixed Adams-BDF variable step size algorithm to simulate transient and long term phenomena in power systems”, Power Systems, IEEE Transactions on Volume 9, Issue 2, May 1994 Page(s):929 – 935.

[11] Fillatre O., Evrard C., Paschini D., Bihain A., Karoui K., Antoine J.P., “A powerful tool for dynamic simulation of unbalanced phenomena”, Advances in Power System Control, Operation and Management, 1997. APSCOM-97. Fourth International Conference on (Conf. Publ. No. 450), Volume 2, 11-14 Nov. 1997 Page(s):526 - 531 vol.2.

[12] Crow, M.L.; Chen, J.G.; "The multi-rate method for simulation of power system dynamics", Power Systems, IEEE Transactions on Volume 9, Issue 3, Aug. 1994 Page(s):1684 – 1690.

[13] Crow, M.L.; Chen, J.G.; "The multi-rate simulation of FACTS devices in power system dynamics", Power Systems, IEEE Transactions on Volume 11, Issue 1, Feb. 1996 Page(s):376 – 382.

[14] Gross G., Bergen A., “An efficient algorithm for simulation on transients in large power systems”, Circuits and Systems, IEEE Transactions on Volume 23, Issue 12, Dec 1976 Page(s):791 – 799.

[15] Gross, G., Bergen, A.R., “A class of new multistep integration algorithms for the computation of power system dynamical response”, Power Apparatus and Systems, IEEE Transactions on Volume 96, Issue 1, Part 1, Jan. 1977 Page(s):293 – 306.

[16] D. Yang, V. Ajjarapu, “A decoupled time-domain Simulation method via Invariant subspace partition for power system analysis,” IEEE Trans on Pwr Sys., Volume 21, Issue 1, Feb. 2006 Page(s):11 – 18

[17] Ward Cheney, David Kincaid, “Numerical mathematics and computing”, Pacific Grove, CA, Brooks/Cole Pub. Co., c1999.

[18] A. Zecevic and N. Gacic, “A partitioning algorithm for the parallel solution of differential-algebraic equations by waveform relaxation” Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on Volume 46, Issue 4, April 1999 Page(s):421 – 434

[19] E. Lelarasmee, A. Ruehli, and A. Sangiovanni-Vincentelli, “The waveform relaxation method for time-domain analysis of large-scale integrated circuits,” IEEE Trans. Computer-Aided Design, vol. 1, pp. 131–145, July 1983.

[20] Crow M.L., Ilic M.D., “The waveform relaxation method for systems of differential/algebraic equations”, Decision and Control, 1990., Proceedings of the 29th IEEE Conference on 5-7 Dec. 1990 Page(s):453 - 458 vol.2.

[21] An IEEE Commitee Report, "Parallel processing in power systems computation", Power Systems, IEEE Transactions on Volume 7, Issue 2, May 1992 Page(s):629 – 638.

[22] http://home.gna.org/getfem/gmm_intro.html

[23] http://www.cs.berkeley.edu/~demmel/SuperLU.html

[24] Decker I.C., Falcao D.M., Kaszkurewicz E., “Conjugate gradient methods for power system dynamic simulation on parallel computers”, Power Systems, IEEE Transactions on Volume 11, Issue 3, Aug. 1996 Page(s):1218 – 1227.

[25] Decker I.C., Falcao D.M., Kaszkurewicz E., “Parallel implementation of a power system dynamic simulation methodology using the conjugate gradient method”, Power Systems, IEEE Transactions on Volume 7, Issue 1, Feb. 1992 Page(s):458 – 465.

[26] W. Xue, J. Shu, Y. Wu, W. Zheng, “Parallel algorithm and implementation for realtime dynamic simulation of power system” , Parallel Processing, 2005. ICPP 2005, International Conference on 14-17 June 2005 Page(s):137 – 144.

[27] P.Amestoy, I. Duff, J. L'Excellent, “Multifrontal parallel distributed symmetric and unsymmetric solvers,” ENSEEIHT-IRIT Tech. Report, revision apprd in Cmput. Mthds Appl. Mech. Eng., 184, 501-520, 2000.

[28] http://graal.ens-lyon.fr/MUMPS/

[29] Paul M. Anderson, A. A. Fouad, “Power system control and stability, the Institute of Electrical and Electronic Engineers”, Inc., 1994.

[30] Yixin Ni, Shousun Chen, Baolin Zhang, “Theory and analysis of power system dynamic”, Tsinghua University publication house, 2002.

[31] Kundur, P. (Prabha), “Power System Stability and Control”, New York : McGraw-Hill, c1994.

[32] EPRI EL 4610, “Extended Transient Midterm Stability Program”, Jan, 1987

[33] SIMENS, “Power Transmission and Distribution PSS/E TM 30.2 Online Documentation”, Nov, 2005

[34] Schulz, R. P. Synchronous machine modeling. Symposium on Adequacy and Philosophy of Modeling: System Dynamic Performance. IEEE Publ. 75 CH 0970-PWR, 1975.

[35] E. Hairer, G. Wanner, “Solving ordinary differential equations II: stiff and differential-algebraic problems”, Springer-Verlag, 1996.

[36] E.Hairer, S.P.Norsett, G. Wanner, “Solving ordinary differential equations I: Nonstiff Problems”, Springer-Verlag, 1987.

[37] J. J. Sanchez-Gasca, R. D’Aquila, W.W. Price, J.J. Paserba, “Variable time step, implicit integration for extended-term power system dynamic simulation”, Power Industry Computer Application Conference, 1995. Conference Proceedings., 1995 IEEE, 7-12 May 1995 Page(s):183 – 189.

[38] M. Berzins, R.M. Furzeland, “An Adaptive Theta Method for the Solution of Stiff and Non-stiff Differential Equations”, App. Num. Math., Vol 9, pp 1-19, 1992

[39] Willem Hundsdorfer, Valeriu Savcenco, “Analysis of a multirate theta-method for stiff ODEs”, Applied Numerical Mathematics, Volume 59, Issues 3-4, March-April 2009, Pages 693-706

[40] Alvarado F.L., Lasseter R.H., Sanchez, J.J., “Testing Of Trapezoidal Integration With Damping For The Solution Of Power Transient Problems”, IEEE Transactions on Power Apparatus and Systems Volume PAS-102, Issue 12, Dec. 1983 Page(s):3783 – 3790.

[41] P.C. Hammer, J.W. Hollingsworth, “Trapezoidal methods of approximating solutions of differential equations”, MATC, volumn 9, 1959, Page(s) 92-96.

[42] G.M. Shroff, H.B. Keller, "Stabilization of unstable procedures: the recursive projection method", SIAM Journal on Numerical Analysis, vol.30, pp.1099-1120, August, 1993.

[43] V. Janovsky, O. Liberda, “Continuation of invariant subspaces via the recursive projection method,” Applications of Mathematics, vol.48, pp. 241-255, 2003.

[44] D. Yang, V. Ajjarapu, “A decoupled time-domain Simulation method via Invariant subspace partition for power system analysis,” IEEE Trans on Pwr Sys., Volume 21, Issue 1, Feb. 2006 Page(s):11 – 18

[45] James W. Demmel, John R. Gilbert, Xiaoye S. Li, “SuperLU Users’ Guide”, Nov, 2007, available at (http://www.cs.berkeley.edu/ ~demmel/SuperLU.html).

[46] James W. Demmel, Stanley C. Eisenstat, John R. Gilbert, Xiaoye S. Li, Joseph W. H. Liu, “A supernodal approach to sparse partial pivoting”, SIAM J. Matrix Analysis and Applications, 1999, Vol 20-3, pp.720-755.

[47] Xiaoye S. Li and James W. Demmel, “Making Sparse Gaussian Elimination Scalable by Static Pivoting”, High Performance Networking and Computing Conference, Orlando, Florida, Nov. 7-13, 1998.

[48] James W. Demmel, John R. Gilbert, Xiaoye S. Li, “An Asynchronous Parallel Supernodal Algorithm for Sparse Gaussian Elimination”, SIAM J. Matrix Analysis and Applications, 1999, Vol 20-4, pp:915-952.

[49] Xiaoye S. Li and James W. Demmel, “A Scalable Distributed-Memory Sparse Direct Solver for Unsymmetric Linear Systems”, ACM Trans. Mathematical Software, 2003, Vol 29-2, pp:110-140.

Part B

Power System Reconfiguration Based on Multi-objective and Multi-level Graph Partitioning

Faculty:

Chen-Ching Liu
University College Dublin, Ireland

Ph.D. Student:

Juan Li
Iowa State University

Information about Part B of this project

For information contact:

Chen-Ching Liu
Professor, School of Electrical, Electronic and Mechanical Engineering
Deputy Principal, College of Engineering, Mathematical and Physical Sciences
University College Dublin
Belfield, Dublin 4
IRELAND
353-1-716-1676 (Phone)
353-1-283-0921(Fax)
liu@ucd.ie

Power Systems Engineering Research Center

The Power Systems Engineering Research Center (PSERC) is a multi-university Center conducting research on challenges facing the electric power industry and educating the next generation of power engineers. More information about PSERC can be found at the Center’s website: http://www.pserc.org.

Power Systems Engineering Research Center
Arizona State University
577 Engineering Research Center
Tempe, Arizona 85287-5706
Phone: 480-965-1643
Fax: 480-965-0745

Notice Concerning Copyright Material

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Overview

Analysis of large scale blackouts in North America, Europe, and other countries shows that following the initiating contingencies, cascaded events may occur, leading to catastrophic power outages. It is important to take proper remedial actions to alleviate the vulnerable operating conditions in a power system in order to avoid a catastrophic outage. Area partitioning that splits a power network into self-sufficient islands is an emergency control to stop the propagation of disturbances and avoid cascading failures.

Modern power systems are designed and equipped to incorporate the "grid protection" control actions, such as special protection schemes (SPS) and remedial action schemes (RAS) against the cascading failures [1]. The real challenge for the future "grid protection" control actions is to evaluate the system vulnerability and response to the disturbance and vulnerable operating condition in real-time. Power system reconfiguration is an important part of the future "grid protection" control actions. However, most of existing power system reconfiguration schemes determine the system partitioning solution based on off-line studies that may not reflect the real time system operation condition and may eventually lead to a poor partitioning plan. The main objective of this research project is to develop an efficient Area-Partitioning algorithm that can partition the power network into k-disjoint areas via minimizing the generation load imbalance in each area. The proposed algorithm must be computationally efficient so that it is feasible to determine the optimal partition in a real-time environment.

1.2 Patterns of Cascaded Events

1.2.1 General Sequence of Events

In general, the sequences of events in the major blackouts followed a common process. Typically, the cascaded events were initiated by a single event or multiple events, such as the 500-kV line outage (U.S. 1996), the generator tripping and the 345-kV line outage (U.S. and Canada 2003 [2]), the line outage (Italy 2003) and the coupling operation of busbars at a substation (Europe 2006).

Following the initiating events, the cascaded events took place sequentially. One component failure may trigger another event, which can bring successive line tripping and/or generator tripping. These subsequent events lead to power flow rerouting, overloads, voltage instability, and angle instability, which further weaken the system. As the cascaded events proceed, angle instability, power oscillations, and significant imbalances between the power supply and demand may take place, leading to a widespread voltage collapse and uncontrolled system splitting.

An excessive imbalance between load and generation after system splitting further caused a severe frequency drop [3] and ultimately led to a large-scale outage [4]. During the early stage of cascaded events, the cascading process can proceed at a relatively slow speed [5], such as in the U.S. and Canada 2003 blackout [2]. However, once a critical point is reached, the successive load and generator tripping events can spread promptly and uncontrollably [6]. At this point, a large-scale blackout is inevitable.

1.2.2 Weakened System Conditions

A weakened system condition is a contributing factor to four major blackouts shown in Table 1. When the system is highly stressed, power flows are high, the system voltage may decline and the power network components are highly loaded. As a result, the interactions between component failures tend to be stronger, and a single event is more likely to trigger other subsequent events that lead to a large blackout [7]. A summer peak-load profile, heavily loaded transmission lines, out-of-service transmission facilities due to the scheduled maintenance can further weaken the system.

Table 1 Major Blackouts in North America and Europe

Date	Location	Scale in terms of GW and Population	Collapse Time
Aug.10th, 1996	US-Western Interconnection	30.5 GW, 7.5M people	> 6 minutes
Aug.14th, 2003	Northeast US & Eastern Canada	62 GW, 50M people	> 1 hour
Sep. 28th, 2003	Italy	27GW, 57M people	> 25 minutes
Nov. 4th, 2006	Europe	17GW, 15M people	> 20 seconds

1.2.3 Patterns of Cascaded Events in Blackouts

There are many causes of cascaded events that contribute to catastrophic outages. They typically include faults, undesired relay operations (including hidden failures), equipment failures or malfunctions, communication and information failures, and operational errors. Due to the mixture of the causes, prediction of the exact sequence of cascaded events that will take place is practically impossible. However, it is important to look into the fundamental patterns of cascaded events, i.e., which event can trigger other events. Some examples of fundamental patterns of cascaded events are:

- Line Tripping due to Overloading [2], [8]
- Generator Tripping due to Over-Excitation [2]
- Line Tripping due to Loss of Synchronism [2], [6], [10]
- Generator Tripping due to Abnormal Voltage and Frequency System Condition [2], [5], [6], [9]
- Under-Frequency/Voltage Load Shedding [6], [8]

1.3 Prevention of Cascaded Events by Power System Reconfiguration

After the initiating contingency, cascaded events may occur one by one. Further contingencies may weaken the operating condition and the system may lose synchronism and be separated into several islands. Those islands are usually accompanied by severe imbalance of the generation and load. If the generation and load within the island cannot be balanced properly, the system frequency may violate constraints and lead to further line and generator tripping events. Disintegration of the islands may evolve and finally blackouts occur.

Unlike the islands formed by uncontrolled separation, the islands determined by the proposed power system reconfiguration concept can minimize the imbalance between generation and load. Therefore, the system frequency within the island can be regulated within allowable limits and the islands formed by controlled separation are less prone to a system collapse. If the power system reconfiguration scheme is conducted appropriately, the cascading events and large scale blackouts can be avoided.

A few methods have been proposed to determine the optimal islanding configuration. A three-phase method utilizing ordered binary decision diagrams (OBDDs) to find proper islanding strategies is proposed in [11]. An automatic islanding approach that determines the islands from the identified slowly varying coherent groups of generators is reported in [12]. A partition strategy using minimal cut sets with minimum net flow is proposed in [13]. A spectral k-way partition algorithm is provided in [14]. Based on the generator grouping information, a two step partitioning approach including graph simplification and multilevel k-way partition is presented in [15] and qualitatively demonstrated on the August 14, 2003 blackout scenario [16].

1.4 Contents of this Part

Section 2 gives an introduction to graph partitioning as a graph-theoretic problem. An overview of multi-level graph partitioning and multi-objective graph partitioning is provided.

Section 3 introduces the proposed graph theoretic area partitioning scheme, conceptual special protection scheme and architecture of the emergency control system. Advantage of using multi-level graph partitioning algorithm to determine the optimal system islanding is discussed.

Section 4 presents the proposed graph theoretic area partitioning scheme that considers both real and reactive power balance. A framework of two stage area partitioning algorithm is provided.

Section 5 presents an application of the proposed area partitioning algorithms for preventing cascading events on a 200 bus system model. Simulation results show that the proposed area partitioning algorithm minimizes both real and reactive power imbalance within the islands, which helps to improve the voltage profile after islanding and mitigate the possibility of further cascading. The proposed area partitioning is also applied on a 22,000 bus system. Simulation results show that the proposed algorithm is computationally efficient. Therefore, it is feasible to determine the partitioning strategy and identify the system islanding configuration in a real-time environment.

Section 6 provides the conclusions of this research and important issues that need to be addressed in the future.

2. Graph Theory Foundation

An important objective in partitioning a power network is to identify optimal cut sets of the sub-networks while the electrical interdependency between sub-networks and the power imbalance within sub-networks are minimized. By graph theory, this task involves partitioning of a weighted graph into several smaller graphs with evenly distributed weights and minimized edge cut sets.

2.1 Multilevel Graph Partitioning

Graph partitioning with Minimum Ratio Cut is known to be NP-complete and it has many applications in scientific and engineering areas. Many graph partitioning algorithms have been developed. However, some of the algorithms are expensive for large graphs in terms of CPU time. The multilevel graph partitioning scheme [17] is a state-of-the-art technique that significantly reduces the computation time while generating high quality partitions. The multilevel partitioning scheme does not partition the original graph directly. Rather, it first reduces the size of graph through a number of levels by collapsing vertices and edges. Then, the condensed graph is partitioned. Finally, a procedure is used to propagate and refine the solution through successive levels to the original graph.

Consider a weighted graph $G_0 = (V_0, E_0)$. The three stages of a multilevel partitioning scheme, as shown in Fig. 1, are summarized here.

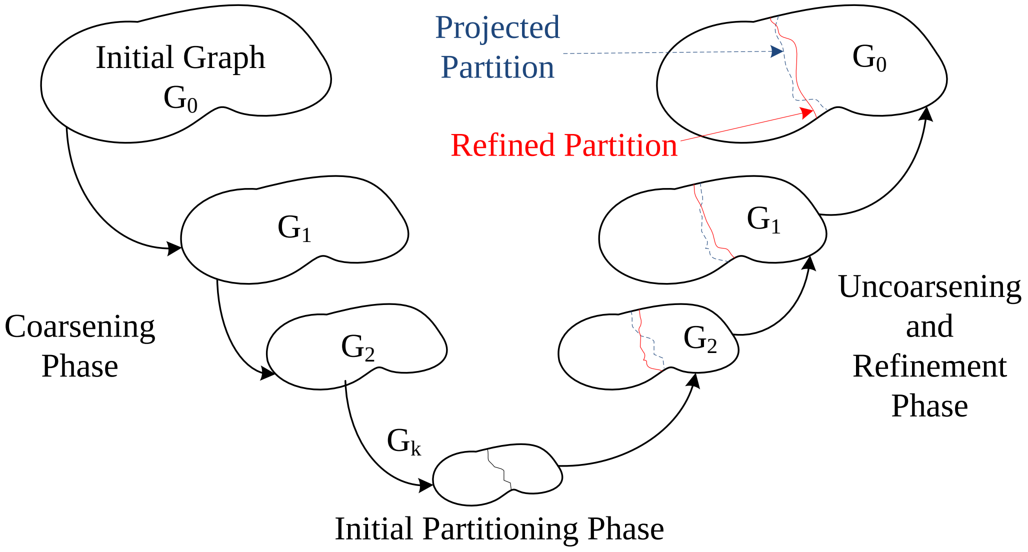


Fig. 1 Multi-level Graph Partitioning

1. **Coarsening Phase.** The initial graph G_0 is transformed into a sequence of smaller graphs G_1, G_2… G_k such that $|V_0| >|V_1| > |V_2| > … > |V_k|$. A simple way to obtain the coarse graph is to group the vertices of the graph into disjoint clusters and collapse the vertices of each cluster into a single vertex.

2. **Partitioning Phase.** A partition P_k of the coarsest graph $G_k = (V_k, E_k)$ that minimizes the edge cut and satisfies the balancing constraints is computed. Since the size of

coarsest graph G_k is small, various partition algorithms, e.g., recursive bisection, can be used to obtain the partition P_k.

3. **Uncoarsening and Refinement Phase.** The partition P_k of the graph G_k is successively projected back to the original graph G_0 by going through intermediate graphs G_{k-1}, G_{k-2}… G_1, G_0. At each step of the uncoarsening phase, the partition is further refined to reduce the cut set and improve the quality of solution.

2.2 Multi-Objective Graph Partitioning

The traditional graph partitioning algorithm is only able to find the optimal partition with a single optimization objective. However, in many engineering applications, there is a need to produce partitioning with multiple optimization objectives. One example is the problem of minimizing the number of wires cut by the partitioning as well as the propagation delay between the chips in the VLSI domain. This kind of problem can be formulated as a graph partitioning problem in which every edge in the graph will be assigned with multiple weights, and the partitioning objectives will minimize the edge-cut with respect to each of the multiple weights.

The multi-objective optimization is challenging since an optimal solution for one objective is not necessary optimal for another. In the multi-objective graph partitioning problem, different types of edge weights increase the difficulty in finding the optimal solution. The simple combination-based approach, such as adding two weights to a single weight, does not make sense. In [18], an algorithm is proposed that can handle both similar as well as dissimilar edge weights, allowing tradeoff among different objectives and resulting in predictable partitioning. This algorithm combines multiple objectives into a single objective in a scientific way and then applies a single objective graph partitioning algorithm to conduct partitioning. The way to combine multiple objectives is based on the intuitive notion of what constitutes a good multi-objective partitioning. The partitioning solution that is close to each optimal partitioning with respect to the single objective is considered to be good. The basic procedure to conduct a two-objective partitioning is shown below,

1. A single objective graph partitioning for the first objective is conducted. The best edge-cuts C_1 for the first objective is obtained.

2. A single objective graph partitioning for the second objective is conducted. The best edge-cuts C_2 for the second objective is obtained.

3. A combined weight for each line is obtained by the sum of the normalized edge weight with best edge-cuts C_1 and C_2 and weighted by the controllable preference factor p. The preference factor p should be within the range of 0 and 1.

$$w_{combined} = \frac{w_1}{c_1}p + \frac{w_2}{c_2}(1-p) \qquad (1.1)$$

The distance that each edge-cut is allowed to stray from the optimal solution is controlled by the preference factor p. Thus, the preference factor p controls the tradeoffs among different objectives. Since all edge weights are normalized with a corresponding objective, they present a fraction of the optimal cutest and can be combined meaningfully [18].

4. A single objective graph partitioning is applied with the new combined normalized edge weights.

3. Power System Reconfiguration with Multi-level Graph Partitioning

Since a power network can be viewed as a weighted graph with buses and transmission lines being vertices and edges, respectively, reconfiguration can be formulated as a graph partitioning problem. The objective of obtaining the system separation strategy with minimum net flow on the islanding boundary can be formulated as a problem of finding the graph partitioning solution with minimum edge cut of the boundary.

3.1 Proposed Area Partitioning Algorithm

Based on the multilevel graph partitioning scheme discussed in section 2.1, an efficient power network area partitioning algorithm is developed [19]. This algorithm does not involve an intentional graph simplification stage for a large scale power network. The mere simplification process would lead to a loss of useful information, further reducing the accuracy of partitioning. A coarsening and uncoarsening process is included in the multilevel graph partitioning algorithm that will be applied.

The proposed algorithm includes two stages. At the first stage, the network is modeled as an edge-weighted graph. Each bus of the power network is a vertex of the graph, each transmission line on the one-line diagram is an edge. The weight of an edge is the absolute value of the MW power flow on the transmission line. The basic graph structure of the power network can be constructed off-line. The final graph can be further updated with information of the topological changes and weights for the edges. Since line flows resulting from contingency analysis are used to determine the islanding configuration, the proposed area partitioning algorithm has the ability to determine the next configuration based on vulnerability considerations.

At the second stage, a multilevel recursive bisection algorithm is used to partition the weighted graph into several isolated areas. In this stage, a circuit partitioning tool, pMETIS that implements a multilevel recursive bisection algorithm is utilized [18]. pMETIS is a professional graph partitioning software tool that has been widely used in VLSI design and its performance in terms of the cut size and computational time is excellent. As an example, it is capable of partitioning a graph with 15,606 vertices and 45,878 edges into 256 subareas with minimum cut set in 3.13 second on a PC [18]. With the fast computational speed, the proposed area partitioning algorithm has the potential to compute the system islanding configuration on-line.

The optimal number k of isolated areas is selected from several area partitioning scenarios ranging from two isolated areas to kmax isolated areas, where kmax is the largest acceptable number of isolated areas that a system can be divided into. The optimal k area partitioning is the one with minimum loss of load among all acceptable partitions. If load shedding is combined with this area partitioning algorithm, the optimal k area partitioning minimizes load shedding compared to all other acceptable partitions.

3.2 Special Protection Scheme Architecture

The controlled islanding actions can be initiated by a special protection system (SPS). A conceptual relaying system for controlled islanding is illustrated in Fig. 2.

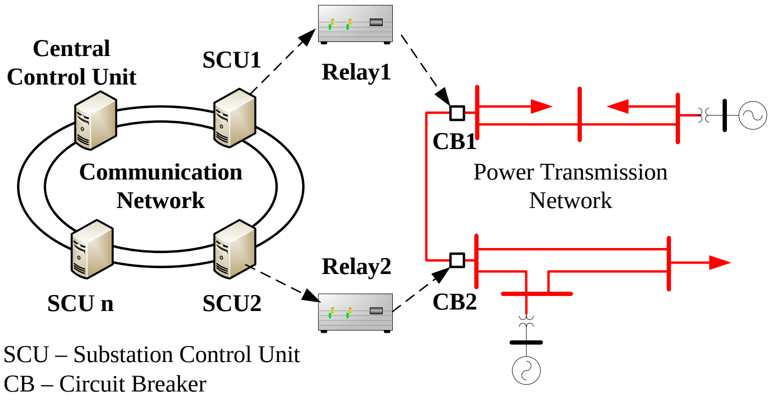


Fig. 2 A Conceptual Relaying Architecture for Controlled Islanding

This system consists of one central control unit (CCU) and several substation control units (SCUs). CCU acquires system data such as topology and power flows from SCUs and generates the system separation strategy using the proposed area partitioning algorithm. Power flow data can be obtained from on-line Energy Management Systems (EMSs). SCUs receive the system separation command from CCU and send breaker opening signals to specific auxiliary relays. These relays then send tripping signals to the appropriate circuit breakers. To ensure reliable and fast information transfer between CCU and SCUs, a dedicated communication network, such as a synchronous optical network (SONET), would be needed.

3.3 Emergency Control and Area Partitioning

The proposed power system reconfiguration scheme consists of a controlled islanding scheme and a load shedding scheme. The procedure shown in Fig.3 demonstrates how the proposed configuration control system can help to absorb a shock, block the propagation of disturbances, and avoid a catastrophic failure.

After the system is separated into islands, load rich or generation rich islands may exist. At a generation rich island, if the system frequency violates the operating constraint, the excess generation can be removed by a rapid response of the speed governor or generator tripping. At a load rich island, load shedding is required to avoid a frequency decline caused by generation shortage.

In response to the disturbance, the optimal configuration and the corresponding boundaries for the islands will be determined by the proposed graph theoretic area partitioning algorithm. If a load rich island exists in the new configuration, a power flow study for the islanding configuration can be performed to determine the amount of load to shed.

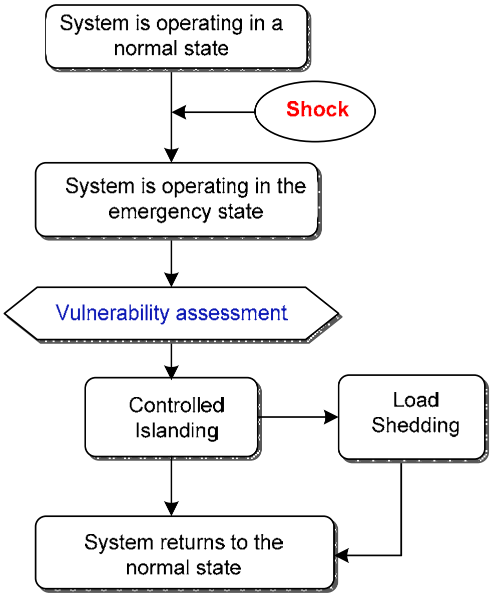

System is operating in a normal state
Shock
System is operating in the emergency state
Vulnerability assessment
Controlled Islanding
Load Shedding
System returns to the normal state

Fig. 3 Emergency Control Procedure

Suppose that the system is initially operating in a normal state. When a major problem occurs in the system, a cascading sequence of events including line tripping, overloading of other lines, protection system malfunctions, and generator tripping events might be triggered. If it is triggered, the system may enter a vulnerable state. If the controlled islanding strategy along with load shedding is applied according to the real time vulnerability assessment result, the impact of disturbances can be isolated to within one island and the remaining system will survive the shock without losing too much load.

4. Power System Reconfiguration Considering Reactive Power Balance

4.1 Introduction of the Methodology

Reactive power plays an important role in supporting the voltage profile of a power system. Insufficient reactive power support leads to a low voltage profile that may result in voltage instability or undesirable line and generator tripping events. The reactive power deficiency in the Idaho area resulted in voltage instability that played a crucial role in the July 2, 1996 blackout [20]. A significant mismatch of the reactive power supply and demand causes high- or low-voltage conditions in the islands [21]. However, reactive power balance has not been incorporated in the development of power system reconfiguration methods. Existing controlled islanding methods are mainly focused on the frequency behavior after system separation through minimization of real power load and generation imbalance within each island.

Real power balance does not necessarily imply that reactive power is also balanced. A small real power flow on islanding boundary lines may be accompanied with a large reactive power flow. Tripping of transmission lines that carry large reactive power flows can lead to insufficient reactive power support in one island, especially in adjacent area of the boundary lines. If the reactive power supply in this area is not increased in time, bus voltages may drop significantly and generators may become overloaded. As a consequence, reactive power demands may eventually exceed the sustainable capacity of reactive power resources, leading to a catastrophic failure.

A balanced reactive power supply and demand in each island helps to improve the voltage profile and enhance the ability of an island to withstand a disturbance. It is desirable to find a system separation plan that keeps the real power balance and, at the same time, avoids insufficient reactive power support caused by tripping of transmission lines on the islanding boundary that carry large reactive power flows. In the proposed algorithm, minimization of the reactive power flow on the partitioning boundary is incorporated in the graph partitioning objectives. In the power system graph, each edge has two distinct weights. One edge weight is the absolute value of the MW power flow on the transmission line and the other is the absolute value of the MVar power flow on the same line. By graph theory, the two objectives in partitioning a power network lead to partitioning of a weighted graph into a number of smaller graphs such that the edge-cut with respect to each different type weight is minimized. Therefore, the power system partitioning problem is transformed into a multi-objective graph partitioning problem.

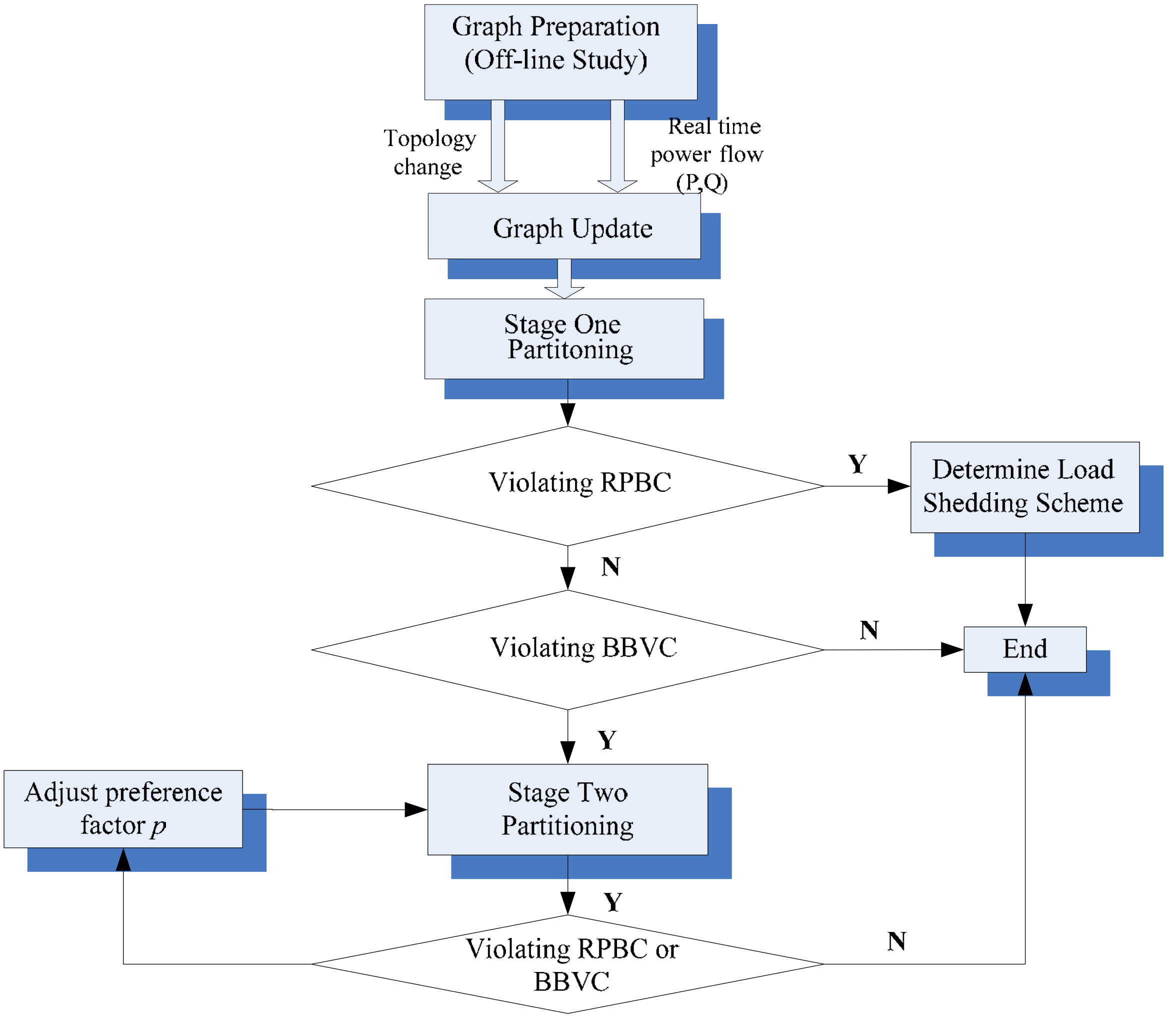
4.2 A Framework of Proposed System Separation Strategy

Since real power balance within the island has to be achieved hand in hand with reactive power balance, a two stage system separation strategy is proposed. There are two constraints for the subsystem formed by system separation:

1. Real Power Balance Constraint (RPBC): In each island, real power generation is equal to or larger than real power load.

2. Boundary Bus Voltage Constraint (BBVC): The bus voltages of the buses at the islanding boundary after system separation should be maintained within tolerances by industry standards (0.95 pu to 1.05 pu for instance).

The flow chart of the proposed system separation strategy is shown in Fig .4.



RPBC – Real Power Balance Constraint
BBVC – Boundary Bus Voltage Constraint

Fig. 4 Proposed Area Partitioning Procedure

1. At Stage one, the area partitioning algorithm only considers minimization of real power flow on the boundary described in section 3.1. Based on the partitioning solution, if RPBC is not satisfied, or if real power load is higher than real power generation in one island, the amount of load to shed is determined and the optimal islanding strategy is the solution obtained by Stage one area partitioning. If RPBC is satisfied, then BBVC is checked. If low bus voltages of the buses at islanding

boundary caused by insufficient reactive power support are expected to occur after islanding, then go to Stage two partitioning.

2. At Stage two partitioning, a multi-objective multilevel graph algorithm is applied. The detailed step-by-step procedure is shown below.

   A. A single objective graph partitioning is computed with the edge weight equal to the absolute value of reactive power flow on the transmission line, and the best edge-cuts C_q for this objective is recorded.

   B. A single objective graph partitioning is computed with the edge weight equal to the combined weight $w_{combined,,}$ where $w_{combined}$ can be obtained from equation 1.1 with the best edge-cuts C_p obtained at Stage one partitioning, the best edge-cuts C_q obtained at the last step and the preference factor *p*. The input preference factor *p* is initiated to reduce the reactive power flow on the boundary.

   C. A single objective graph partitioning is applied with the new combined normalized edge weights.

After Stage two partitioning, RPBC and BBVC are checked again. If both constraints are satisfied, this partitioning solution is chosen as the optimal system separation solution. Otherwise, modify the preference factor *p* and then apply Stage two area partitioning again. The multi-objective graph partitioning algorithm is able to enhance the minimization of real power flow and reduce the ability to minimize reactive power flow on the boundary by increasing the preference factor p. On the other hand, decreasing the preference factor p could reduce the total real power flow on the islanding boundary and increase the total reactive power flow on the islanding boundary. Similar to the proposed algorithm in section 3.1, the multilevel graph partitioning tool, pMETIS[17], is utilized to compute the single objective area partitioning at each step.

5. Numerical Studies and Simulation Result

5.1 Simulation with a 200-Bus System

A 200-bus system that is a variation of the simplified model of the western interconnection in North America is used to evaluate the performance of the proposed power network partitioning algorithm. The one-line diagram of the system is shown in Fig .5.

This system is operating under a peak load condition with a total generation of 64,417.32 MW, 17,236.9 MVar, and a total load of 63,510.41MW and 16,171.75 MVar. The simulation is based on PSS/E power flow solutions and PSS/E time-domain simulation results. A cascading scenario is created on the 200-bus test system, and the sequence of cascading events is as follows,

1) At t=0 second, three transmission lines (196-197, 76-181, 69-76) are out-of-service. (There are two double circuit transmission lines between bus 196 to bus 197 and between bus 69 to 76.)

2) At t=60 second, line 72-197 is de-energized due to the line fault.

3) At t=120 second, line 78-196 is de-energized due to the line fault. Generator G70 at bus 70 become overloaded.

4) At t=240 second, generator G70 hit its field thermal capability limit and is tripped by over-excitation protection.

In this cascading scenario, successive tripping of the lines leads to power flow rerouting and overloads. After tripping of generator G70, power swings become lightly damped. Progressive drops in bus voltage are observed with rotor angle instability. The generator angle difference between generator G77 in the North Island and generator G15 in the South Island, observed in Fig. 6, increases continuously after t=240 seconds and finally exceeds 180° at t=263 seconds approximately. As shown in Fig. 7, the voltage collapse occurs at about 280 second. Since load characteristics are considered in the dynamic model, with the cascading events, the total load in the system is reduced to 61,039.34 MW and 15,227.48 MVar at t=240 second.

5.1.1 System Separation Performance without Considering Reactive Power Balance

In order to prevent the impending blackout, the system is separated into two areas, i.e., North Island and South Island, using the proposed area partitioning algorithm. The performance of system separation without considering reactive power balance is evaluated first. Only real power flow on the partitioning boundary is minimized. In other words, only Stage one graph partitioning described in section 4.2 is conducted.

System islanding is initiated at 1s after tripping of generator G70. The edge cutset and the corresponding absolute value of real and reactive power flow on the islanding boundary are shown in Table 2. The Stage one partitioning boundary is shown in Fig. 5. It can be seen that the system is partitioned into two islands, North Island and South Island, with a roughly equal size.

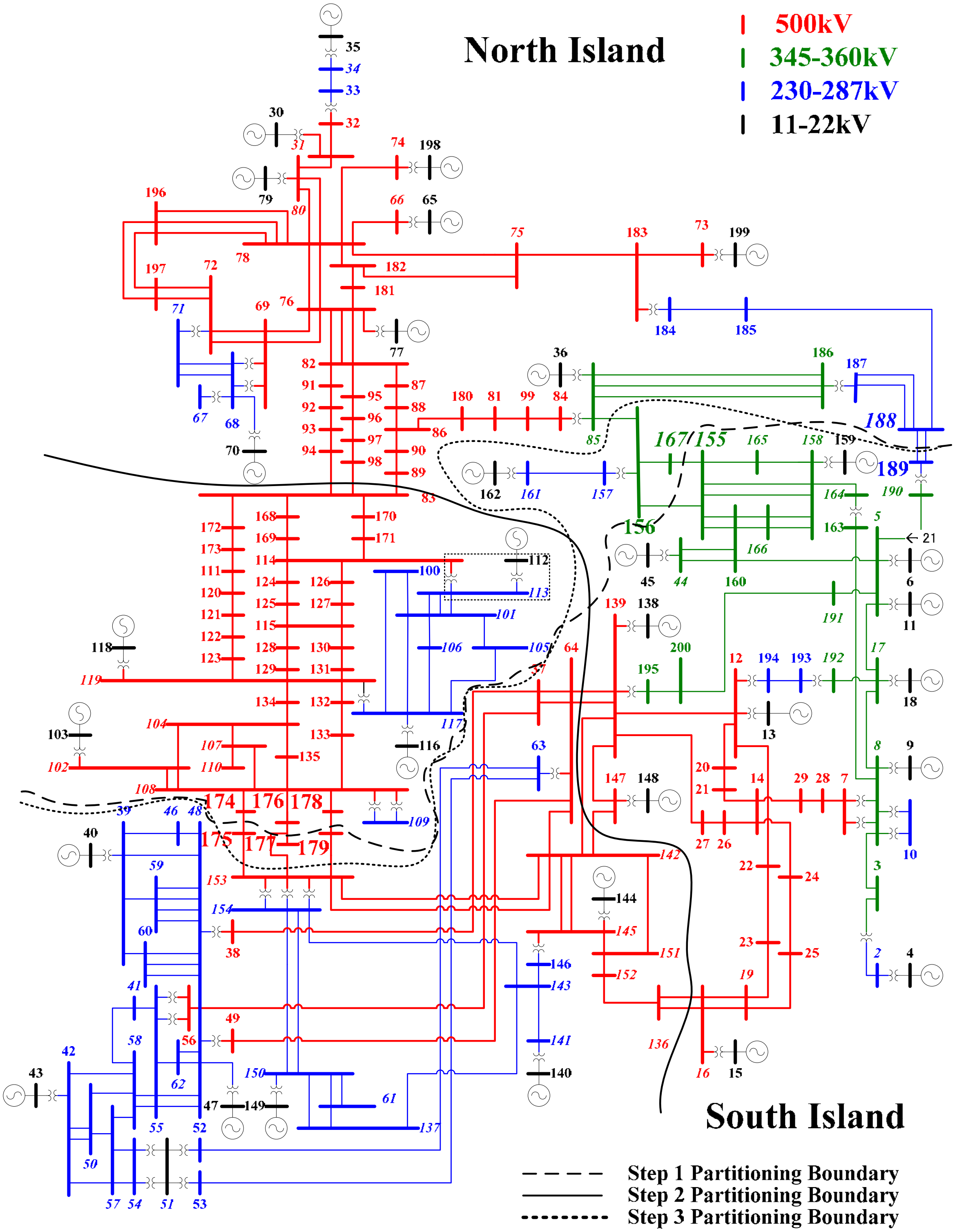


Fig. 5 200-Bus System

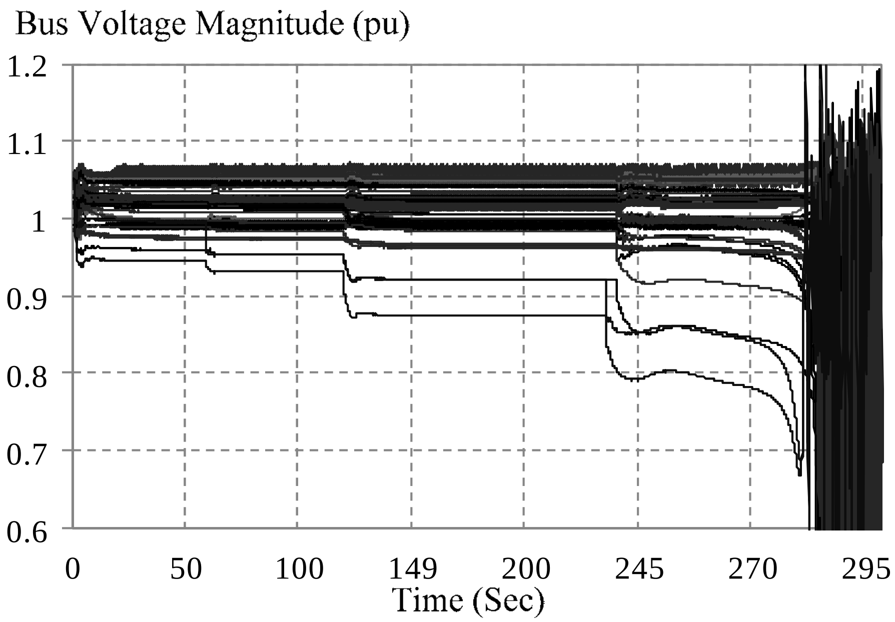


Fig. 6 Bus Voltages without Islanding Strategy

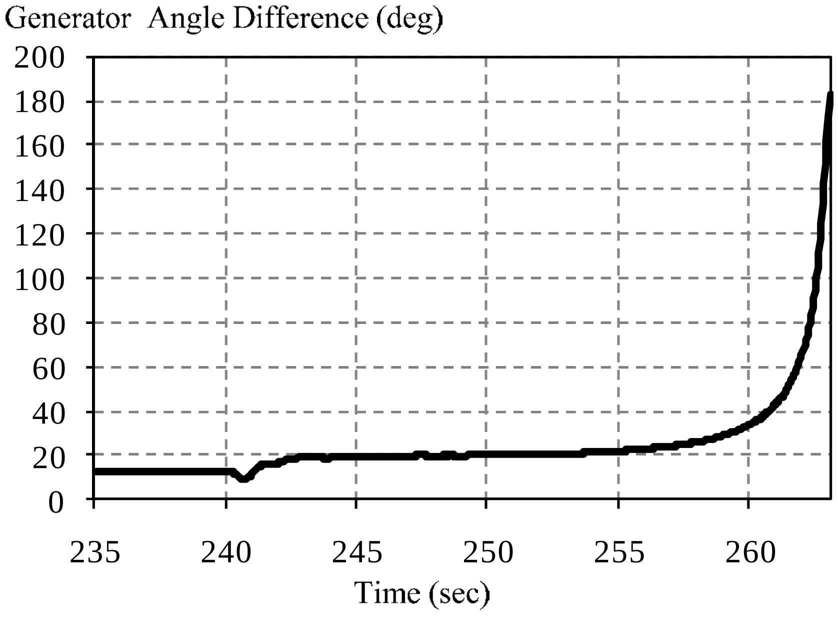


Fig. 7 Generator Angle Difference without Islanding Strategy

The total real power flow on the islanding boundary is 893.5 MW. The total real power load and generation in North Island are 37203.4 MW and 37607.2 MW, respectively. The total real power load and generation in South Island are 23835.9 MW and 24335.5 MW, respectively. Both islands are generation rich islands and satisfy the Real Power Balance Constraint (RPBC). As shown in Fig. 8, after intentional system separation, South Island is successfully stabilized, all bus voltages are higher than 0.95 pu and no line is tripped by the impedance relay.

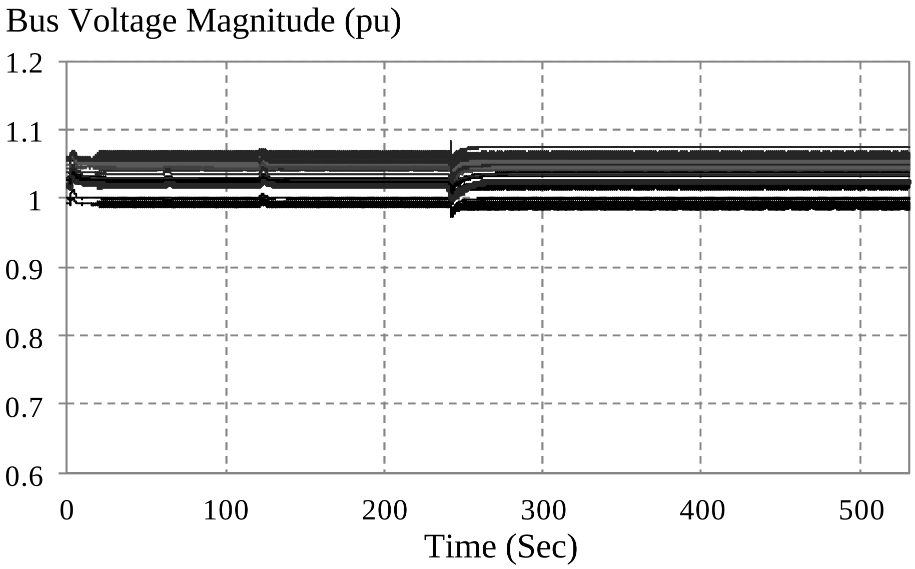


Fig. 8 South Island Bus Voltages

Stage one partitioning does not consider the objective of minimizing the total reactive power on the partitioning boundary. As shown on Table 2, three transmission lines that carry about 1240 MVar reactive power in total are included in the islanding boundary. The direction of the reactive power flow on these three transmission lines is from South to North, which is the opposite direction of the real power flow. Tripping of these transmission lines that carry a large amount of reactive power flow will cause insufficient reactive power support in the area adjacent to the islanding boundary in North Island. As a result, after tripping of transmission lines on the islanding boundary that separates the system into two islands, significant voltage drop occurs at a few buses in the North Island. The bus voltages that are lower than the 0.95 pu in North Island are given in Table 3.

Table 2 Stage One Partitioning Cut Set

Cut Set	174-175	176-177	178-179	156-155	155-167	189-188(1)	189-188(2)	Total
MW	239.2	240.4	255.0	80.0	45.6	17.5	15.8	893.5
Mvar	402.8	405.9	430.8	30.0	41.6	13.9	8.3	1333.3
There are two transmission lines between bus 188 and 189.								

Table 3 Low Bus Voltages in North Island

Bus	104	107	108	111	174	176	178	196
Voltage(pu)	0.934	0.914	0.916	0.908	0.916	0.916	0.916	0.904

As shown in Table 3, bus voltages at eight buses are below 0.95 pu. Although bus 196 is far from the system separation boundary, two 500 kV transmission lines connected with this bus are de-energized in the cascading process that leads to the low voltage profile at this bus. If the reactive power device such as a shunt capacitor is switched on, the low voltage at this bus can be avoided.

Except for bus 196, all other seven buses are close to buses 174, 176 and 178 that are on the system separation boundary. The low bus voltages in the adjacent area of the islanding boundary in North Island cause overloading of the generator at Bus 103. The plot in Fig. 9 shows that the generator field voltage (EFD) at bus 103 is increased to 1.15 pu after system separation. At t=350 second, the generator bus 103 is tripped by over-excitation protection because the generator field voltage at bus 103 continuously exceeds threshold pick-up value of the generator field voltage capability limits for 100 seconds [22]. As soon as the generator at bus 103 is tripped, a voltage collapse occurs at North Island. Fig. 10 shows bus voltages at North Island in the process of cascading events and system separation.

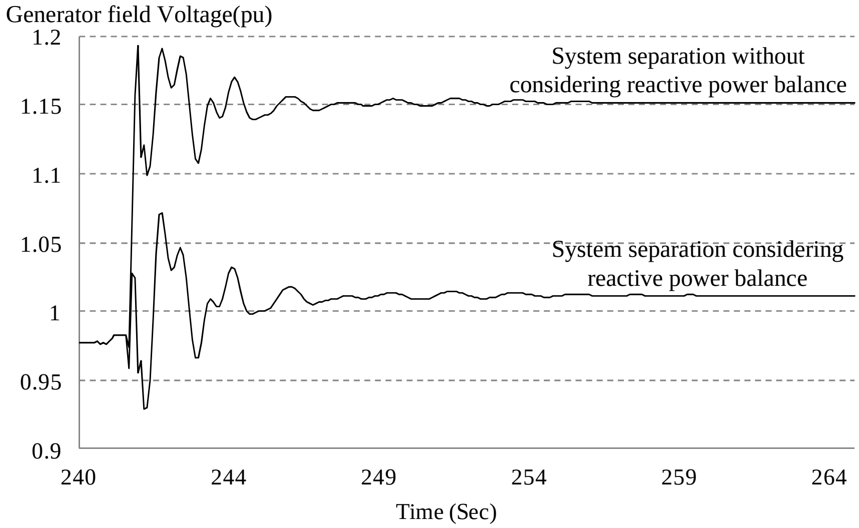


Fig. 9 Generator Field Voltage at Bus 103

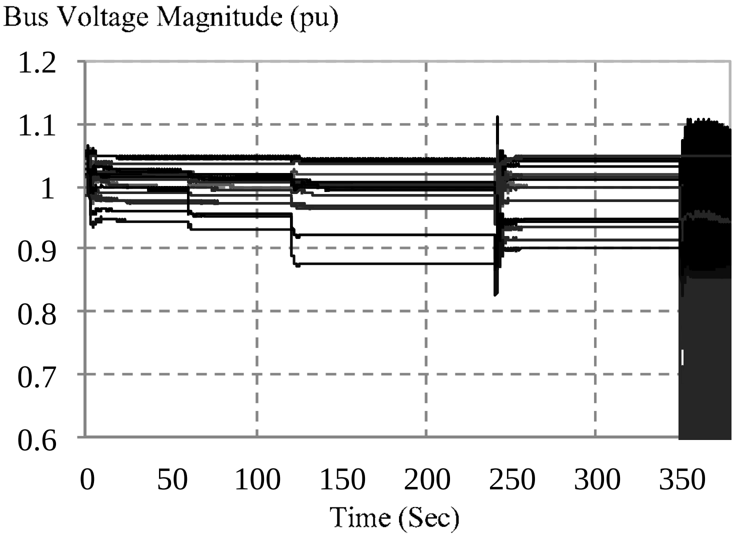


Fig. 10 North Island Bus Voltages

In this simulation case, the system separation strategy without considering reactive power balance successfully prevents a voltage collapse at South Island that has abundant reactive power support but fails to stabilize North Island after the intentional separation.

5.1.2 System Separation Performance Considering Reactive Power Balance

Since there is insufficient reactive power support and low voltage profile exists in the boundary area of North Island after the system separation based on Stage one partitioning, Stage two area partitioning described in section 4.2 is conducted. At first, the area partitioning boundary with the objective of minimizing reactive power flow on the islanding boundary is shown in Fig. 5. The islanding boundary obtained with this objective divides the system into East Island and West Island, which is significantly different from the islanding boundary obtained at Stage one. The total minimized reactive power flow on the islanding boundary is 490.3 Mvar. Note that this number is much lower than reactive power flow on the islanding boundary without considering reactive balance.

The new combined edge weight is calculated with equation 1.1, where $c_1 = c_p$=893.5, $c_2 = c_q$=490.3. The preference factor is selected as 0.7. The calculated system separation boundary with new combined edge weights is shown in Fig. 5. This boundary is close to the boundary obtained at Stage one partitioning geographically. However, the reactive power flows on the partitioning boundary are significantly reduced. The edge cut set and corresponding absolute values of real and reactive power flows and new combined edge weights are shown in Table 4. The total real power load and generation at North Island are 36513.4 MW and 37178.1MW, respectively. The total real power load and generation at South Island are 24525.9 MW and 24764.6 MW, respectively. Both islands are generation rich islands and satisfy the Real Power Balance Constraint (RPBC).

After the system is separated according to the boundary shown in Table 4 at 1s after tripping of generator G70, both islands are stabilized. Fig. 11 shows the bus voltages in both islands. As shown in Fig. 10, only the bus voltage at bus 196 in North Island is stabilized at 0.905 pu that is lower than the 0.95 pu, all other voltages are equal to or higher than 0.95 pu. No generator hits the field capability limits. The Boundary Bus Voltage Constraint (BBVC) is satisfied. Fig. 12 shows that the stable frequency of the North Island and the South Island are 60.05 Hz and 59.97 Hz respectively.

Table 4 Partitioning Cut Set Considering Real and Reactive Power Balance

Cut Set	175-153	177-153	179-153	85-156	189-188(1)	189-188(2)	Total
MW	239.2	240.4	255.0	326.6	17.5	15.8	1094.5
Mvar	188.0	191.8	237.6	78.0	13.9	8.3	717.6
Combined Weight	302.4	305.7	345.2	303.6	22.2	17.5	1296.5
There are two transmission lines between bus 188 and 189.							

This example demonstrates that with the same islanding initiation time, the system separation strategy considering both real and reactive power achieves better performance than the strategy that only considers real power balance. The system separation strategy considering both real and reactive power successfully prevents the post-islanding system collapse that occurs when the partitioning only considers real power balance. The proposed multi-objective multi-level graph partitioning algorithm is able to identify the optimal system separation boundary with minimized real and reactive power flow. The proposed power system reconfiguration scheme enhances the grid's shock absorption capability with respect to the cascading failures.

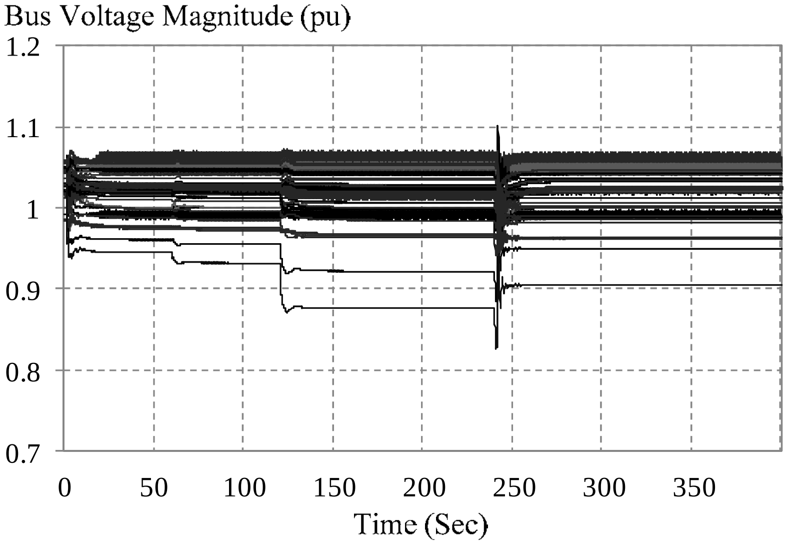


Fig. 11 Bus Voltages with Reactive Power Balancing Islanding

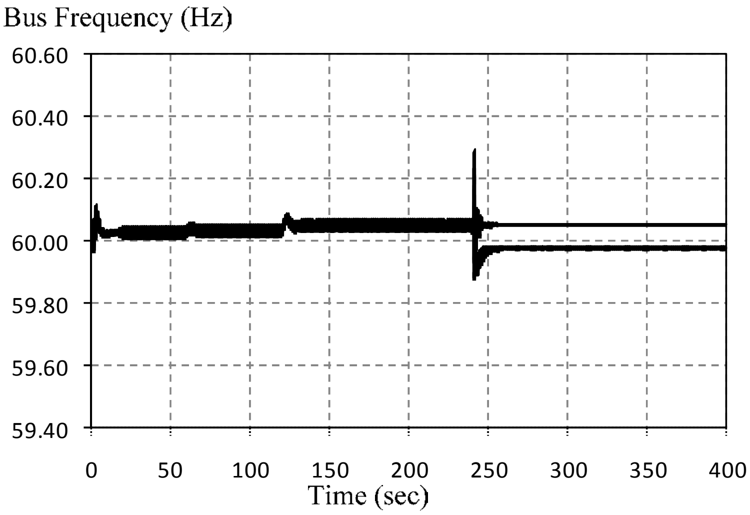


Fig. 12 Bus Frequencies with Reactive Power Balancing Islanding

5.2 Simulation with a 22,000-Bus System

In order to evaluate the computational efficiency of the developed area partitioning algorithm, the proposed algorithm is tested with a 22,000 bus system. After converting this power network into a weighted graph, 22,000 vertices and 32,749 edges are obtained. To partition this graph into 2, 3, 4 islands with one single objective multilevel graph partitioning, the computation time using based on 2 GHz Pentium CPU and 1GB RAM is

0.07s, 0.081s and 0.09s, respectively. Since the proposed multi-objective multilevel graph partitioning algorithm involves three single objective graph partitioning processes, the computation time for this 22,000 bus system with proposed reactive power balance partitioning is less than 1s. The computational speed of the proposed algorithm is expected to meet the real time requirement.

The simulation results of applying the proposed multi-level graph partitioning algorithm to separate the 22,000-bus system into two islands are as follows,

Step 1. Only minimizing the real power flow on the islanding boundary

The number of edge cuts is 299 for two islands. The total real and reactive power flow on the islanding boundary of two islands are 4708 MW and 2913.5 MVar, respectively. The maximum reactive power flow on one edge of the boundary is 202.4 MVar, the real power flow on this edge is 26.2 MW.

Step 2. Only minimizing the reactive power flow on the islanding boundary

The number of edge cuts is 325 for two islands. The total real and reactive power flow on the islanding boundary of two islands are 6913.2 MW and 1434.7 MVar, respectively. The maximum reactive power flow on one edge of the boundary is 101.5 Mvar, the real power flow on this edge is 135.8 MW.

Step 3. Minimizing both the real and reactive power flow on the islanding boundary

The preference factor p is selected as 0.7. The number of edge cuts is 308 for two islands. The total real and reactive power flow on the islanding boundary of two islands are 5426.7 MW and 1734.7 MVar, respectively. The maximum reactive power flow on one edge of the boundary is 134.4 Mvar, the real power flow on this edge is 208.7 MW.

Fig. 13 and Fig. 14 compares the partitioning results on the 22,000 bus system obtained by the Step 1,2 and 3 partitioning described above. It is seen that the proposed multi-objective graph partitioning algorithm provides a good partitioning solution that minimizes both real and reactive power flows on the islanding boundary. The proposed algorithm is able to generate the partitioning that has a good tradeoff between the two objectives.

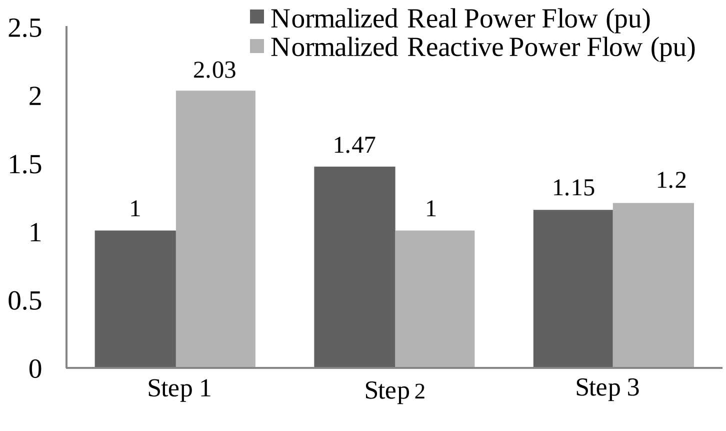


Fig. 13 Normalized Edge-Cut Results with Different Partitioning Objectives

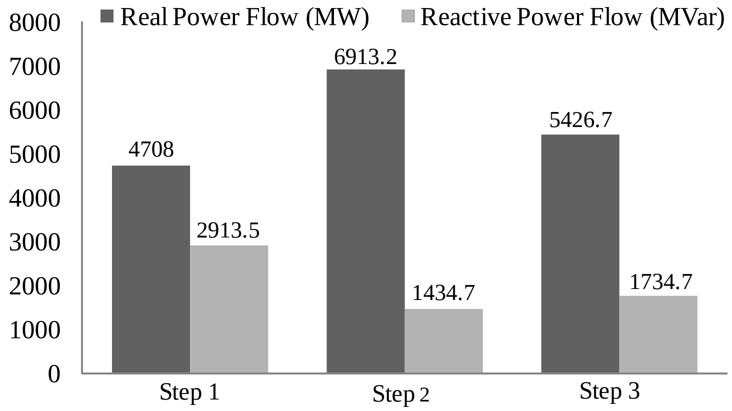


Fig. 14 Edge-Cut Results with Different Partitioning Objectives

6. Conclusion and Future Work

Controlled partitioning of the power network is an emergency control that involves partitioning of the power network into self-sufficient islands. This report presents an area partitioning algorithm that not only minimizes the imbalance of real power generation and load in each island, but also minimizes the imbalance of reactive power generation and loads in each island. More balanced reactive power supply and demand in each island could avoid the low voltage profile after system separation caused by tripping transmission lines that carry large reactive power. Time domain simulations are conducted on a 200-bus system to test the performance of the proposed algorithm. A simulation example shows that the proposed area partitioning strategy successfully prevents the post-islanding system collapse that occurs when the partitioning only considers real power balance. The proposed area partitioning strategy has the capability to enhance the grid's shock absorption capability with respect to cascading failures.

Simulations on a 22,000 bus system demonstrate that the proposed algorithm is computational efficient. It is feasible to implement a real-time power system reconfiguration scheme with high speed communication devices and intelligent control equipments.

The proposed area partitioning algorithm can be combined with a load shedding scheme to achieve a "smart grid" technology that enhances the robustness of the system and minimizes the impact of cascading events.

There are important issues that need to be addressed in the future:

- More detailed simulations including dynamic simulations should be conducted on very large systems, such as the 22,000 bus system to validate the proposed algorithm.
- Since the boundaries identified by the proposed area partitioning algorithm are not always identical with company boundaries, the partition needs to take into account practical considerations.
- More work is needed to determine the wide-area protection and control system needed for implementation of the proposed method.

References

[1] J. Giri, D. Sun, and R. Avila-Rosales, “Wanted: A More Intelligent Grid ,” *Power and Energy Magazine, IEEE*, vol 7, issue 2, pp. 34-40, Mar-Apr 2009.

[2] U.S.-Canada Power System Outage Task Force, “Final Report on the August 14th blackout in the United States and Canada,” United States Department of Energy and National Resources Canada, April 2004. [Online] available: https://reports.energy.gov/BlackoutFinal-Web.pdf

[3] UCTE, “Final Report of the Investigation Committee on the 28 September 2003 Blackout in Italy,” April 2004. [Online] available: http://www.ucte.org/_library/otherreports/20040427_UCTE_IC_Final_report.pdf

[4] UCTE, “Final Report on the Disturbances of 4 November 2006,” Jan. 2007. [Online] available: http://www.ucte.org/_library/otherreports/Final-Report-20070130.pdf

[5] Y. V. Makarov, V. I. Reshetov, V. A. Stroev, and N. I. Voropai, ”Blackout Prevention in the United States, Europe, and Russia,” *Proceedings of the IEEE*, vol. 93, No. 11, pp. 1942–1955, Nov. 2005.

[6] P. Pourbeik, P.S. Kundur, and C. W. Taylor, “The Anatomy of a Power Grid Blackout - Root Causes and Dynamics of Recent Major Blackouts,” *IEEE Power and Energy Magazine*, pp. 22-29, April 2006.

[7] I. Dobson, B.A. Carreras, V. Lynch, D.E. Newman, “Complex systems analysis of series of blackouts: cascading failure, criticality, and self-organization,” Bulk Power System Dynamics and Control VI Conf., Cortina, Italy, 2004.

[8] WSCC, “Western Systems Coordinating Council (WSCC) Disturbance Report for the Power System Outage that Occurred on the Western Interconnection, August 10, 1996,” approved by the WSCC Operations Committee on October 18, 1996.

[9] IEEE Power System Relaying Committee Working Group, “Performance of Generator Protection during Major System Disturbances,” *IEEE Trans. on Power Delivery*, vol. 19, No. 3, pp. 1650-1662, March 2004.

[10] IEEE PES Power Systems Relaying Committee Working Group D6, “Power Swing and Out-of-Step Considerations on Transmission Lines,” Tech. Rep. July 2005. [Online] available: http://www.pespsrc.org/Reports/Power Swing and OOS Considerations on Transmission Lines F.pdf

[11] K. Sun, D. Zheng, and Q. Lu, “Splitting Strategies for Islanding Operation of Large-scale Power Systems Using OBDD-Based Methods,” *IEEE Trans. Power Systems*, vol. 18, no. 2, pp. 912-923, May 2003.

[12] H. You, V. Vittal, and X. Wang, “Slow Coherency-Based Islanding,” *IEEE Trans. Power Systems*, vol. 19, no. 1, pp. 483–491, Feb. 2004.

[13] X.Wang and V.Vittal, “System Islanding Using Minimal Cutsets with Minimum Net Flow,” *Proceedings of the 2004 IEEE PES Power System Conference and Exposition*, New York, October, 2004.

[14] H. Li, G. Rosenwald, J. Jung, and C.C. Liu “Strategic Power Infrastructure Defense,” *Proceedings of the IEEE*, vol. 93, no. 5, pp. 918–933, May. 2005.

[15] B. Yang, “Slow Coherency Based Graph Theoretic Islanding Strategy,” Ph.D Dissertation, Arizona State University, 2007.

[16] B. Yang, V. Vittal and G. T. Heydt, “Slow Coherency Based Controlled Islanding – A Demonstration of the Approach on the August 14, 2003 Blackout Scenario,” *IEEE Trans. Power Systems*, vol. 21, No. 4, pp.1840-1847, Nov 2006.

[17] K. Schloegel, G. Karypis, and V. Kumar, “A New Algorithm for Multiobjective Graph Partitioning,” *Proceedings of the 1999 Euro-Par Conference*, Toulouse, France, 1999, pp. 322–331.

[18] G. Karypis and V. Kumar, “A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs,” SIAM Journal on Scientific Computing, vol.20, no. 1, pp. 359-392, 1998.

[19] J. Li and C.C. Liu, “Power System Reconfiguration Based on Multilevel Graph Partitioning,” *Proceedings of the 2009 PowerTech Conference*, Bucharest, Romania, June, 2009.

[20] P. Kundur, *Power System Stability and Control*, New York: McGraw-Hill, 1994.

[21] V. Venkatasubramanian and Y. Li, “Analysis of 1996 western American electric blackouts,” presented at the Bulk Power System Dynamics and Control VI Conf., Cortina, Italy, 2004.

[22] ANSI C50.13-1977 - ANSI National Standard Requirements for Cylindrical RotorSynchronous Generators, 1977.

Project Publications

[1] J. Li, C. C. Liu and K. Schneider, “Controlled Partitioning of a Power Network Considering Real and Reactive Power Balance,” Accepted for publication in *IEEE Trans. Smart Grids*, 2010.

[2] K. Yamashita, J. Li, P. Zhang and C. C. Liu, “Analysis and control of major blackout events,” *Proceedings of the 2009 IEEE PES Power System Conference and Exposition*, Seattle, March, 2009.

[3] J. Li and C.C. Liu, “Power System Reconfiguration Based on Multilevel Graph Partitioning,” *Proceedings of the 2009 PowerTech Conference*, Bucharest, Romania, June, 2009.

Part C

Methods to Estimate Propagation and the Distribution of Blackout Extent for Cascading Blackouts

Faculty:

Ian Dobson
University of Wisconsin-Madison

Ph.D. Student:

Janghoon Kim
University of Wisconsin-Madison

Information about Part C of this report

For information contact:

Ian Dobson
Professor
Department of Electrical and Computer Engineering
University of Wisconsin - Madison
1415 Engineering Drive
Madison, Wisconsin 53706
dobson@engr.wisc.edu
608-262-2661

Power Systems Engineering Research Center

The Power Systems Engineering Research Center (PSERC) is a multi-university Center conducting research on challenges facing the electric power industry and educating the next generation of power engineers. More information about PSERC can be found at the Center's website: http://www.pserc.org.

Power Systems Engineering Research Center
Arizona State University
577 Engineering Research Center
Tempe, Arizona 85287-5706
Phone: 480-965-1643
Fax: 480-965-0745

Notice Concerning Copyright Material

Contents

1 Introduction

This is the final report for the part of the 2007-2010 PSerc project on Fast simulation, monitoring and mitigation of cascading failure (S-32) performed at the University of Wisconsin-Madison. The main objective of this part of the project is to develop methods and tools to estimate propagation and the distribution of blackout extent in cascading outage data and simulations.

1.1 Main accomplishments

The main accomplishments of the UW-Madison part of the project are

1. We analyzed industry data for cascading transmission line outages recorded over ten years. The propagation of the outages increased as the cascades proceed. The number of line outages in a blackout is a measure of the blackout extent. We developed a practical method to predict the distribution of the total number of lines outaged from the initial line outages and the propagation. The new method uses data that must be reported to NERC and appears to be promising and practical. Access to testing data is limited, but the results on the available data are very good. One outcome could be new metrics for cascading failure.

2. We pursued a parallel effort to analyze simulation data to quantify the propagation of the cascades and predict the blackout size. This work not only gives a metric for cascade propagation to help interpret simulation results, but allows prediction of distribution of blackout size from much fewer simulation runs. In particular, we generalized the method from predicting distributions of lines outaged to predicting distributions of load shed and also developed the statistical analysis of these methods to demonstrate the order of magnitude improvements in using the approach to predict the distribution of the numbers of outages for the larger blackouts. The improved statistical analysis is not documented here, but is described in the journal papers [42, 24]. The generalization to quantifying propagation of load and predicting the distributions of load shed is summarized in section 3 and the journal paper [31] in the appendix.

3. We have completed work that quantifies how well a branching process model approximates a probabilistic model of cascading failure. We have obtained useful bounds on the ratio and difference of the probabilities from these two models. This work helps to justify the use

branching processes in the project to quantify cascading failure. This work is summarized in section and will appear in a journal paper [32] in 2010.

4. We put much effort into papers and talks to communicate our results and respond to questions from reviewers. A few of the papers were initiated with previous PSerc support and finished under the current project. Contributions, including being task leader, to the IEEE PES CAMS committee on cascading failure were made to help form the community of engineers working on cascading failure. This participation led to the conference papers [49, 50].

1.2 Students and publications

We summarize the student education supported and the papers and talks produced.

The project helped to support the following education at the University of Wisconsin-Madison:

- MS degree for Janghoon Kim. Thesis: Properties of the branching model and the cascading model of the failure propagation of the power network, MS Thesis, University of Wisconsin-Madison 2008.
- PhD degree for Janghoon Kim (ongoing)

The conference papers produced are:

- Initial review of methods for cascading failure analysis in electric power transmission systems IEEE PES CAMS Task Force on Cascading Failure, IEEE Power Engineering Society General Meeting, Pittsburgh PA USA, July 2008.
- IEEE PES CAMS Task Force on Cascading Failure, Vulnerability assessment for cascading failures in electric power systems, IEEE PES Power Systems Conference and Exposition, Seattle WA USA, March 2009.
- J. Kim, I. Dobson, Propagation of load shed in cascading line outages simulated by OPA, COMPENG 2010: Complexity in Engineering, Rome Italy, February 2010.

The journal papers that are published to date or will be published are the following:

- H. Ren, I. Dobson, B.A. Carreras, Long-term effect of the n-1 criterion on cascading line outages in an evolving power transmission grid, *IEEE Transactions on Power Systems*, vol. 23, no. 3, August 2008, pp. 1217-1225.
- H. Ren, I. Dobson, Using transmission line outage data to estimate cascading failure propagation in an electric power system, *IEEE Transactions on Circuits and Systems Part II*, vol. 55, no. 9, Sept. 2008, pp. 927-931.
- I. Dobson, J. Kim, K.R. Wierzbicki, Testing branching process estimators of cascading failure with data from a simulation of transmission line outages, *Risk Analysis*, vol. 30, no. 4, 2010, pp. 650-662.
- J. Kim, I. Dobson, Approximating a loading-dependent cascading failure model with a branching process, to appear in *IEEE Transactions on Reliability* (accepted March 2010).
- D.E. Newman, B.A. Carreras, V.E. Lynch, I. Dobson, Exploring complex systems aspects of blackout risk and mitigation, to appear in *IEEE Transactions on Reliability* in 2011.

The following presentations were made:

- Criticality, Self-organization and Cascading Failure in Electric Power System Blackouts, Princeton Plasma Physics National Lab, September 2007.
- Understanding Cascading failure, EPRI Workshop, Palo Alto CA December 2007.
- Cascading failure analysis, Lecture at EEI Transmission and Wholesale Markets School Madison, Wisconsin, August 2008.
- Can we quantify the risk of cascading failure blackouts with branching processes? Center for Nonlinear Studies seminar, Los Alamos National Lab, April 2009.
- How can complex system feedbacks shape cascading failure blackout risk towards criticality?, Center for Control, Dynamical Systems, and Computation seminar, University of California at Santa Barbara, April 2009.

- Modeling engineered and sustainable systems with complex system feedbacks, First international conference on computational sustainability CompSust09, Cornell University, Ithaca NY, June 2009.

- Extreme event research, California Energy Commission Transmission Research Program Colloquium, Costa Mesa CA, September 2009.

- Modeling cascading failure, Presentation at workshop: Vulnerability assessment of critical infrastructure, with case studies on power transmission networks and dams, Madison WI, January 2009

- Can we quantify the risk of cascading failure blackouts with branching processes? Center for Nonlinear Studies seminar, Los Alamos National Lab, April 2009.

- How can complex system feedbacks shape cascading failure blackout risk towards criticality?, Center for Control, Dynamical Systems, and Computation seminar, University of California at Santa Barbara, April 2009.

- Modeling engineered and sustainable systems with complex system feedbacks, First international conference on computational sustainability CompSust09, Cornell University, Ithaca NY, June 2009.

- Propagation of load shed in cascading line outages simulated by OPA, COMPENG 2010: Complexity in Engineering, Rome Italy, February 2010.

- Models of cascading failure in blackouts of electric power transmission systems, Electronic Power Grid Resilience Workshop, Naval Post Graduate School, Monterey CA, May 2010.

- Modeling cascading failure with branching processes, Workshop on Optimization and Control Theory for Smart Grids, Los Alamos national laboratory, Los Alamos NM, August 2010.

1.3 Budget and challenges

The UW-Madison share of the budget for S-32 was a total of $56K planned to be spread over 2 years. The S-32 funding was intended to be supplemental funding for a larger project, but the base funding of the larger project came through but was unexpectedly and unpredictably delayed for years. This was partially addressed by a no-cost extension extending the time of performance of the project.

There were many difficulties in obtaining reliability data for the project. Many requests came to naught or encountered legal difficulties. There was finally success in obtaining the line trip data analyzed in section 2. The delays were partially addressed by a no-cost extension extending the time of performance of the project. The lack of real data was also addressed in part by working further on simulated data.

There were protracted negotiations with the reviewers of some of the journal papers. All the papers were eventually published in good journals, but the process was time consuming.

2 Observed line outage data and testing new branching models

2.1 Summary

We analyze cascading transmission line outages recorded over ten years in a North American utility, based on data that must be reported to NERC. We obtain the empirical distributions of the number of lines outaged for the initial line outages and for the total line outages after cascading. The propagation of the outages increases as the cascades proceed. We test a new method based on branching processes for predicting the distribution of the total number of lines outaged from the initial line outages and the increasing propagation. The new method is validated in the sense of being consistent with the data and appears to be practical.

2.2 Outage data

The transmission line outage data is 8864 outages recorded by a North American utility over a period of ten years. This is standard data that is now required to be reported by all utilities to NERC under the TADS Transmission Availability Data System. The data for each transmission line outage includes the trip time (to the nearest minute) as well as other data. All the line outages are automatic trips. More than 99% of the outages are of lines rated 69 kV or above and more than 96% of the outages are of lines rated 115 kV or above. There are several types of line outages in the data and a variety of reasons for the trips. In processing the data, both voltage levels and all types of line outages are regarded as the same and the reasons for the line outages are neglected. For this bulk statistical analysis, neglecting these distinctions is a useful first step as we proceed.

2.3 Grouping outages into cascades and stages

For our analysis it is necessary to group the line outages first into different cascades, and then into different stages within each cascade. Here we use a simple method based on outages' timing [42, 17]. Since operator actions are usually completed within one hour, we assume that successive outages separated in time by more than one hour belong to different cascades. Since fast transients or auto-recloser actions are completed within one minute, we assume that successive outages in a given cascade separated in time by more than one minute are in different stages within that cascade. One result of

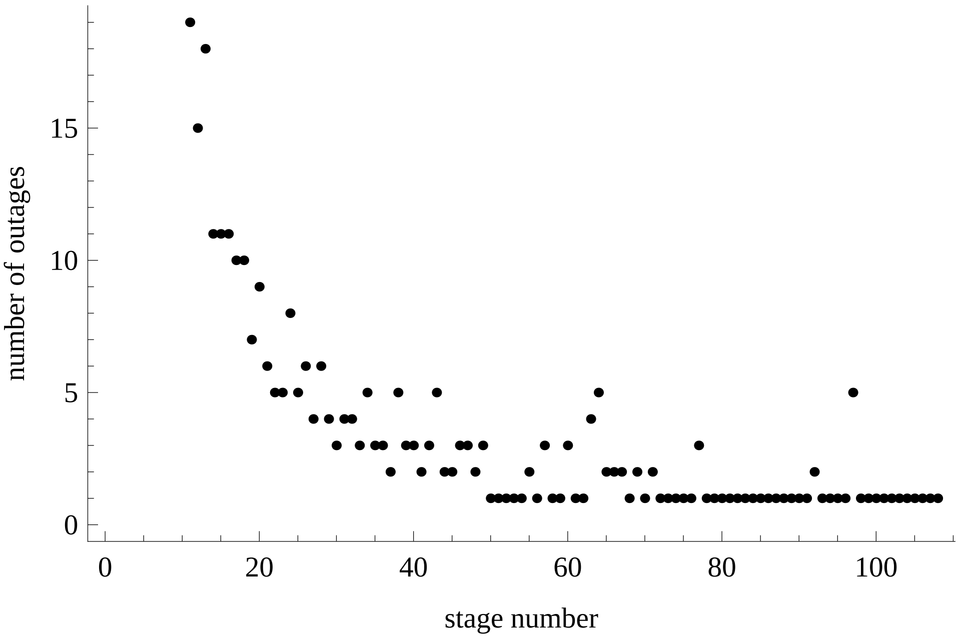


Figure 1: Sum of line outages in each of stages 11 to 109 of all cascades.

the grouping of the outages into cascades and stages is that there are 5227 cascades and the longest cascade has 110 stages.

Table 1 is obtained by summing over all the 5227 cascades the number of outages in each of stages 0 to 10. That is, of the 8864 outages, 6254 are in stage 0 of a cascade, 1143 are in stage 1 of a cascade, and so on. The number of outages in each of stages 11 to 109 is shown in Fig. 1.

The probability distribution of the number of initial outages is shown by the circles in Fig. 2. The probability distribution of the total number of outages is shown by the squares in Fig. 2. Both distributions seem to have a power law character. However, there are too few line outages at the highest number of line outages to accurately estimate the higher ends of these distributions. The exponent of the power law distribution of the total number of failures is roughly −2.7.

Table 1: Number of outages in initial stages summed over the cascades

stage number	0	1	2	3	4	5	6	7	8	9	10
number of outages	6254	1143	434	227	155	95	78	53	46	32	31

2.4 Propagation in the cascades

In our branching process model of cascading, each outage in each stage (a "parent" outage) independently produces a random number 0,1,2,3,... of outages ("child" outages) in the next stage according to an offspring distribution that is a Poisson distribution of mean λ. The child outages then become parents to produce the next generation and so on. If the number of outages in a stage becomes zero, the cascade stops. The mean number of child outages for each parent (the average family size) is the parameter λ. λ quantifies the average tendency for the cascade to propagate.

There are two main ways to estimate the propagation. The first way is to count all the outages in the cascade that are children and divide this by all the outages that are parents. This gives the propagation λ averaged over the number of stages [42]. We report on this calculation at the end of this subsection.

The second way is to look at how many children are produced by each parent at each stage. This gives an estimate λ_k for each stage $k = 1, 2, 3, ...$ which is computed by dividing the number of outages in stage k by the number of outages in stage $k-1$. For example, stage 0 has 6254 outages and these parents produced 1143 child outages in stage 1. Therefore the average number of children in stage 1 per parent in stage 0 is $\lambda_1 = 1143/6254 = 0.18$. Stage 1 has 1143 outages and these outages considered as parents produced 434 child outages in stage 2. Therefore the average number of children in stage 2 per parent in stage 1 is $\lambda_2 = 434/1143 = 0.38$. The results of computing λ_k for stages $k = 1, 2, 3, ..., 19$ are shown in Fig. 3 and Table 2. As the cascade progresses, λ_k increases from 0.18 and appears to level off at approximately 0.75. The higher stages have too few outages to accurately estimate λ_k and the results for higher stages become noisy.

Table 2: Estimated stage propagations λ_k

k	1	2	3	4	5	6	7	8	9	10	11	12
λ_k	0.18	0.38	0.52	0.68	0.61	0.82	0.68	0.87	0.70	0.97	0.61	0.79

If we compute the propagation λ averaged over the number of stages using the method of [42] by dividing the total number of children in all the cascades by the total number of parents in all the cascades, we get $\lambda = 0.29$. This value averaged over the stages is dominated by the stages 0 and 1. If we omit stage 0 and recompute λ averaged over the remaining stages, we get $\lambda = 0.56$. If we omit stage 0 and stage 1 and recompute λ averaged over the remaining stages, we get $\lambda = 0.70$. It seems unsatisfactory to be

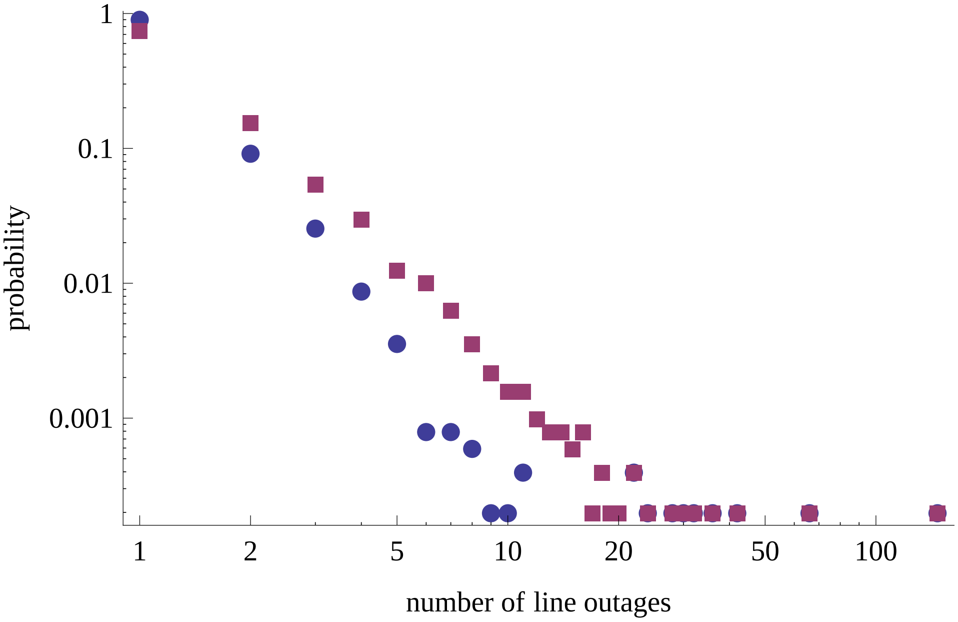


Figure 2: Probability distribution of initial (circles) and and total (squares) line outages.

using methods that essentially assume λ to be roughly constant when it is increasing significantly, so in this note we use a new method that accounts for the increase.

2.5 Predicting total outage distribution with branching process

We predict the distribution of the total number of outages using a branching process model from the distribution of initial outages and the propagation. The new aspect is that we account for the change in propagation as the cascade proceeds. In particular we assume that the stage propagation is given by Table 3, which is obtained from the estimated propagations for the first 4 stages in Table 3, followed by an assumption of $\lambda_k = 0.75$ for $k \geq 5$. $\lambda_k = 0.75$ is a guesstimate of the asymptotic propagation based on the noisy data in Fig. 3. The branching process is assumed to have the initial distribution of outages given by the data as shown by the circles in Fig. 2 and propagation at each stage with a Poisson distribution with mean given by the stage propagations in Table 3. The general reasons for

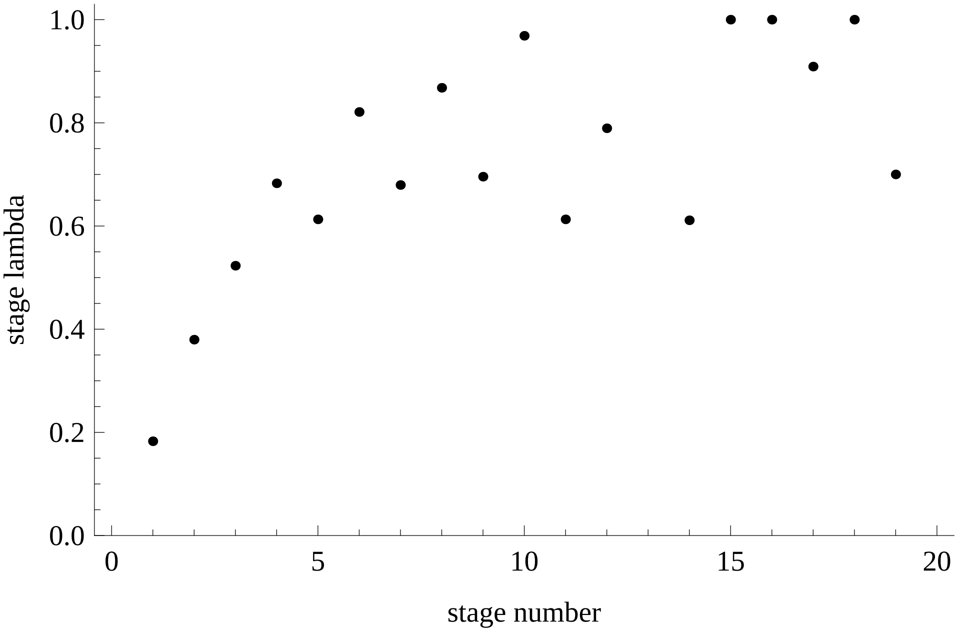


Figure 3: λ_k estimated from the outage data at stages $k = 1, 2, ..., 19$.

Table 3: Stage propagations λ_k for predicting total outage distribution

k	1	2	3	4	≥ 5
λ_k	0.18	0.38	0.52	0.68	0.75

assuming a Poisson distribution are explained in [42]. The details of this new computation are quarantined in the appendix. The predicted distribution of total number of outages is shown by the line in Figure 4 and it can be seen that the match with the empirical distribution of total number of outages is very good.

2.6 Summary of calculations

This section assumes some familiarity with branching processes [27]. Consider a single line outage that occurs in stage k and let the total number of outages that are descendants of this outage in any subsequent stage (children plus grandchildren plus great grandchildren and so on) be Y_k. Let the generating function of Y_k be $F_k(s) = Es^{Y_k}$. The number of descendants of the single line outage plus the single line outage itself is $Y_k + 1$ and $Y_k + 1$ has generating function $sF_k(s)$. Let the generating function of the offspring

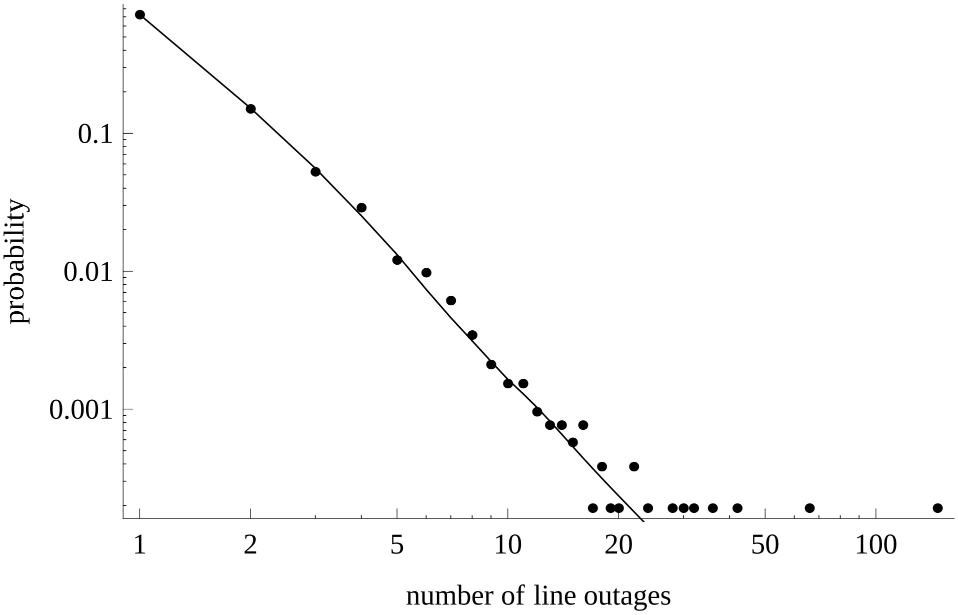


Figure 4: Distribution of total number of outages from data (dots) and estimated using branching process (line).

distribution producing stage k from stage $k-1$ be $f(s)$. Then the basic recursion for computing all the descendants of an outage at a given stage is

$$F_{k-1}(s) = f(sF_k(s)) \tag{1}$$

We now apply this recursion to compute the total number of outages given the stage propagations λ_k in Table 3. Since $\lambda_k = 0.75$ for $k \geq 5$, the total number of outages that are descendants of an outage in stage 4 plus the outage itself is given by a Borel-Tanner distribution with parameter 0.75. We write $f_B(s, 0.75)$ for the generating function of the Borel-Tanner distribution with parameter 0.75. We write $f(s, \lambda_k) = \exp[\lambda_k(s-1)]$ for the generating function of the Poisson offspring distribution with mean λ_k. We write $f_0(s)$ for the generating function of the initial distribution of failures. $f_0(s)$ is computed from the empirical initial distribution of outages. We write $F(s)$ for the generating function of the total number of outages that we wish to compute. Then applying the recursion (1) successively, we get

$$\begin{aligned}
sF_4(s) &= f_B(s, 0.75)\\
F_3(s) &= f(sF_5(s), 0.68) = f(f_B(s, 0.75), 0.68)\\
F_2(s) &= f(sf(f_B(s, 0.75), 0.68), 0.52)\\
F_1(s) &= f(sf(sf(f_B(s, 0.75), 0.68), 0.52), 0.38)\\
F_0(s) &= f(sf(sf(sf(f_B(s, 0.75), 0.68), 0.52), 0.38), 0.18)\\
F(s) &= f_0(sf(sf(sf(sf(f_B(s, 0.75), 0.68), 0.52), 0.38), 0.18))
\end{aligned} \tag{2}$$

Equation (2) shows that $F(s)$ is a complicated polynomial, but it can be evaluated by Mathematica for as many terms as needed. For example, computing 500 terms of $F(s)$ predicts the total number of outages as shown in Fig. 5. Note the power law character of the distribution up to about 100 outages.

2.7 Conclusions

For this utility data set we conclude that:

- Propagation of line outages increases as the cascade progresses and then appears to level out.
- The distributions of the initial and total number of line outages have an approximate power law character over their initial portions that correspond to smaller number of outages.

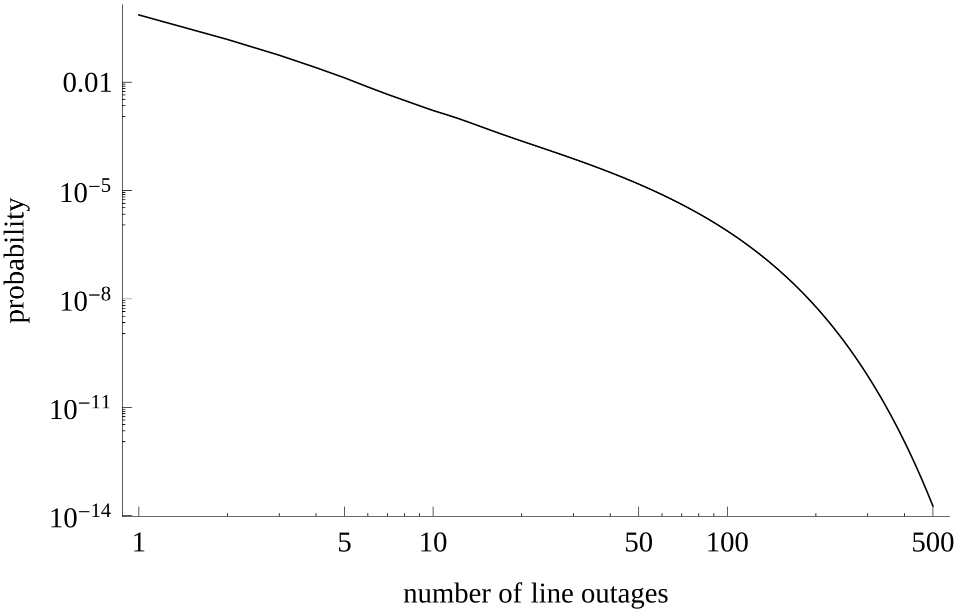


Figure 5: Distribution of total number of outages predicted using branching process up to 500 outages.

- The distribution of the total number of outages predicted with the branching process matches well the empirical distribution of the total number of outages. This validates the branching process model for predicting the distribution of the total number of outages in the sense that it is consistent with this data set.

- A branching process that accounts for the varying propagation as the cascade progresses can give a good prediction of the distribution of the total number of line outages from the distribution of the initial number of line outages. Conventional risk analysis or a contingency list can give the distribution of the initial number of line outages. The varying propagation can be estimated from recorded line trips as demonstrated in this section. This is a new method to predict the effect of cascading on known or assumed initial line trips. The new method seems to be practical and the computations are easy to implement with computer algebra. It is likely that much less than ten years of data is necessary for an accurate prediction of the propagation, but this has not yet been analyzed.

- The method uses standard TADS data that is required to be reported to NERC.

This is the second line outage data set analyzed for propagation after the initial analysis of a smaller data set in [42]. This analysis accounts for varying propagation as the cascade progresses whereas [42] does not. In both this section and [42], the outages are simply grouped into stages and cascades according to their timing.

3 Estimating the distribution of load shed

This section briefly summarizes a new method of estimating the probability distribution of load shed from the initial load shed and an estimate of the average amount of propagation of load shed. The details are given in the conference paper [31] that is reprinted in the appendix. The average propagation of the simulated load-shed data is estimated and then the initial load shed is discretized and propagated with a Galton-Watson branching process model of cascading failure to estimate the probability distribution of total load shed. The distribution of total load shed, which gives the frequencies of small, medium and large blackouts, is a basic metric of blackout frequency since the load shed directly affects customers and society. The main difference from previous work [24] that similarly estimated the probability distribution of the total number of lines outaged is that load shed is a continuously varying quantity whereas the number of lines outaged is an integer. Hence the need to devise a way to discretize the load-shed data before applying the Galton-Watson branching process model.

This estimated distribution of total load shed was initially tested using load-shed data generated by the OPA simulation of cascading transmission line outages on the IEEE standard 300-bus test system. A key advantage of this method is that it requires much less data (fewer simulated cascades) to estimate the probability distribution of total load shed, and especially to estimate the total load shed of the larger blackouts. (Direct simulation of these rare events is very time consuming). This is not only helpful in reducing simulation times, which are always burdensome and often prohibitive for cascading-failure simulations of large power-system models, but also will be crucial in designing practical methods of estimating the probability distribution of load shed from cascades observed in the power system. The testing showed that the estimated distribution is close to the empirical distribution in most of the cases tested, suggesting that the branching process model with an averaged propagation can capture some aspects of the cascading of load shed, at least for the purpose of estimating the probability distribution of total load shed. These first results are sufficiently promising that further testing with other cascading failure simulations or on larger grid models is warranted.

4 Approximating a high-level cascading failure model with a branching process

4.1 Introduction

This section motivates and illustrates the detailed analysis that will soon appear in the 2010 journal paper [32]. The objective of [32] is to quantify the closeness of the approximation between two high-level probabilistic models of cascading failure. In one model called CASCADE, failing components successively load the unfailed components, whereas the other model is based on a Galton-Watson branching process. The CASCADE model more directly summarizes a successive loading mechanism of cascading failure. Both models are generic, idealized models of cascading failure of a large, but finite number of components. For suitable parameters, the distributions of the total number of failures from the branching process and CASCADE models are close enough to make the branching process a useful approximation. It is advantageous to use the simpler branching process model when it is a good approximation. Moreover, there is a substantial and useful literature on applying branching processes to other cascading processes [27, 29, 25].

This section summarizes the CASCADE and branching process models and illustrates the closeness of the distributions of the total number of failures. All the analytic work giving quantitive bounds on the closeness is quarantined in [32].

4.2 Summary of models

The CASCADE model is an analytically tractable probabilistic model of cascading failure that captures the weakening of the system as the cascade proceeds [19]. There are a large but finite number n of identical components and each component has a level of loading or stress. The initial load on each component is an independent uniform random variable over a fixed range of loading. There is an initial disturbance to the system that adds additional loading to each component. Each component has a maximum loading threshold and fails if this threshold is exceeded. When any component fails, all the other components are additionally loaded so that initial failures can lead to a cascading sequence of failures as components successively overload and additionally load the other components. The cascade continues until there are no further failures or all the components are failed. The main parameters are the size d of the initial disturbance and the amount p by which load of other components is incremented when a component fails, which controls the extent to which the cascade propagates.

The branching process model of cascading failure is a standard Galton-Watson branching process [27] with Poisson offspring distributions, except that there are a finite number n of components. The failures are produced in generations. In generation zero, there is an initial Poisson distribution of failures with mean θ that represents the initial disturbance to the system. Each failure in each generation produces further failures according to a Poisson offspring distribution with mean λ until no more failures are produced or all the components fail. The main parameters are the mean size θ of the initial disturbance and the mean number of offspring failures λ which controls the extent to which the cascade propagates.

The parameters of the CASCADE and branching process models correspond according to

$$\theta = nd \tag{3}$$

$$\lambda = np \tag{4}$$

4.3 Showing the qualitative agreement between the models

Both the CASCADE and the branching process model have analytic formulas for the probability distributions of the total number of failures in terms of the model parameters. The objective of [32].is to quantify how well these formulas agree. In particular, [32]. gives explicit bounds for the closeness of the probability distributions of the total number of failures for the CASCADE and branching models.

Fig. 6 shows examples of the CASCADE probability distribution as p increases in the case of $n = 5000$ components and small initial disturbance $d = 0.0002$. The distribution for $p = 0.0001$ and $np = 0.5$ has an exponential tail slightly heavier than binomial. Hence there is an extremely small probability of cascades in which a large fraction of the components fail. The tail becomes heavier as p increases and the distribution for $p = 0.0002$ and $np = 1$ has an approximate power law region over a range of r. This implies a non negligible probability of cascades that extend to the system size, and, in this case, the probability of all 5000 components failing is 0.00054. The distribution for $p = 0.0003$ and $np = 1.5$ has an approximately exponential tail for small r, zero probability of intermediate r, and a probability of 0.44 of all 5000 components failing. (If an intermediate number of components fail, then the cascade always proceeds to all 5000 components failing.)

Fig. 7 shows the probability distribution of the total number of failures for $p = 0.0001$ and $np = 0.5$, but with a larger initial disturbance $d = 0.0002$ that gives a mean initial disturbance of $nd = 10$ components failed.

Comparing Figs. 6 and 8 illustrates the qualitative agreement between the CASCADE and branching process models. The model parameters chosen in Figs. 6 and 8 correspond according to (3) and (4). Plotting the branching process probability distribution corresponding to the CASCADE distribution in Fig. 7 yields a figure indistinguishable from Fig. 7.

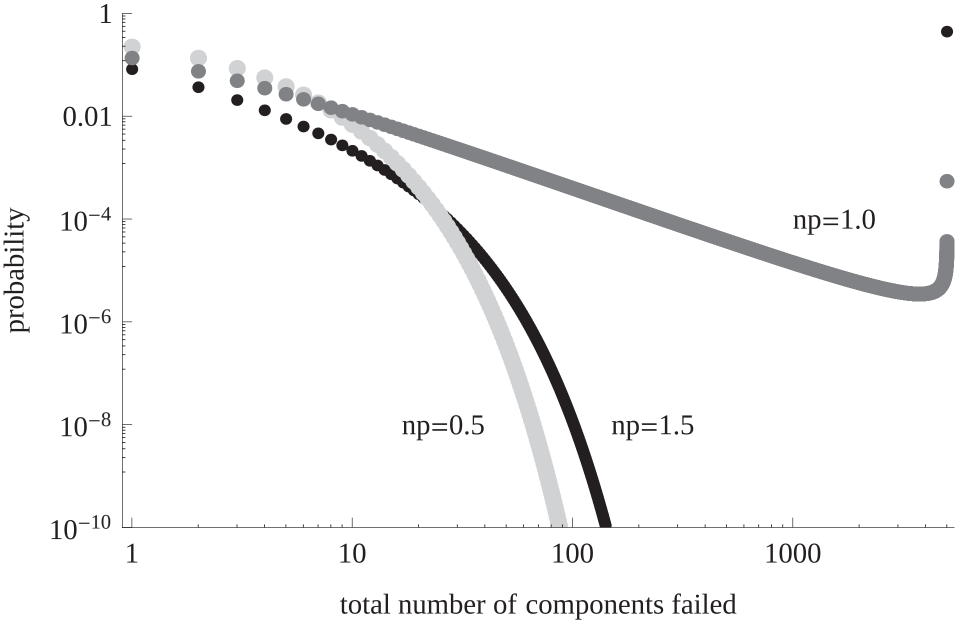


Figure 6: Probability distribution of total number of failures from CASCADE model with $n = 5000$ components, initial disturbance d with $nd = 1$ and each failure causing load increment p with $np = 0.5$ (light gray dots), $np = 1.0$ (dark gray dots), and $np = 1.5$ (black dots). The probability of 5000 failures is negligible for $np = 0.5$, 0.00054 for $np = 1$, and 0.44 for $np = 1.5$. The probability of zero failures is 0.3678 in all cases.

4.4 Examples of results of analysis

We now give some examples of typical results from [32].

Example 1.

In our motivating application of cascading failure blackouts in power transmission networks, estimates for the costs vary widely. For example, es-

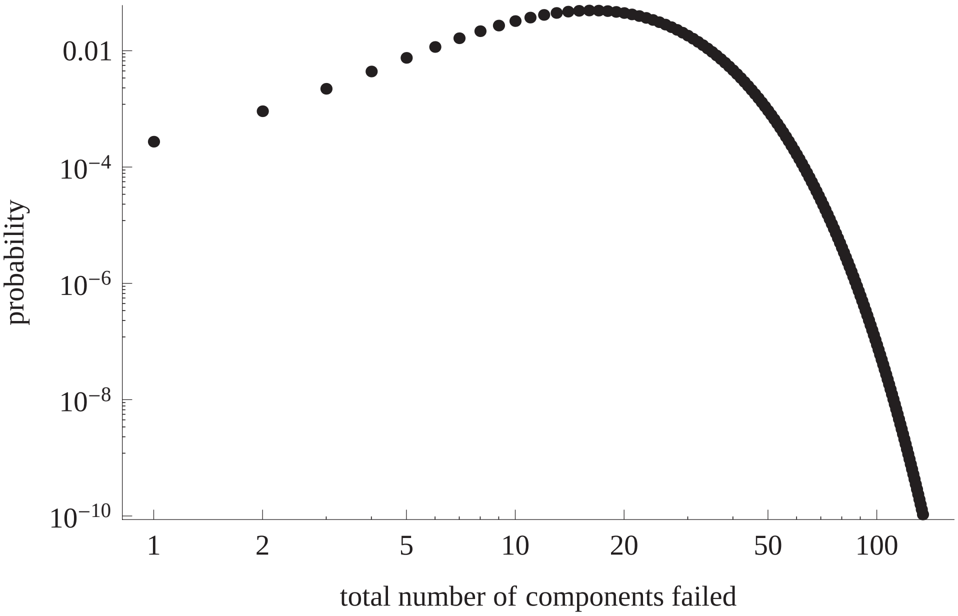


Figure 7: Probability distribution of total number of failures from CASCADE model with $n = 5000$ components, initial disturbance d with $nd = 10$ and each failure causing load increment p with $np = 0.5$. The probability of 5000 failures is negligible. The probability of zero failures is 0.000045.

timates for direct costs of the August 2003 blackout of Northeastern America vary from about 4 to 12 billion dollars. And indirect costs, such as when there is rioting or damage to other infrastructures, can readily double or triple the costs, but are uncertain and hard to quantify. Suppose that risk is computed as probability of blackout times cost. Then there is little use for estimates of blackout probability that are significantly more accurate than the costs. For the sake of illustration, we measure the cascading blackout size by the number of failures and require blackout probabilities to be accurate within a factor of 2. That is, if we write R for the ratio of the probability of the branching process having r failure and the probability of the CASCADE model having r failures, then we require our branching process approximation to have ratio R satisfy $\frac{1}{2} < R < 2$. A typical result from the analysis in [32] is that a condition guaranteeing $\frac{1}{2} < R < 2$ is that

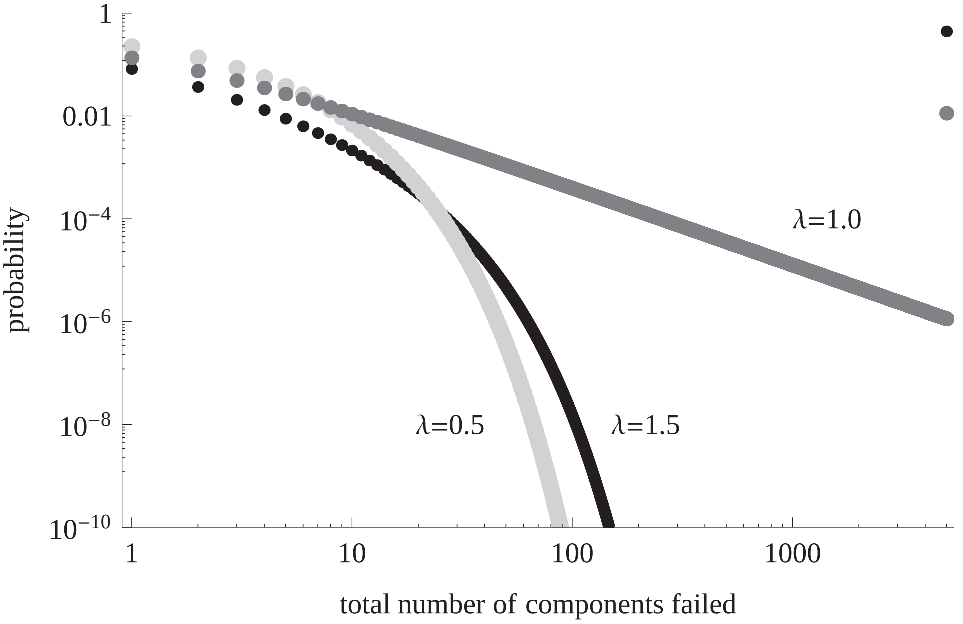


Figure 8: Probability distribution of total number of failures from branching process model with $n = 5000$ components, initial disturbance $\theta = 1$ and offspring mean $\lambda = 0.5$ (light gray dots), $\lambda = 1.0$ (dark gray dots), and $\lambda = 1.5$ (black dots). The probability of 5000 failures is negligible for $\lambda = 0.5$, 0.011 for $\lambda = 1$, and 0.44 for $\lambda = 1.5$. The probability of zero failures is 0.3679 in all cases.

$0 < \lambda < 1$, $\theta \leq \frac{n}{2}$ and

$$\frac{\theta}{1-\lambda} < r < \min\left\{\frac{0.83\sqrt{n}+\theta}{1-\lambda}, \frac{n}{2}\right\}$$

The range over which the approximation is valid increases with n.

Example 2. Practical industry models for power transmission networks typically range from hundreds to tens of thousands of nodes. We choose $n = 1000$ nodes, a small initial disturbance $\theta = 1$ and $\lambda = 0.5$. Then a typical result from the analysis in [32] is that $\frac{1}{2} < R < 2$ for $0 \leq r \leq 73$. This bound is fairly tight: direct calculation shows that in this case, the maximum range of r over which $\frac{1}{2} < R < 2$ is $0 \leq r \leq 76$.

To show the effect of increasing n, redoing Example 2 with $n = 10000$ nodes yields $\frac{1}{2} < R < 2$ for $0 \leq r \leq 234$.

Example 3. We choose $n = 1000$, $\theta = 1$ and $\lambda = 0.98$. Then a typical result from the analysis in [32] is that $\frac{1}{2} < R < 2$ for $0 \leq r \leq 750$.

Redoing Example 3 with $n = 10000$ nodes yields $\frac{1}{2} < R < 2$ for $0 \leq r \leq 4439$.

4.5 Conclusion

High-level probabilistic models of cascading failure such as the CASCADE model are emerging as one of the useful approaches in the study of large blackouts. In this section we approximate CASCADE with a Galton-Watson branching process and motivate and illustrate the closeness of the approximation for the probability distribution of the total number of failures that is analyzed in detail in [32]. Since the branching process is a simple and well understood probabilistic model, it is advantageous to use it when it is a good approximation. The analysis accounts for the large but finite number of components needed in the study of large blackouts.

References

[1] R. Adler, S. Daniel, C. Heising, M. Lauby, R. Ludorf, T. White, An IEEE survey of US and Canadian overhead transmission outages at 230 kV and above, IEEE Transactions Power Delivery, vol. 9, no. 1, Jan. 1994, pp. 21-39.

[2] M. Anghel, K. A. Werley, A. E. Motter, Stochastic Model for Power Grid Dynamics, 40th Hawaii International Conference on System Sciences, Hawaii, January 2007.

[3] K.B. Athreya, P.E. Ney, *Branching Processes*, Dover NY 2004 (reprint of Springer-verlag Berlin 1972).

[4] B.A. Carreras, D. E. Newman, I. Dobson, A. B. Poole, Initial evidence for self-organized criticality in electric power blackouts, 33rd Hawaii International Conference on System Sciences, Maui, Hawaii, January 2000.

[5] B.A. Carreras, V.E. Lynch, I. Dobson, D.E. Newman, Critical points and transitions in an electric power transmission model for cascading failure blackouts, Chaos, vol. 12, no. 4, December 2002, pp. 985-994.

[6] B.A. Carreras, V.E. Lynch, D.E. Newman, I. Dobson, Blackout mitigation assessment in power transmission systems, 36th Hawaii International Conference on System Sciences, Hawaii, 2003.

[7] B.A. Carreras, V.E. Lynch, I. Dobson, D.E. Newman, Complex dynamics of blackouts in power transmission systems, Chaos, vol. 14, no. 3, September 2004, pp. 643-652.

[8] B.A. Carreras, D.E. Newman, I. Dobson, A.B. Poole, Evidence for self organized criticality in a time series of electric power system blackouts, IEEE Transactions on Circuits and Systems I, vol. 51, no. 9, September 2004, pp. 1733-1740.

[9] J. Chen, J.S. Thorp, I. Dobson, Cascading dynamics and mitigation assessment in power system disturbances via a hidden failure model, International Journal of Electrical Power and Energy Systems, vol. 27, no. 4, May 2005, pp. 318-326.

[10] Q. Chen, J.D. McCalley, Identifying high risk n-k contingencies for online security assessment, IEEE Transactions on Power Systems, vol. 20, no. 2, May 2005, pp. 823-834.

[11] Q. Chen, C. Jiang, W. Qiu, J.D. McCalley, Probability models for estimating the probabilities of cascading outages in high-voltage transmission network, IEEE Trans. Power Systems, vol. 21, no. 3, August 2006, pp. 1423-1431.

[12] Defense plans against extreme contingencies, CIGRE Task Force C2.02.24 report, and a paper summarizing the report in Electra no. 231, April 2007.

[13] P.C. Consul, M.M. Shoukri, Some chance mechanisms related to a generalized Poisson probability model, American Journal of Mathematical and Management Sciences, vol. 8, nos. 1 and 2, pp. 181-202, 1988.

[14] P.C. Consul, *Generalized Poisson Distributions*, Dekker, NY 1989.

[15] J.-P. Dion, N. Keiding, Statistical inference in branching processes, in *Branching Processes*, editors A. Joffe, P. Ney, Marcel Dekker, New York 1978, pp. 105-140.

[16] I. Dobson, B.A. Carreras, D.E. Newman, A branching process approximation to cascading load-dependent system failure, 37th Hawaii International Conference on System Sciences, Hawaii, 2004.

[17] I. Dobson, B.A. Carreras, D.E. Newman, Branching process models for the exponentially increasing portions of cascading failure blackouts, 38th Hawaii International Conference on System Sciences, January 2005, Hawaii.

[18] I. Dobson, K.R. Wierzbicki, B.A. Carreras, V.E. Lynch, D.E. Newman, An estimator of propagation of cascading failure, 39th Hawaii International Conference on System Sciences, January 2006, Kauai, Hawaii.

[19] I. Dobson, B.A. Carreras, D.E. Newman, A loading-dependent model of probabilistic cascading failure, Probability in the Engineering and Informational Sciences, vol. 19, no. 1, 2005.

[20] I. Dobson, B.A. Carreras, V.E. Lynch, B. Nkei, D.E. Newman, Estimating failure propagation in models of cascading blackouts, Probability in the Engineering and Informational Sciences, vol. 19, no. 4, October 2005, pp 475-488.

[21] I. Dobson, B.A. Carreras, V.E. Lynch, D.E. Newman, Complex systems analysis of series of blackouts: cascading failure, critical points, and self-organization, Chaos, vol. 17, no. 2, June 2007.

[22] I. Dobson, Where is the edge for cascading failure?: challenges and opportunities for quantifying blackout risk, IEEE Power Engineering Society General Meeting, Tampa FL USA, June 2007

[23] I. Dobson, K.R. Wierzbicki, J. Kim, H. Ren, Towards quantifying cascading blackout risk, Bulk Power System Dynamics and Control-VII, Charleston SC USA, August 2007.

[24] I. Dobson, J. Kim, K.R. Wierzbicki, Testing branching process estimators of cascading failure with data from a simulation of transmission line outages, *Risk Analysis*, vol. 30, no. 4, 2010, pp. 650 - 662.

[25] P. Guttorp, *Statistical inference for branching processes*, Wiley, NY, 1991

[26] R.C. Hardiman, M.T. Kumbale, Y.V. Makarov, An advanced tool for analyzing multiple cascading failures, Eighth International Conference on Probability Methods Applied to Power Systems, Ames Iowa, September 2004.

[27] T.E. Harris, *Theory of branching processes*, Dover NY 1989.

[28] Blackout Experience and Lessons, Best Practices for System Dynamic Performance, and the Role of New Technologies, IEEE Special Publication 07TP190, prepared by the IEEE PES Task Force on Blackout Experience, Mitigation, and Role of New Technologies of the Power System Dynamic Performance Committee of the IEEE Power Engineering Society, July 2007.

[29] P. Jagers, *Branching processes with biological applications*, Wiley London,New York, 1975.

[30] P.J.M. Kallenburg, *Branching processes with continuous state space*, Mathematical Centre Tracts 117, ISBN 90 6196 188 2, Amsterdam 1979.

[31] J. Kim, I. Dobson, Propagation of load shed in cascading line outages simulated by OPA, COMPENG 2010: Complexity in Engineering, Rome Italy, February 2010.

[32] J. Kim, I. Dobson, Approximating a loading-dependent cascading failure model with a branching process, to appear in *IEEE Transactions on Reliability* (accepted March 2010).

[33] D.S. Kirschen, D. Jawayeera, D.P. Nedic, R.N. Allan, A probabilistic indicator of system stress, IEEE Transactions on Power Systems, vol. 19, no. 3, 2004, pp. 1650-1657.

[34] D.S. Kirschen, Do investments prevent blackouts?, IEEE Power Engineering Society General Meeting, Tampa FL USA, June 2007.

[35] H. Liao, J. Apt, S. Talukdar, Phase transitions in the probability of cascading failures, Electricity Transmission in Deregulated Markets, conference at Carnegie Mellon University, Pittsburgh PA USA Dec. 2004.

[36] S. Mei, Yadana, X. Weng, A. Xue, Blackout model based on OPF and its self-organized criticality, Proceedings of the 25th Chinese Control Conference, Harbin, Heilongjiang, China, August 2006.

[37] L. Mili, Q. Qui, A.G. Phadke, Risk assessment of catastrophic failures in electric power systems, International Journal of Critical Infrastructures, vol. 1, no. 1, pp.3863, 2004.

[38] D.P. Nedic, D.S. Kirschen, Discovering mechanisms of disturbance development, IREP Conference on Bulk Power System Dynamics and Control VI, Cortina D'Ampezzo, Italy, August 2004.

[39] D.P. Nedic, I. Dobson, D.S. Kirschen, B.A. Carreras, V.E. Lynch, Criticality in a cascading failure blackout model, International Journal of Electrical Power and Energy Systems, vol. 28, 2006, pp. 627-633. (journal publication of conference paper in Fifteenth Power Systems Computation Conference, Liege Belgium, August 2005)

[40] D.E. Newman, B.A. Carreras, V.E. Lynch, I. Dobson, The impact of various upgrade strategies on the long-term dynamics and robustness of the transmission grid, Electricity Transmission in Deregulated Markets, conference at Carnegie-Mellon University, Pittsburgh PA USA, December 2004.

[41] T. Nippert, Improvement of the (n-1) criterion introducing a probabilistic failure-related reliability criterion, CIRED97, 14th International Conference and Exhibition on Electricity Distribution, vol. 6, IEE Conf. Publ. No. 438, Birmingham, UK, 1997, pp. 37/1-37/6.

[42] H. Ren, I. Dobson, Using transmission line outage data to estimate cascading failure propagation in an electric power system, to appear in IEEE Transactions on Circuits and Systems Part II, in 2008.

[43] H. Ren, I. Dobson, B.A. Carreras, Long-term effect of the n-1 criterion on cascading line outages in an evolving power transmission grid, to appear in IEEE Transactions on Power Systems, accepted April 2008.

[44] N.D. Reppen, Increasing utilization of the transmission grid requires new reliability criteria and comprehensive reliability assessment, Eighth International Conference on Probabilistic Methods Applied to Power Systems, Ames, Iowa USA September 2004.

[45] M.A. Rios, D.S. Kirschen, D. Jayaweera, D.P. Nedic, R.N. Allan, Value of security: modeling time-dependent phenomena and weather conditions. IEEE Transactions on Power Systems, vol. 17, no. 3, pp. 543-8, 2002.

[46] E. Seneta, D. Vere-Jones, On the asymptotic behaviour of subcritical branching processes with continuous state space, Zeitschrift fur Wahrscheinlichkeitstheorie und verwandte Gebiete, vol. 10, pp. 212-225, 1968.

[47] B. Stott, E. Hobson, Power system security control calculations using linear programming, Part I and Part II, IEEE Transactions on Power Apparatus and Systems, vol. PAS-97, no. 5, Sept/Oct 1978, pp. 1713-1731.

[48] U.S.-Canada Power System Outage Task Force, Final Report on the August 14th blackout in the United States and Canada. United States Department of Energy and National Resources Canada, April 2004.

[49] IEEE PES CAMS Task Force on Cascading Failure, Initial review of methods for cascading failure analysis in electric power transmission systems, IEEE Power and Energy Society General Meeting, Pittsburgh PA USA, July 2008

[50] IEEE PES CAMS Task Force on Cascading Failure, Vulnerability assessment for cascading failures in electric power systems, IEEE PES Power Systems Conference and Exposition, Seattle WA USA, March 2009.

[51] Transmission reliability evaluation for large-scale systems (TRELSS): version 6.0 User's manual, EPRI, Palo Alto, CA: 2000. 1001035

[52] T. Van Cutsem, Voltage instability: phenomena, countermeasures, and analysis methods, Proceedings of the IEEE, vol. 88, no. 2, Feb 2000, pp. 208-227.

[53] E.T. Whittaker, G.N. Watson, *A Course of Modern Analysis*, Cambridge University Press, Cambridge, UK, 1948.

[54] D.V. Widder, *The Laplace Transform*, Princeton University Press, Princeton, NJ, 1946.

[55] K. R. Wierzbicki, Statistical estimation of cascading blackout size and propagation with branching processes, MS thesis, ECE department, University of Wisconsin-Madison, 2006.

[56] K.R. Wierzbicki, I. Dobson, An approach to statistical estimation of cascading failure propagation in blackouts, CRIS, Third International Conference on Critical Infrastructures, Alexandria, Virginia, September 2006.

[57] N.M. Yanev, On the statistics of branching processes, Theory of Probability and its Applications, vol. 20, 1975, pp. 612-622.

[58] M. Zima, G. Andersson, On security criteria in power systems operation, Power Engineering Society General Meeting, vol. 3, San Francisco CA USA 2005, pp. 3089- 3093.

APPENDIX

J. Kim, I. Dobson, Propagation of load shed in cascading line outages simulated by OPA, COMPENG 2010: Complexity in Engineering, Rome Italy, February 2010. ©2010 IEEE. Reprinted by permission.

COMPENG 2010: Complexity in Engineering, Rome Italy, February 2010 ©IEEE 2010

Propagation of load shed in cascading line outages simulated by OPA

Janghoon Kim, *Student Member IEEE* Ian Dobson, *Fellow IEEE*
Electrical and Computer Engineering Department
University of Wisconsin, Madison WI 53706 USA
dobson@engr.wisc.edu

Abstract

We estimate with a branching process model the propagation of load shed and the probability distribution of load shed in simulated blackouts of an electric power system. The average propagation of the simulated load shed data is estimated and then the initial load shed is discretized and propagated with a Galton-Watson branching process model of cascading failure to estimate the probability distribution of total load shed. We initially test the estimated distribution of total load shed using load shed data generated by the OPA simulation of cascading transmission line outages on the 300 bus IEEE test system. We discuss the effectiveness of the estimator in terms of how many cascades need to be simulated to predict the distribution of load shed accurately.

1 Introduction

Large blackouts are rarer than small blackouts, but are costly to society when they do occur and have substantial risk [12]. Large blackouts generally become widespread by a cascading process of successive failures [18, 22, 23]. It is useful to study mechanisms of cascading failure so that blackout risk may be better quantified and mitigated. The electric power infrastructure is vital in maintaining our society, and maintaining high reliability is especially important as the electric power infrastructure is being transformed in response to changes in new energy sources, new loads, technological advances, sustainability, markets and climate change.

There are many and diverse mechanisms in power systems by which components tripping or failures cause further components tripping [12, 15, 18, 22, 23]. These include line overloads, failures in protection, communication, maintenance or software, various types of instability, and errors in coordination, situational awareness, planning or operations. It is infeasible to analyze a full range of these mechanisms with one simulation, so cascading failure simulations model and analyze a selected subset of these mechanisms [15]. In this paper we analyze load shed data produced by the OPA simulation of cascading line overloads. Each simulated cascade has successive generations in which transmission lines are tripped and load is shed, and the total number of lines tripped and the total amount of load shed are measures of the size of the blackout.

In the OPA simulation model [2], the power system is represented with a standard DC load flow approximation. Starting from a solved base case, blackouts are initiated by random line outages. Whenever a line is outaged, the generation and load is redispatched using standard linear programming methods. The cost function is weighted to ensure that load shedding is avoided where possible. If any lines were overloaded during the optimization, then these lines are outaged with a specified probability. The process of redispatch and testing for outages is iterated until there are no more outages. Then the total load shed is the power lost in the blackout. The OPA model neglects many of the cascading processes in blackouts and the timing of events. However, the OPA model does represent in a simplified way a dynamical process of cascading overloads and outages that is consistent with some basic network and operational constraints. This paper considers a restricted form of the OPA model in which the power grid is fixed and does not evolve or upgrade; in other work the OPA model also represents the complex dynamics of an evolving grid [3, 12, 19].

Branching processes have long been used in a variety of applications to model cascading processes [1, 14], but their application to the risk of cascading failure is recent [7, 8]. In particular, Galton-Watson branching processes give a high-level and tractable probabilistic model of cascading failure. There is some initial evidence that Galton-Watson branching processes can capture some general features of simulated and observed cascading line trips [8, 9, 20] and can approximate other probabilistic models of cascading failure [7, 11, 17]. The branching process gives a simple probabilistic description of the cascading process as an initial disturbance followed by an average tendency for the cascade to propagate in stages until the cascade dies out or all the components fail.

In previous work [9, 10], we obtained cascading failure data from the OPA simulation with 118 and 300 bus IEEE standard test systems, estimated the initial number of lines tripped and average propagation of line trips from this data, and then used the branching process to predict the probability distribution of the total number of lines tripped. This predicted distribution was then shown to match well with the empirical distribution produced by exhaustively running the OPA simulation in most of the cases tested. It is useful to predict the distribution of total number of lines tripped via the branching process because this can be done with significantly fewer simulated cascades. The total number of lines tripped is a measure of blackout size of interest to utilities, whereas load shed is a measure of blackout size and impact of much more direct interest to all users of electricity. Therefore in this paper we test estimating the propagation and probability distribution of load shed.

In contrast with the case of number of lines tripped, which are nonnegative integers, the amounts of load shed are nonnegative real numbers. We estimate the initial distribution of load shed and the average propagation λ from the simulated load shed data. Then we discretize the continuous initial distribution of load shed and use this discrete distribution as the initial distribution of a Galton-Watson branching process with average propagation λ to estimate a discretized distribution of the total load shed.

Our previous work [10, 24] also estimated the initial distribution of load shed and the average propagation λ from simulated load shed data, but then took a different approach using continuous state branching processes [13, 16, 21] to estimate the distribution of the total load shed. The offspring distribution was assumed to be a gamma distribution, with mean λ and variance estimated from the data. Then computer algebra was used to manipulate cumulant generating functions to compute the distribution of total load shed. In this approach, it is not yet known what form of offspring distribution fits power system cascading data well (the gamma distribution was chosen in [10, 24] because it is easy to compute with). Also, there remain challenges in estimating a second parameter of the offspring distribution such as variance and in improving the methods that compute the distribution of load shed for general offspring distributions. These challenges for the approach based directly on continuous state branching processes may be met in the future, but here we are able to suggest an alternative approach that seems simpler.

We assume some background explanations in previous papers. The OPA model is explained in detail in [2] and references to a variety of cascading failure methods and simulations are in [12, 15]. The branching process model and parameter estimation are explained in more detail in [9] and general background on branching processes is in [1, 13, 14].

2 Estimating propagation and distribution of load shed with a branching process

This section describes the procedure for estimating the propagation and probability distribution of load shed with a branching process.

For each simulated cascade the total load shed as well as the load shed at each intermediate generation of the cascade is recorded. The first step is to round very small load shed amounts that are considered negligible (less than 0.5% of total load) to zero. Then the data is modified so that each cascade starts with a nonzero amount of shed. In particular, cascades with no load shed are discarded. The remaining K cascades are those with some non-negligible load shed. Therefore the computed statistics, such as the probability distributions of initial and total load shed, are conditioned on the cascade starting with some non-negligible amount of load shed. Moreover, for the cascades with no load shed in initial generations and non-negligible load shed in subsequent generations, we discard the initial generations with no load shed so that generation zero always starts with a positive amount of load shed.

Now the data has K cascades with non-negligible load shed. Letting X_n^i denote the load shed at generation n of cascade i, the data looks like this:

$$\begin{array}{lcccc}
 & \text{gen. } 0 & \text{gen. } 1 & \text{gen. } 2 & \cdots \\
\text{cascade } 1 & X_0^{(1)} & X_1^{(1)} & X_2^{(1)} & \cdots \\
\text{cascade } 2 & X_0^{(2)} & X_1^{(2)} & X_2^{(2)} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\text{cascade } K & X_0^{(K)} & X_1^{(K)} & X_2^{(K)} & \cdots
\end{array}$$

The total load shed in cascade i is

$$Y^{(i)} = X_0^{(i)} + X_1^{(i)} + \ldots$$

The estimator for the average propagation λ is the standard Harris estimator [6, 13, 14, 25]:

$$\widehat{\lambda} = \frac{\sum_{k=1}^{K}\left(X_1^{(k)} + X_2^{(k)} + ...\right)}{\sum_{k=1}^{K}\left(X_0^{(k)} + X_1^{(k)} + ...\right)} \tag{1}$$

The Harris estimator (1) is an asymptotically unbiased maximum likelihood estimator [14, 25]. Our cascading process is assumed to be subcritical ($\lambda < 1$) and saturation effects are neglected. (In supercritical or saturating cases, other estimators for λ are appropriate as discussed in [10, 17].)

The load shed amounts $X_0^{(1)}, X_0^{(2)}, \cdots, X_0^{(K)}$ are samples from the probability distribution of initial load shed, assuming that some non-negligible load is shed. The average initial load shed θ is estimated as

$$\theta = \frac{1}{K}\sum_{k=0}^{K} X_0^{(k)} \tag{2}$$

To estimate the probability distribution of total load shed from the initial load shed and the estimated propagation $\widehat{\lambda}$, we discretize the samples of the initial load distribution and assume they are propagated by a Galton-Watson branching process with a Poisson offspring distribution of mean $\widehat{\lambda}$.

There are general arguments suggesting that the choice of a Poisson offspring distribution is appropriate [4, 5]. The Poisson distribution is a good approximation when each load shed increment is related to stress on the supply to a large number of other loads so that each load shed increment can be associated with a small, fairly uniform probability of independently leading to other load shed increments in a large number of locations.

We choose a discrete amount of load shed Δ. Then each initial load shed sample X_0^k is discretized to an integer multiple of Δ:

$$Z_0^k = \text{int}\left[\frac{X_0^k}{\Delta} + 0.5\right], \tag{3}$$

where $\text{int}[x]$ = integer part of x. Write Z_0 for the initial load shed expressed in integer multiples of Δ. Then the empirical probability distribution of Z_0 is

$$P[Z_0 = z_0] = \frac{1}{K}\sum_{k=1}^{K} I[Z_0^k = z_0] \tag{4}$$

Now, given the probability distribution (4) of the initial distribution Z_0 and the average propagation estimated from (1), branching process theory implies that the discretized total load shed is distributed according to a mixture of Borel-Tanner distributions:

$$P[Y = r\Delta] = \sum_{z_0=1}^{r} P[Z_0 = z_0] z_0 \lambda (r\lambda)^{r-z_0-1} \frac{e^{-r\lambda}}{(r-z_0)!} \tag{5}$$

3 Results

The cascading failure data is produced by the OPA simulation on the IEEE 300 bus standard test system [26]. Three load levels are considered: 1.0, 1.05 and 1.1 times the base case load. 20 000 cascades were simulated for each load level. The number of cascades K with non-negligible load shed is shown in Table 1 for each load level. The probability of a cascade with non-negligible load shed (that is, a significant blackout) is $K/(20\,000)$.

For the IEEE 300 bus system the load shed discretization Δ is chosen to be 952 MW, which is 4% of the base case load of 23 800 MW. This value of Δ is chosen by experimenting with a range of values. (As a possible point of reference, the power system contains 409 lines as discrete elements and each line comprises 0.24% of the total number of lines.) Too small a value of Δ does not allow sufficient samples within each discretization bin to get a good estimate of the frequency of blackouts in that discretization bin. Too large a value of Δ gives insufficient resolution in the load shed. In the cases tested we find that varying Δ by a factor of 2 has not much effect on the results. The choice of Δ does affect the way that the branching process models the cascading load, and we hope that future work will establish more systematic methods for the choice of discretization.

The average propagation λ is estimated using (1) for each load level and is shown in Table 1. The average initial load shed θ estimated using (2) for each load level is also shown in Table 1.

Table 1. Average propagation λ and average initial load shed θ in IEEE 300 bus test system

load level	λ	θ(GW)	K
1.0	0.09	3.72	4137
1.05	0.21	3.57	8568
1.1	0.42	3.29	9381

For the base case load level 1.0, the probability distribution of total load shed estimated via the branching process is compared to the empirical distribution of total load shed in Figure 1. Although both probability distributions are discretized in load, the distribution of total load shed estimated via the branching process has its points joined by a line so it can be clearly distinguished. The match is good, but this is expected in this case since the average propagation $\lambda = 0.09$ is small and the cascading effect is small, so that the distribution of total load is close to the initial distribution of load.

For the higher load level 1.05, the probability distribution of total load shed estimated via the branching process is compared to the empirical distribution of total load shed in Figure 3. The average propagation $\lambda = 0.21$ and the match is good. The empirical initial load shed distribution is shown in Figure 2. The cascading has the effect of changing the initial distribution of load shed into a distribution of total load shed with larger blackouts.

For the higher load level 1.1, the probability distribution of total load shed estimated via the branching process is compared to the empirical distribution of total load shed in Figure 5. The average propagation $\lambda = 0.42$ and the match is good except for the sharply dropping portion of the tail. The empirical initial load shed distribution is shown in Figure 4. The cascading has a larger effect of changing the

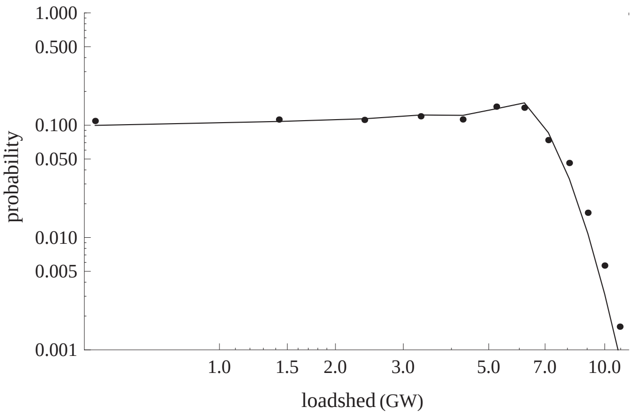


Figure 1. Probability distributions of total load shed for IEEE 300 bus system at load level 1.0. Dots are the empirical distribution; line is estimated with the branching process.

initial load shed distribution into the total load shed distribution.

4 Number of cascades for accurate estimates

This section roughly estimates how many fewer cascades are needed to estimate propagation and then estimate the probability distribution of load shed with the branching process compared to direct empirical estimation of the probability distribution of load shed.

In our case of a Poisson offspring distribution, the asymptotic standard deviation of the Harris estimator can be worked out using the methods of [25] to be

$$\sigma(\widehat{\lambda}) \sim \frac{\sqrt{\lambda(1-\lambda)}}{\sqrt{K\theta/\Delta}} \tag{6}$$

Note that θ/Δ estimates $EX_0/\Delta = EZ_0$, which is the mean number of discretized amounts of initial load shed.

Let $p_{\rm branch}$ be the probability of shedding total load S, computed via estimating λ from $K_{\rm branch}$ simulated cascades with non-negligible load shed and then using the branching process model. $p_{\rm branch}$ is conditioned on a non-negligible amount of load shed. Assume that the initial distribution of load shed is known with high accuracy. Then the standard deviation of $p_{\rm branch}$ is

$$\begin{aligned}\sigma(p_{\rm branch}) &= \left|\frac{dp_{\rm branch}}{d\lambda}\right| \sigma(\widehat{\lambda}) \\ &= \left|\frac{dp_{\rm branch}}{d\lambda}\right| \sqrt{\frac{\lambda(1-\lambda)\Delta}{K_{\rm branch}\theta}} \end{aligned} \tag{7}$$

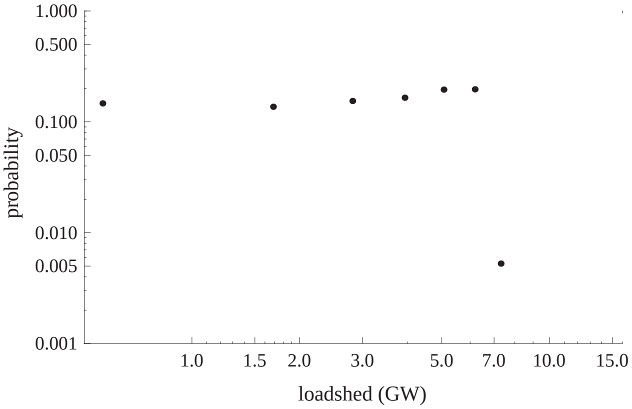


Figure 2. Probability distribution of initial load shed at load level 1.05.

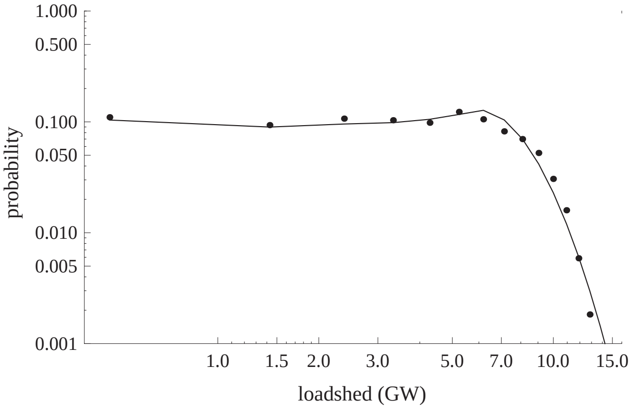


Figure 3. Probability distributions of total load shed at load level 1.05. Dots are the empirical distribution; line is predicted with the branching process.

Let $p_{\rm empiric}$ be the probability of shedding total load S, computed empirically by simulating $K_{\rm empiric}$ cascades with non-negligible load shed. Then the standard deviation of $p_{\rm empiric}$ is

$$\sigma(p_{\rm empiric}) = \sqrt{\frac{p_{\rm empiric}(1-p_{\rm empiric})}{K_{\rm empiric}}} \tag{8}$$

If we require the same standard deviation for both methods, then we can equate (7) and (8) to approximate the ratio

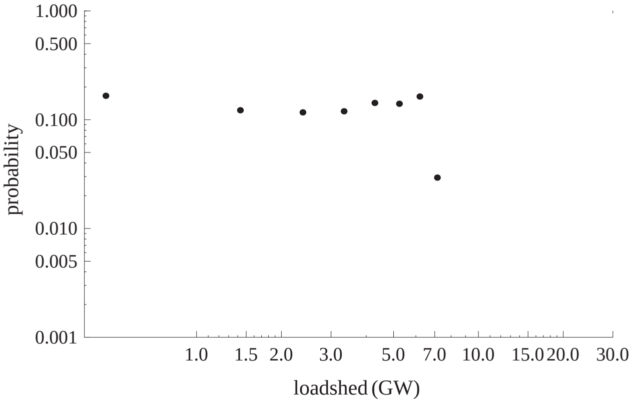


Figure 4. Probability distribution of initial load shed at load level 1.1.

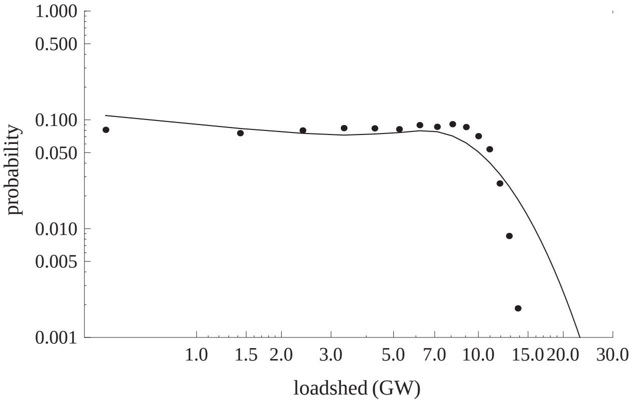


Figure 5. Probability distributions of total load shed at load level 1.1. Dots are the empirical distribution; line is predicted with the branching process.

of the required number of simulated cascades as

$$\frac{K_{\rm empiric}}{K_{\rm branch}} = \frac{p_{\rm empiric}(1-p_{\rm empiric})\theta}{\lambda(1-\lambda)\Delta}\left(\frac{dp_{\rm branch}}{d\lambda}\right)^{-2} \tag{9}$$

To obtain a rough estimate of the ratio, we evaluate (9) for total load shed $S = 9.52$ GW for each of the three load levels. $dp_{\rm branch}/d\lambda$ is estimated by numerical differencing. We find that $K_{\rm empiric}$ exceeds $K_{\rm branch}$ by an order of magnitude or more.

5 Conclusion

In this paper, we suggest approximating the cascading process of load shed in blackouts by discretizing the load shed and then using a Galton-Watson branching process. The average propagation of failures λ is estimated using the standard Harris estimator from cascading load shed data that records the load shed in each cascade generation. Then the branching process model estimates the probability distribution of load shed from the discretized distribution of initial load shed and the estimate of λ. We test this estimation on cascading failure data from the OPA simulation of cascading transmission line outages in the 300 bus IEEE electric power test system. The estimated distribution is close to the empirical distribution in most of the cases tested, suggesting that the branching process model with an averaged propagation can capture some aspects of the cascading of load shed, at least for the purpose of estimating the probability distribution of total load shed.

The approach via propagation and the branching process opens opportunities for estimation of the probability distribution of load shed from fewer observed or simulated cascades. We assume that the probability distribution of initial load shed is known accurately. These initial load shed statistics can be estimated by methods of conventional reliability or by observations, since some load is shed much more frequently than there is a large cascading blackout. Given that the probability distribution of initial load shed is known accurately, our initial testing of the estimation via the branching process of the probability distribution of total load shed suggests that an order of magnitude or more fewer cascades are needed for this estimation in the tail of the distribution than is needed for direct empirical estimation of the probability distribution of load shed. This is not only helpful in reducing simulation times, which are always burdensome and often prohibitive for cascading failure simulations of large power system models, but also will be a crucial attribute in designing practical methods of estimating the probability distribution of load shed from cascades observed in the power system. Empirical methods of accumulating blackout statistics that simply wait for enough cascades to occur take too long to be practical when estimating the rare but important large blackouts in the tail of the distribution. Model based approaches to cascading failure such as the method presented here are needed to estimate the probability of large blackouts from observations over a time scale of about a year rather than over decades.

The approach seems to be easier than a previous method [10, 24] that estimates the offspring distribution of a continuous state branching process and then uses computer algebra to compute cumulant generating functions of the distribution of total load shed.

This paper estimates average propagation and the distribution of load shed using a branching process. These first results are sufficiently promising that further testing with other power system models or more detailed cascading failure simulations is warranted.

Acknowledgments

We gratefully acknowledge funding provided in part by the California Energy Commission, Public Interest Energy Research Program. This paper does not necessarily represent the views of the Energy Commission, its employees or the State of California. It has not been approved or disapproved by the Energy Commission nor has the Energy Commission passed upon the accuracy or adequacy of the information. We gratefully acknowledge support in part from NSF grants SES-0623985 and ECCS-0606003. We gratefully acknowledge that this paper is an account of work sponsored in part by the Power Systems Engineering Research Center (PSERC).

References

[1] K.B. Athreya, P.E. Ney, Branching Processes, Dover NY 2004 (reprint of Springer-verlag Berlin 1972).

[2] B.A. Carreras, V.E. Lynch, I. Dobson, D.E. Newman, Critical points and transitions in an electric power transmission model for cascading failure blackouts, Chaos, vol. 12, no. 4, December 2002, pp. 985-994.

[3] B.A. Carreras, V.E. Lynch, I. Dobson, D.E. Newman, Complex dynamics of blackouts in power transmission systems, Chaos, vol. 14, no. 3, September 2004, pp. 643-652.

[4] P.C. Consul, M.M. Shoukri, Some chance mechanisms related to a generating Poisson probability model, American Journal of Mathematical and Management Sciences, vol. 8, nos. 1 and 2,1988, pp. 181-202.

[5] P.C. Consul, Generalized Poisson Distributions, Dekker NY, 1989.

[6] J-P. Dion, N. Keiding, Statistical inference in branching processes, pp. 105-140. In A. Joffe, P. Ney (eds). Branching Processes. Marcel Dekker, New York 1978.

[7] I. Dobson, B.A. Carreras, D.E. Newman, A branching process approximation to cascading load-dependent system failure, 37th Hawaii International Conference on System Sciences, Hawaii, January 2004.

[8] I. Dobson, B.A. Carreras, D.E. Newman, Branching process models for the exponentially increasing portions of cascading failure blackouts, 38th Hawaii International Conference on System Sciences, January 2005, Hawaii.

[9] I. Dobson, K.R. Wierzbicki, B.A. Carreras, V.E. Lynch, D.E. Newman, An estimator of propagation of cascading failure, 39th Hawaii International Conference on System Sciences, January 2006, Kauai, Hawaii.

[10] I. Dobson, K.R. Wierzbicki, J. Kim, H. Ren, Towards quantifying cascading blackout risk. Bulk Power System Dynamics and Control-VII, Charleston SC, USA, August 2007.

[11] I. Dobson, B.A. Carreras, D.E. Newman, A loading-dependent model of probabilistic cascading failure, Probability in the Engineering and Informational Sciences, vol. 19, no. 1, January 2005.

[12] I. Dobson, B.A. Carreras, V.E. Lynch, D.E. Newman, Complex systems analysis of series of blackouts: cascading failure, critical points, and self-organization, Chaos, vol. 17, 2007, 026103 (13 pages).

[13] P. Guttorp, Statistical inference for branching processes, Wiley, NY, 1991

[14] T.E. Harris, Theory of branching processes, Dover NY 1989.

[15] IEEE PES CAMS Task Force on Cascading Failure, Initial review of methods for cascading failure analysis in electric power transmission systems. IEEE Power & Energy Society General Meeting, Pittsburgh PA USA, July 2008.

[16] P.J.M. Kallenburg, Branching processes with continuous state space, Mathematical Centre Tracts 117, ISBN 90 6196 188 2, Amsterdam 1979.

[17] J. Kim, Properties of the branching model and the cascading model of the failure propagation of the power network, MS thesis, Electrical and Computer Engineering Department, University of Wisconsin, Madison WI 53706 USA, 2008.

[18] NERC (North American Electric Reliability Council), 1996 system disturbances. (Available from NERC, Princeton Forrestal Village, 116-390 Village Boulevard, Princeton, New Jersey 08540-5731), 2002.

[19] H. Ren, I. Dobson, B.A. Carreras, Long-term effect of the n-1 criterion on cascading line outages in an evolving power transmission grid, IEEE Transactions on Power Systems, vol. 23, no. 3, August 2008, pp. 1217-1225.

[20] H. Ren, I. Dobson, Using transmission line outage data to estimate cascading failure propagation in an electric power system. IEEE Transactions on Circuits and Systems Part II, vol. 55, no. 9, 2008, pp. 927-931.

[21] E. Seneta, D. Vere-Jones, On the asymptotic behaviour of subcritical branching processes with continuous state space, Zeitschrift fur Wahrscheinlichkeitstheorie und verwandte Gebiete, vol. 10, pp. 212-225, 1968.

[22] U.S.-Canada Power System Outage Task Force, Final Report on the August 14th blackout in the United States and Canada, United States Department of Energy and National Resources Canada, April 2004.

[23] Union for the Co-ordination of Transmission of Electricity (UCTE), Final report, system disturbance on 4 November 2006. www.ucte.org.

[24] K.R. Wierzbicki, I. Dobson, An approach to statistical estimation of cascading failure propagation in blackouts, CRIS, Third International Conference on Critical Infrastructures, Alexandria, Virginia, September 2006.

[25] N.M. Yanev, On the statistics of branching processes, Theory of Probability and its Applications, vol. 20, 1975, pp. 612-622.

[26] The IEEE 300 bus test system is available at http://www.ee.washington.edu/research/pstca/