



# **Next Generation On-Line Dynamic Security Assessment**

## **Parts III and IV**

*Final Project Report*

**Power Systems Engineering Research Center**

*Empowering Minds to Engineer  
the Future Electric Energy System*



# **Next Generation On-Line Dynamic Security Assessment**

## **Parts III and IV**

### **Final Project Report**

#### **Parts III and IV Project Team**

**James McCalley, Project Leader  
Iowa State University**

**Lei Tang, Graduate Student  
Siddhartha Kumar Khaitan, Research Associate  
Iowa State University**

**PSERC Publication 12-26**

**October 2012**

**For information about Parts III and IV of this project, contact:**

James McCalley  
Harpole Professor of Electrical and Computer Engineering  
Department of Electrical and Computer Engineering  
Iowa State University  
Ames, IA 50011  
Tel: 515-294-4844  
Fax: 515-294-4267  
Email: [jdm@iastate.edu](mailto: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>.

**For additional information, contact:**

Power Systems Engineering Research Center  
Arizona State University  
527 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.

**© 2012 Iowa State University. All rights reserved.**

## **Acknowledgements**

This document contains Parts III and IV of the final report for the Power Systems Engineering Research Center (PSERC) research project titled “Next Generation On-Line Dynamic Security Assessment.” (PSERC project S-38). The overall project leader was Vijay Vittal from Arizona State University. Parts I and II are available in a separate document.

We express our appreciation for the support provided by PSERC’s industrial members and by the National Science Foundation under the Industry/University Cooperative Research Center Program.

We wish to thank:

Jianzhong Tong – PJM

Eugene Litvinov – ISO-NE

Jinan Huang – IREQ

Doug McLaughlin – Southern Co.

Sharma Kolluri – Entergy

Dede Subakti – CAISO

## **Project Executive Summary (Parts I-IV)**

This project addresses five elemental aspects of analysis for the enhanced performance of on-line dynamic security assessment. These five elemental components include: a) A systematic process to determine the right-sized dynamic equivalent for the phenomenon to be analyzed, b) Employing risk based analysis to select multi-element contingencies, c) Increased processing efficiency in decision-tree training, d) Using efficient trajectory sensitivity method to evaluate ability for changing system conditions, and e) Efficient determination of the appropriate level of preventive and/or corrective control action to steer the system away from the boundary of insecurity. An overview of the work accomplished in each of four parts is presented below. This document contains Parts III and IV of the final project report.

### **Part I. Determination of the Right-Sized Dynamic Equivalent (work done at Arizona State University)**

To account for the challenges associated with rapid expansion of modern electric power grid, power system dynamic equivalents have been widely applied for the purpose of reducing the computational effort of dynamic security assessment. Dynamic equivalents are commonly developed using a coherency based dynamic equivalencing approach in which a study area and external area are first demarcated. Then the coherency patterns of the generators in the external areas are determined. A commonly used method is to introduce faults on the boundary of the study area and to group the generator with similar dynamic responses in the external area. Other methods, such as slow coherency-based method and weak-link method have also been proposed. As a result, the coherent generators in the external area are equivalenced. Network reduction is then performed at the interface between the study area and the external area to suitably interconnect the equivalent generators. In the process of building a dynamic equivalent, the definition of the retained area can significantly impact the effectiveness of the final reduced system. As more components are included in the retained area, more attributes related to the dynamic characteristics of the study area can be retained. In conventional dynamic equivalencing applications, the study area and external area are arbitrarily determined without examining the phenomenon of interest with respect to the system dynamic behavior. An improperly defined retained area boundary can result in detrimental impact on the effectiveness of the equivalenced model in preserving dynamic characteristics of the original unreduced system. Additionally, under realistic situations, generator coherency information obtained under one particular operating condition might not be applicable to another operating condition. For a new system condition the process of re-evaluating the generator coherency is time-consuming, especially for large-scale power systems. Therefore significant strides can be made if an efficient technique can be developed to predict the variation in coherency behavior as system condition changes.

The approach that has been implemented in this project first considers the system representation that is available in the control center and for which the supervisory control and data acquisition (SCADA) implementation is available. The boundary of this system representation is then transferred to the planning case being considered for dynamic equivalencing. This represents the initial boundary of the study area. Tie-line interfaces

between the study area and external area are identified. Then three criteria, namely the power transfer distribution factors (PTDFs), the estimated generator rotor acceleration during the fault duration, and the oscillatory mode participation factors are applied to identify the critical generators in the initial external area that exert significant impacts on the dynamic performance of the study area. Based on this determination, the buffer area to be additionally retained can be formed by including these identified generators. The coherent generators in the new external area are then equivalenced, followed by network reduction and load aggregation. The proposed approach is efficient in the sense that the criteria, such as PTDFs and mode participation factors can be readily assessed in many commercial software packages (e.g., PowerWorld, PSS/E, and DSA Tools); while the rotor acceleration-based criterion can be readily computed without the need of time-domain simulation.

To account for the impacts of system condition change on dynamic equivalents, an eigensensitivity-based approach has been proposed in this project to trace the changes in generator slow coherency patterns. Instead of computing the slow coherency patterns from scratch for a new operating condition, the proposed method aims at capturing the significant changes in generator slow coherency after either generation level, load level, or system topology is changed. Based on the predicted coherency patterns, the retained area boundary is adjusted by including the critical generators in the initial external area that become tightly coherent with the initial retained area. The advantages of the proposed approach in saving computational time and improving the equivalencing accuracy for varied operating condition are significant.

The research conducted also reveals that the improvement resulting from revising the retained area boundary might become insignificant when the retained area is already large enough. This limitation is important in today's environment because detailed information regarding the component models and system topologies in the entire system is often inaccessible to a signal entity under the restructured environment. To address this issue, a novel hybrid dynamic equivalent, consisting of both a coherency-based equivalent and an artificial neural network (ANN)-based equivalent, has been proposed. The ANN-based equivalent complements the coherency-based equivalent at the retained area boundary buses. It is designed to compensate for the discrepancy between the full system and the conventionally reduced system. The test on a portion of the WECC system shows that the hybrid dynamic equivalent method can improve the accuracy of the coherency-based equivalent for both the trained and untrained cases. Additionally, the measurements collected by the synchronized phasor measurement units (PMUs) also allow the proposed method to improve the dynamic models for on-line dynamic security assessment (DSA).

The approaches developed have been tested on a large portion of the WECC system and on a test case provided by ISO-NE which includes a significant portion of the Eastern interconnection. The techniques developed have also been used in conjunction with the new version of DYNRED developed by Powertech Labs. This represents significant large scale testing of the method and demonstrates its capability of technology transfer to PSERC member companies. The student working on this portion of the project also performed a summer internship at ISO-NE where the method was applied to the ISO-NE system and demonstrated.

## **Part II. Trajectory Sensitivity Analysis to Determine Stability Limits Under Changing System Conditions (work done at Arizona State University)**

Currently some utilities have existing time domain simulation based approaches to perform DSA in near real time. However, when the network topology or the operating condition changes significantly in a short time horizon, the derivation of the stability limits is computationally burdensome. In this part of research effort, an approach to compute stability limits based on a computationally enhanced trajectory sensitivity analysis has been developed to improve the computational efficiency and accuracy. The most attractive advantage of the trajectory sensitivity approach is that it can provide valuable insights into system responses due to parameter changes within a very short time at the expense of only a negligible amount of extra computational burden.

Firstly, various system parameters sensitivity calculation software packages have been implemented. The implementations include sensitivity to system generation change, load change, load modeling parameters change, generator control parameters change, network topology change and FACTS control parameters change. A Graphic User Interface (GUI) is also developed to simplify the usage. The implementations are fully tested by comparing the approximated system variable trajectories based on the base case with the actual perturbed trajectories obtained by running repeated time domain simulation for the changed condition. The results show the correctness of the implementation and enhanced performance of the trajectory sensitivity method.

Secondly, the computational efficiency problem is also addressed. The trajectory sensitivity method requires augmenting the existing system differential algebraic equations (DAEs) with a new set of sensitivity DAEs corresponding to each system parameter changes. This increases the computation burden. A high performance parallel computing platform has been utilized to reduce this burden. It is observed that each parameter sensitivity calculation is independent of other parameter sensitivity calculations. Moreover, they share the same Jacobian matrices for the solutions. Therefore, when multiples element sensitivities are evaluated, they can be performed in this cluster simultaneously. The test of this cluster shows great efficiency improvement.

The third issue tackled in this research effort is the linear approximation accuracy. The application basis of the trajectory sensitivity method is the first order linear approximation. When there is a small change in certain parameter, the system responses for this changed condition can be approximated based on the base case and the sensitivities evaluated along the base case. However, there is no quantitative measurement on the relation between the linear approximation accuracy and the perturbation size. In this research effort, it is found that there is a relationship between the bound on the linear approximation error and the bound on the product of the maximum normalized sensitivity and the perturbation size. This relationship appears to be system independent and system operating point independent. Thus, the error-perturbation size analysis method based on this finding serves as a general application guide to evaluate the accuracy of the linear approximation.

The uncertainty problem of power system modeling is also addressed using the trajectory sensitivity method. Currently most widely used methods mainly rely on the Monte Carlo type simulation to estimate the probability distribution of the outputs. These methods

require repeated simulations for each possible set of values of load models. Therefore, these approaches are computationally intensive. The trajectory sensitivity analysis can provide an alternative approach to deal with this problem. When load modeling parameter uncertainty is considered, the possible system operational boundary can be obtained by linear approximation with load parameter sensitivity information evaluated along a base case simulation in the time horizon. Based on this operational boundary, the amount of control needed to maintain the system voltage stability should not be fixed. Rather it should be within a certain defined range. An example to study the uncertainty of the composite load modeling and its effect on the system voltage stability problem is given.

The applicability of the trajectory sensitivity approach to a large realistic network is demonstrated in detail. This work applies the trajectory sensitivity analysis method to the WECC system. Several typical power system stability problems have been addressed:

- The angle stability problem

A systematic preventive control analysis method in terms of generation rescheduling to maintain the system transient angle stability is developed and demonstrated

- The voltage stability problem

The trajectory sensitivity approach is used to determine the system operational boundary corresponding to a set of uncertain parameters. Several preventive control actions are then determined according to this operational boundary rather than one fixed operating point. This consideration provides border information for control decision making

- Interface real power flow limit calculation

The trajectory sensitivity method is also applied to calculate the interface real power flow limit. First the generation limits of the key generators in the interface are calculated utilizing the trajectory sensitivity method. Then the flow limit through the interface can be determined by the limits of the key generators and their PTDFs.

All these approaches have been demonstrated on a large realistic model of the WECC system. The software was developed in conjunction with an open source time domain software package called PSAT. All the elements of the development can be easily transported to any member company for demonstration and use.

### **Part III. Introduction to Dynamic Security Assessment Processing System and Applications to High Probability Events (work done at Iowa State University)**

This section introduces a “next generation” design for dynamic security assessment which we refer to as the Dynamic Security Assessment Processing System (DSAPS). The DSAPS operates for two classes of events: high probability events such as NERC category B and C events, and low probability events such as NERC category D events. In Chapter 1, we briefly describe our developed risk-based method for event selection and, once selected, how to distinguish between high and low probability events. The remainder of this section focuses on DSAPS applications developed within this project for high probability events. There are three: load balancing for parallelization of

contingency evaluation within DSAPS, automated failure detection from simulation swing curves, and transient security-constrained optimal power flow.

An effective and practical way of enhancing computational speed for contingency evaluation within DSAPS is to parallelize via use of multiple processors simultaneously handling different contingencies. However, a serious impediment to this approach is the load balancing, so that processor idle time is minimized. In Chapter 2, we identify, illustrate, and compare methods of static and dynamic load balancing.

A key problem in dynamic security assessment is the ability to automate detection of instability from the swing curves produced by the time-domain simulator. In Chapter 3, we briefly summarize a method to do this which is based on Lyapunov exponents.

Preventive control to prevent transient instability is generally handled off-line, by identifying conservative operating rules, usually in terms of generation limits, that are imposed in operational procedures. In Chapter 3, we report on a method to embed transient instability constraints within an optimal power flow program based on trajectory sensitivities. The trajectory sensitivities can be updated in real time, accurately reflecting the necessary constraints, so that production costs are not over penalized. To this end, we describe our Transient Stability Constrained Optimal Power Flow (TSCOPF).

#### **Part IV. Dynamic Security Assessment Processing System and Application to Low Probability Events (work done at Iowa State University)**

A careful review of the major blackouts for the past decades reveals that the power system is experiencing increasing risk worldwide. About half of such events are “slow,” meaning they unfold over a period of minutes to hours, which is enough time for human intervention to be effective. In this project, we have extended previous work to further develop a system for tracking “slow” cascading events. Our approach defends against low probability, high consequence events. The approach makes use of the state-of-art computing facilities for extended term time domain simulation to automate design corrective actions for low-probability high-consequence events. Chapter 4 introduces the approach in terms its two operational modes: anticipatory mode and emergency mode.

In the anticipatory mode, a set of identified high consequence initiating contingencies are continuously analyzed under forecasted near-term conditions. The events are selected via the functional group decomposition described in Chapter 1 which uses topology processing for identifying initiating events in sets of roughly equal probability, accounting for substation breaker-switch configuration. This approach determines the number of element outages which occur for sets of single failure events, e.g., a single fault on a line (probability order  $n=1$ ), and for sets of  $n$ -failure events, e.g., a single fault and a breaker failure to open (probability order  $n=2$ ), resulting in a classification of events by probability and number of element outages. Very high-risk events are those of probability order  $n=1$  which result in an  $N-k$  ( $k \geq 2$ ) event. Events of probability order  $n=2$  that result in a large number of element outages (high  $k$ ) are also considered to be high-risk events.

In the anticipatory mode, initiating events are simulated, and when unacceptable performance is detected, corrective controls are designed for each one and stored in a database. The database contains the information of initiating contingencies, conditions

and corresponding corrective controls. This information is extracted and utilized for decision support immediately after a contingency occurrence.

The anticipatory mode relies on effective corrective control design to be effective. However, initiating events or cascading sequences can occur that have not been assessed a-priori. We address such events using the emergency mode. The emergency mode is utilized when DSAPS fails to find the event in the database or it finds the event but its simulated conditions are not within proximity of the existing conditions. In this case, the high speed extended term time domain simulator is initiated, executing faster than real time such that unacceptable system performance is identified before it occurs.

A key attribute, for both the anticipatory mode and the emergency mode, is high-speed simulation. To this end, we have initiated development of a high-speed extended term time domain simulator (HSET-TDS). The current state of this simulator and our planned future extensions are described in Chapter 5.

### **Project Publications (Parts III and IV)**

Tang, L.; McCalley, J. “An Efficient Transient Stability Constrained Optimal Power Flow based on Trajectory Sensitivity.” North American Power Symposium 2012, Champaign, Illinois, September 2012.

Tang, L.; McCalley, J. “Trajectory Sensitivities: Applications in Power Systems and Estimation Accuracy Refinement.” Submitted to the 4<sup>th</sup> Conference on Innovative Smart Grid Technologies (ISGT 2013).

Khaitan, S; McCalley, J. “TDPSS: A Scalable Time Domain Power System Simulator for Dynamic Security Assessment.” Under review HiPCNA-PG 2012.

Khaitan, S; McCalley, J. “EmPower: An Efficient Load Balancing Approach for Massive Dynamic Contingency Analysis in Power Systems.” Under review HiPCNA-PG 2012.

### **Student Theses (Parts III and IV)**

Tang, Lei. *Next Generation On-Line Dynamic Security Assessment*. Ph.D. thesis in progress at Iowa State University, expected in March 2013.

## Table of Contents

Acknowledgment .....	i
Project Executive Summary (Parts I-IV) .....	ii
Table of Contents .....	viii
List of Figures .....	x
List of Tables .....	xi

### **Part III: Introduction to Dynamic Security Assessment Processing System and Application to High Probability Events ..... 1**

<b>1 Introduction to Dynamic Security Assessment Processing System .....</b>	<b>2</b>
1.1 Event Identification by Functional Group Decomposition .....	3
<b>2 Load Balancing for Parallel Dynamic Contingency Analysis .....</b>	<b>5</b>
2.1 Parallel Contingency Analysis and High Performance Computing .....	5
2.2 Load Balancing .....	5
2.2.1 Static Load Balancing .....	5
2.2.2 Dynamic Load Balancing .....	7
2.3 Implementation .....	11
2.3.1 Static Load Balancing .....	11
2.3.2 Dynamic Load Balancing .....	12
2.4 High Performance Computing at Iowa State University .....	12
2.5 Test System .....	13
2.6 Results and Discussion .....	13
2.7 Conclusion and Future Work .....	14
<b>3 Lyapunov Exponents for Failure Detection .....</b>	<b>16</b>
3.1 Motivation .....	16
3.2 Approach and Computational Efficiency .....	16
<b>4 Transient Stability Constrained Optimal Power Flow .....</b>	<b>17</b>
4.1 Motivation and Literature Review of Transient Instability Constrained OPF ...	17
4.2 Nomenclature .....	18
4.3 Introduction to Trajectory Sensitivity .....	19
4.3.1 Dynamic Trajectory Estimation after Parameter Change .....	21

## Table of Contents (continued)

4.3.2	Other Applications of Trajectory Sensitivity .....	23
4.3.3	Efficient Calculation of Trajectory Sensitivities.....	24
4.4	High-Level Description of TSCOPF-TS .....	25
4.5	Obtaining Transient Stability Constraints .....	26
4.6	Formulation of TSCOPF-TS .....	26
4.7	Case Study .....	28
4.7.1	9-Bus System .....	28
4.7.2	New England 39-Bus System .....	30
4.8	Summary of TSCOPF.....	32

## Part IV: Introduction to Dynamic Security Assessment Processing System and Application to High Probability Events

<b>5</b>	<b>Addressing Low Probability Events .....</b>	<b>34</b>
5.1	Motivation .....	34
5.2	Overview .....	38
5.2.1	Anticipatory Mode .....	38
5.2.2	Emergency Mode .....	39
<b>6</b>	<b>Design and Operation of a High Speed Extended-Term Time Domain Power System Simulator (HSET-TDS) .....</b>	<b>41</b>
6.1	Introduction .....	41
6.2	Literature Review .....	42
6.3	Application and Architecture View .....	43
6.4	Modeling and Design View .....	44
6.4.1	Mathematical Modeling .....	44
6.4.2	Strategy .....	44
6.4.3	Integration Method.....	45
6.4.4	Nonlinear Solver .....	45
6.4.5	Linear Solvers .....	45
6.4.6	Simulation Modeling .....	46
6.5	HSET-TDS Efficiency.....	49
6.5.1	Algorithmic Efficiency .....	49

**Table of Contents (continued)**

6.5.2 Simulation Platform Efficiency ..... 49

6.6 HSET-TDS Validation Results..... 50

6.7 Performance Comparison ..... 51

6.8 Further Development..... 52

**References..... 54**

## List of Figures

Figure 1.1: The Online DSA Process to Address High Consequence Events .....	3
Figure 1.2: Functional Group Decomposition .....	4
Figure 2.1: Static Scheduling: Initial contingency allocation.....	6
Figure 2.2: Static Scheduling: Load on individual processor as time progresses.....	6
Figure 2.3: Static Scheduling: One processor finishes all its tasks and sits idle .....	7
Figure 2.4: Static Scheduling: Multiple idle processors waiting for last task to finish .....	7
Figure 2.5: Master-Slave Scheduling: Initial contingency allocation.....	8
Figure 2.6: Master-Slave Scheduling: Contingency allocation by the master to the slave	9
Figure 2.7: Master-Slave Scheduling: Slave sends a request to the master upon job.....	9
Figure 2.8: Master-Slave Scheduling: Master responds to the request of the slave .....	10
Figure 2.9: Master-Slave Scheduling: Multiple slave nodes request the master node leading .....	10
Figure 2.10: Algorithm 1: The Static Scheduling Algorithm .....	11
Figure 2.11: Algorithm 2: The Master-Slave Scheduling Algorithm .....	12
Figure 2.12: Speedup Compared to 8 Processors .....	14
Figure 4.1: Structure of Power System Model for Dynamics Analysis.....	20
Figure 4.2: New England 39-Bus System.....	22
Figure 4.3: Rotor Angle Trajectory Sensitivity w.r.t. Generator 30 .....	22
Figure 4.4: Trajectories of Generator 30 Rotor Angle.....	23
Figure 4.5: Flowchart of TSCOPF-TS .....	25
Figure 4.6: 9-Bus System.....	28
Figure 4.7: Rotor Angle Comparison between Scenarios.....	29
Figure 5.1: Handling Low Probability Events within DSAPS .....	38
Figure 5.2: Illustration of Anticipatory Mode.....	39
Figure 6.1: Synergistic View of Power System Time Domain Simulator Design.....	41
Figure 6.2: Conceptual Illustration of the Power System .....	47
Figure 6.3: Voltage Profile .....	50
Figure 6.4: Generator Angle Profile .....	50
Figure 6.5: Speed Deviation Profile.....	51

## List of Tables

Table 2.1: Master Slave Scheduling: Simulation Time (in seconds).....	13
Table 2.2: Speedup Values Compared to 8 Processors.....	14
Table 4.1: Comparison Between w/ and w/o Transient Stability Constraints .....	29
Table 4.2: Scenario Description.....	30
Table 4.3: Case Descriptions .....	30
Table 4.4: TSC-OPF Results on New England 39-Bus System .....	31
Table 5.1: Number of Notable Major Blackouts Around the World Since 1965 .....	34
Table 5.2: Summary of Extreme Major Blackouts Since 1965 .....	35
Table 5.3: Analysis of Extreme Major Blackouts Since 1965.....	36
Table 6.1: Example Composition in HSET-TDS .....	48
Table 6.2: Speed Comparison with Different Linear Solvers.....	51
Table 6.3: Speed Comparison with PSSE.....	52

## **Part III**

# **Introduction to Dynamic Security Assessment Processing System and Application to High Probability Events**

**James McCalley, Project Leader**

**Lei Tang, Graduate Student**

**Siddhartha Kumar Khaitan, Research Associate**

**Iowa State University**

# 1 Introduction to Dynamic Security Assessment Processing System

---

Power system security assessment refers to the analysis and quantification of the degree and risk in a power system's ability to survive imminent disturbances (contingencies) without interruption to customer service [1], and then corresponding actions are designed and applied, if necessary, to reduce the risk. Power system security assessment includes steady state security assessment (SSA) and dynamic security assessment (DSA). Steady state security assessment studies the system steady state operating points between dynamic transitions. On the other hand, DSA focuses on the security of the dynamic process in various time frames, from fast transients of several seconds to slow dynamics of several minutes or even hours.

In this project, we have considered the “next generation” of DSA. In doing so, we have conceptualized the Dynamic Security Assessment Processing System (DSAPS) an on-line functionality to perform DSA for two types of events:

**High probability events:** The main feature of this class of events is that, because of their high likelihood, we perform preventive actions to ensure their consequences (or impact), should they occur, are low and satisfy reliability criteria. This event class would include NERC category B and C events. The DSA functionality of EMS today addresses such events.

**Low probability events:** The main feature of this class of events is that, because of their low likelihood, we will not perform preventive actions to mitigate their consequences, since reliability criteria does not require it and since it costs money to do so. Normally, events in this class would be NERC category D events. Today, we do not have any EMS functionality which addresses these events.

Figure 1.1 illustrates our vision of the DSAPS, where we first identify initiating events and classify them into high probability and low probability. In previous work, we have developed a means to accomplish this classification based on processing substation topology; this approach is briefly summarized in Subsection 1.1 below. In this project, we have made contributions to assessment of both high probability and low probability events. Chapters 2-4 address the former and Chapters 5-6 address the latter.

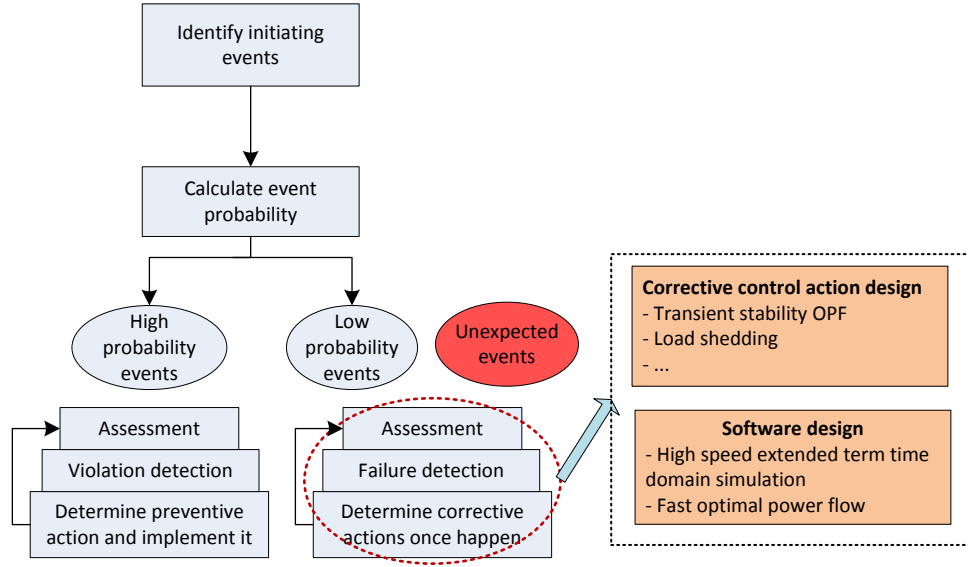


Figure 1.1: The Online DSA Process to Address High Consequence Events

### 1.1 Event Identification by Functional Group Decomposition

Reference [2-4] presented a systematic method to identifying low probability high-consequence events, including not only N-1 contingencies, but also N- $k$  ( $k \geq 2$ ) contingencies. This identification method covers most of the high consequence events that cause the major blackouts. The contingency list is formed through partitioning the whole power system topology into functional groups. A functional group is a group of components that operate and fail together due to their connection structure and protection scheme; it has breakers and open switches as interfaces, as shown in Figure 1.2.

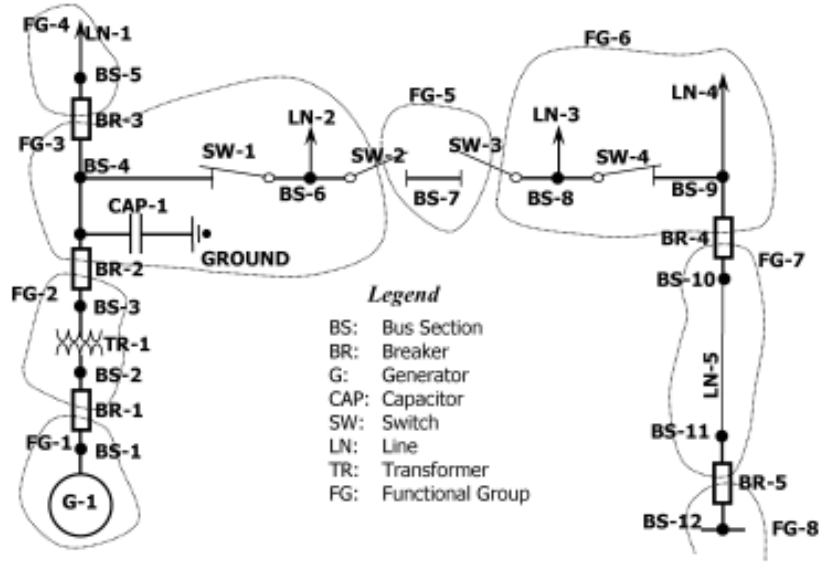


Figure 1.2: Functional Group Decomposition

The process of partitioning the power system topology is called functional group decomposition. Based on the decomposition results, the selected contingencies fall into the following three categories:

- **Functional Group Tripping (FGC):** Trip of one functional group due to failure (fault).
- **Stuck Breaker Contingency (SBC):** Failure (fault) of a functional group, followed by stuck breaker, and then cleared by tripping both the failed functional group and the adjacent functional group.
- **Inadvertent Tripping Contingency (ITC):** Failure (fault) of a functional group, followed by inadvertent tripping of an adjacent line, and then cleared by tripping the failed functional group.

There may be more than one element in a functional group, and more than one functional group tripping at one time, so  $N-k$  ( $k \geq 2$ ) contingencies are identified. A probability order (order of magnitude of the probability) is assigned to each initiating event based on the number of failures necessary for the event to occur. A failure can be a fault, a stuck breaker, or an inadvertent breaker operation.

## 2 Load Balancing for Parallel Dynamic Contingency Analysis

---

### 2.1 Parallel Contingency Analysis and High Performance Computing

For secure power grid operation, dynamic contingency assessment (DCA) is an integral part of power systems operation and markets. Traditionally, simulation studies are carried out for contingency analysis. However, with increasingly complex operation of modern EMS and increasing security requirements, analysis of large number of contingencies is becoming infeasible on existing platforms. To address this challenge, we use high performance computing resources. High performance computing (HPC) refers to parallel processing at a large scale by efficient deployment of multiple processing units in parallel using suitable programming interfaces to solve large problems.

HPC based solutions have been proposed for addressing intractable problems in different domains, such as power systems [5-7], bio-informatics [8, 9], nuclear physics [10] and processor architecture design [11-13]. However, because of the unique characteristics of power system DCA, several issues need to be addressed for benefiting DCA from HPC resources. To maximally utilize the potential of multiple processing units to solve large problems, efficient scaling and parallelization techniques are required. A naive parallelization approach is likely to lead to unbalanced distribution of tasks on the available processors for contingency analysis since the simulation times vary widely across the contingencies. This problem increases dramatically with large number of processors. Thus, novel approaches are required to accelerate power system dynamic contingency analysis using HPC resources.

### 2.2 Load Balancing

The scheduling techniques can be broadly divided into two categories, namely static and dynamic load balancing techniques.

#### 2.2.1 Static Load Balancing

A static load balancing approach pre-computes the schedule for each process in advance and to minimize the runtime overhead of scheduling and monitoring. To achieve good load balancing for all processors, static allocation requires that all the tasks (contingencies) to be available (as in the case of power system) and their characteristics (e.g., precise run-time) to be known precisely at the beginning of the work. However, significant variations in the simulation times of the individual contingency would result in significant load imbalance and improper resource utilization. Except for few processors, all the rest would remain idle, waiting for other processors to finish. Thus the finish time of the schedule is the time for the last job/contingency to finish. Other researchers have shown the ineffectiveness of static scheduling in the case of steady state contingency analysis [14, 15].

This concept of the static allocation is communicated through the medium of Figure 2.1-Figure 2.4. Figure 2.1 shows the initial work load on the different processors.

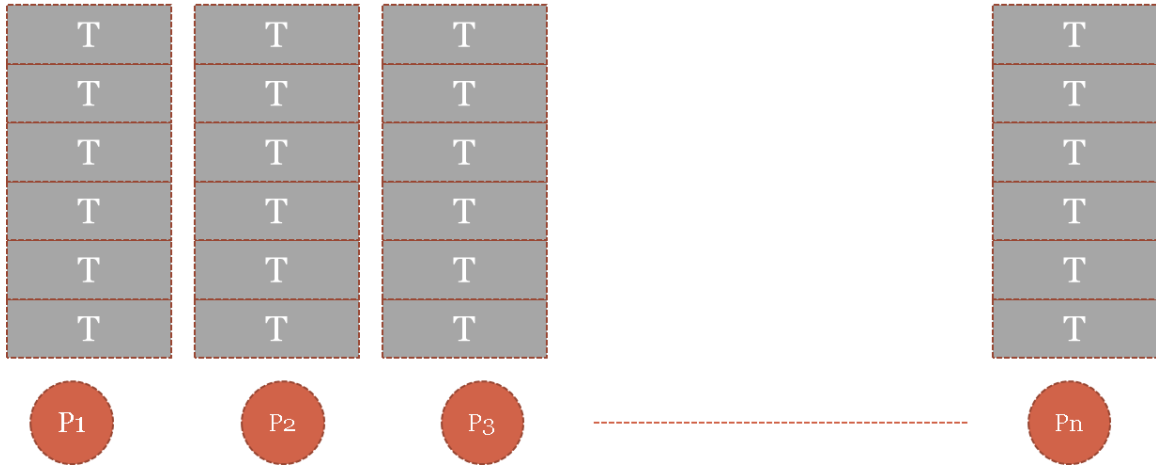


Figure 2.1: Static Scheduling: Initial contingency allocation

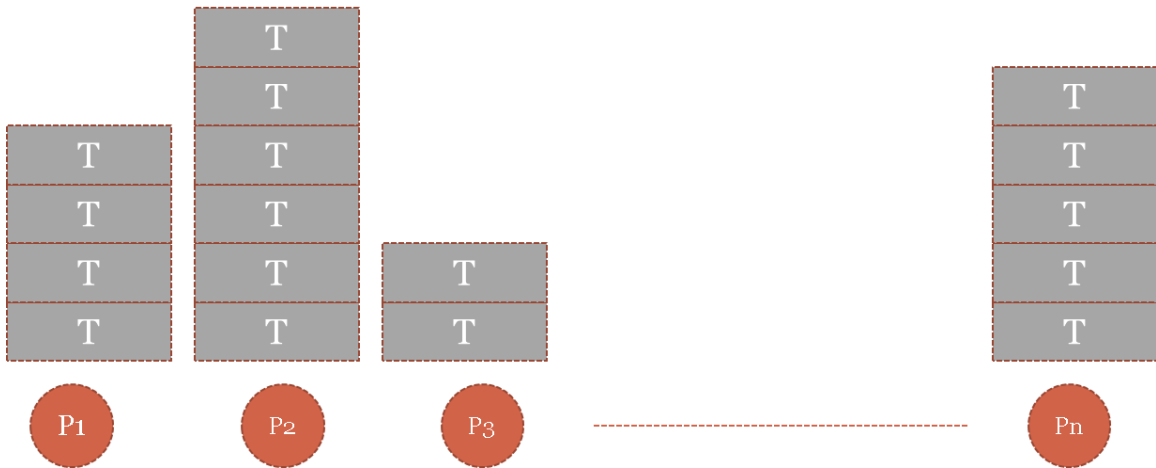


Figure 2.2: Static Scheduling: Load on individual processor as time progresses

In static allocation an attempt is made to load all the processors equally or almost equally in terms of number of tasks/contingencies to be analyzed and the amount of time each processor would take. A simple allocation strategy is to divide the contingencies by number to each processor. Since different contingencies normally have different simulation times, the work load on different processors change with time as shown in Figure 2.2. With further evolution of time, different processors finish their tasks and sit idle waiting for the last job to finish as shown in Figure 2.3 and Figure 2.4.

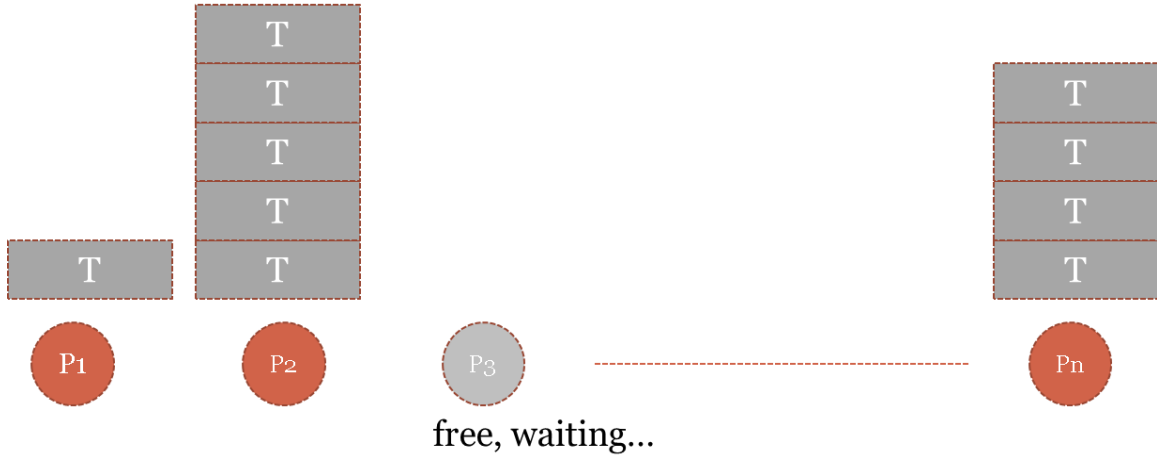


Figure 2.3: Static Scheduling: One processor finishes all its tasks and sits idle



Figure 2.4: Static Scheduling: Multiple idle processors waiting for last task to finish

Thus static scheduling could result high load imbalance if precise run time of different contingencies are not known and allocated randomly to the processors. Thus it is not preferred in most applications. However most of the parallel contingency analysis studies reported in the literature implement static load allocation strategy due to its simplicity in the implementation.

Static load balancing has the advantage that it does not required online scheduling and has low overhead. Also, it is useful when the task lengths are almost equal. However, if task lengths are quite different, then it leads to poor load balancing. This leads to poor computation efficiency and huge resource wastage.

### 2.2.2 Dynamic Load Balancing

Applications that have workloads which are unpredictable, as in the case of contingency analysis, or which change with time during the computation require dynamic load balancing approach to dynamically adjust the decomposition of workloads across the processors. The goal is the minimize processor idle time, keep the communication overhead low and maximize efficiency.

There are a number of dynamic load balancing strategies proposed however a simple dynamic scheduling technique known as the master-slave scheduling is discussed and implemented here. In master-slave strategy one node is used as a master and other nodes are used as slaves which actually perform the computation or do the job (e.g., [14, 15]). The master initially schedules a task queue (generally one job in each queue) on slave-nodes and on finishing a task, the slaves request the master for new task allocation. This technique has the disadvantage that the master-processor becomes occupied with scheduling and hence cannot be used for useful work. If the master also does work then it cannot handle simultaneous requests from the slaves and thus rendering the whole process slow and ineffective.

The concept of the master-slave based dynamic scheduling is communicated through Figure 2.5-Figure 2.9. In the beginning, the master node has all the contingencies queued at its node and all the slaves waiting to receive a task as shown in Figure 2.5. The master then allocates tasks to each of the slave nodes as in Figure 2.6.

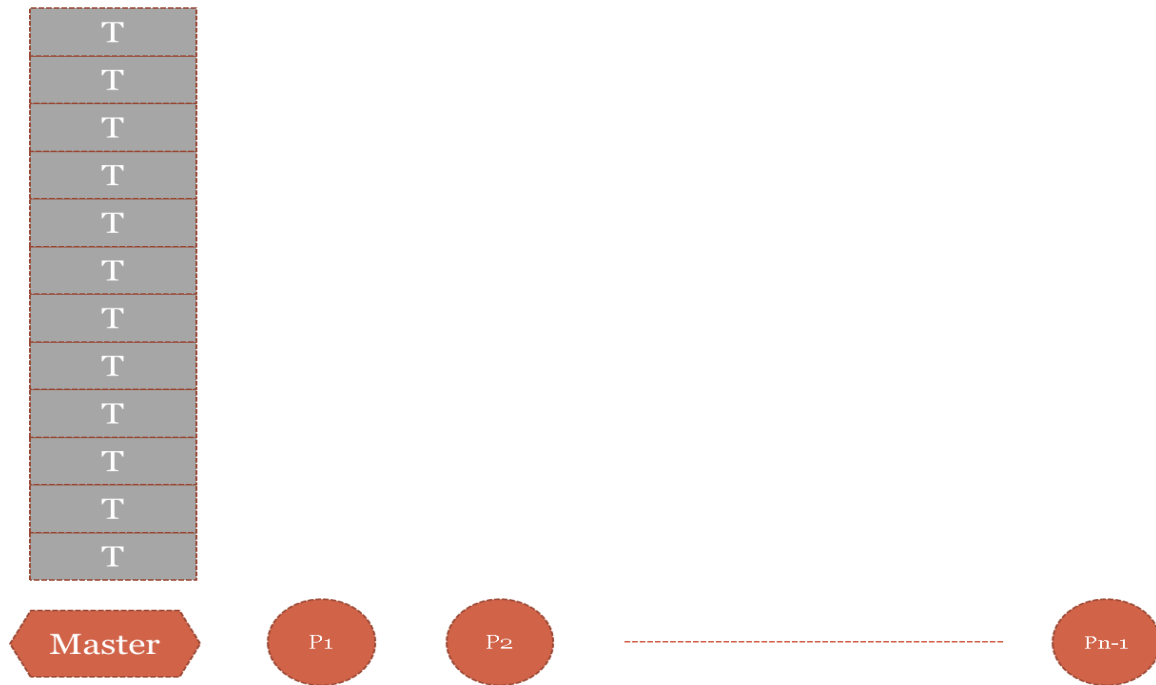


Figure 2.5: Master-Slave Scheduling: Initial contingency allocation

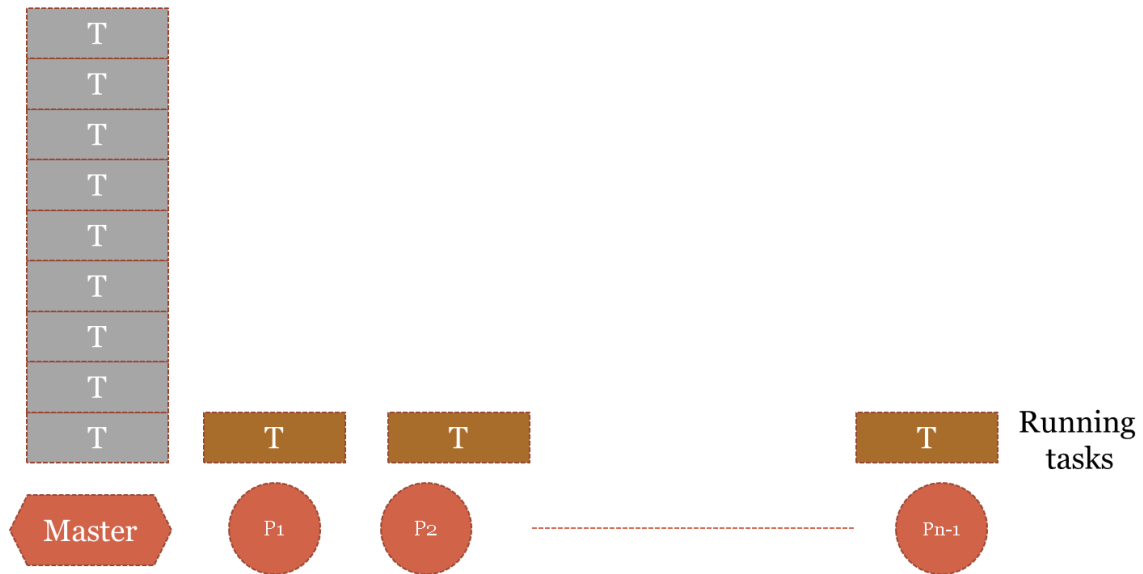


Figure 2.6: Master-Slave Scheduling: Contingency allocation by the master to the slave

When the individual processor finishes the assigned contingency analysis it requests a task from the master node as shown in Figure 2.7. The master node assigns an available task to the processor if it has pending contingencies to be simulated and reduces its queue by one as shown in Figure 2.8.

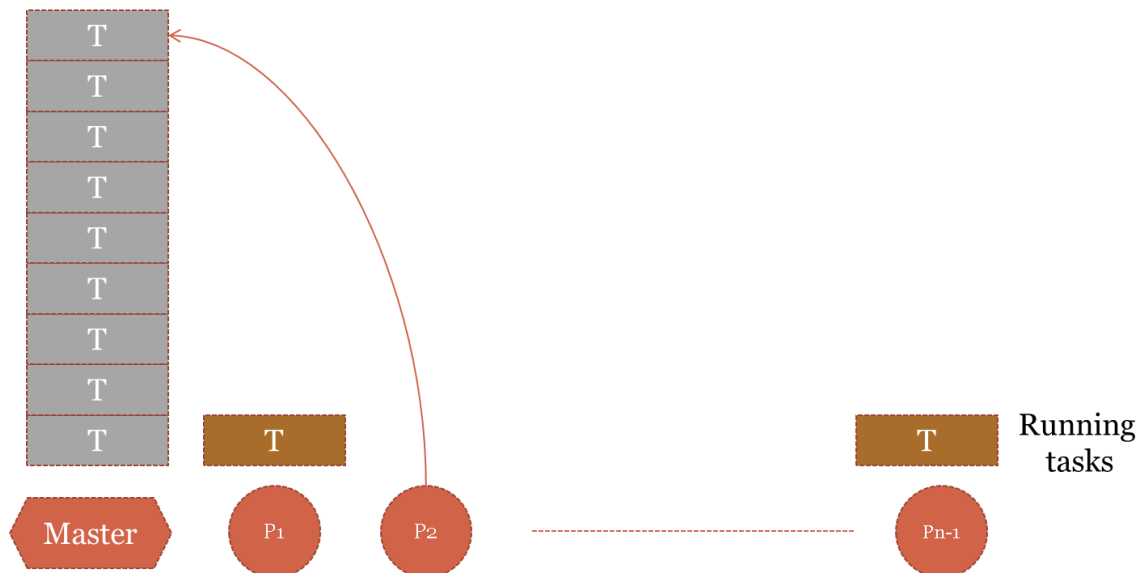


Figure 2.7: Master-Slave Scheduling: Slave sends a request to the master upon job

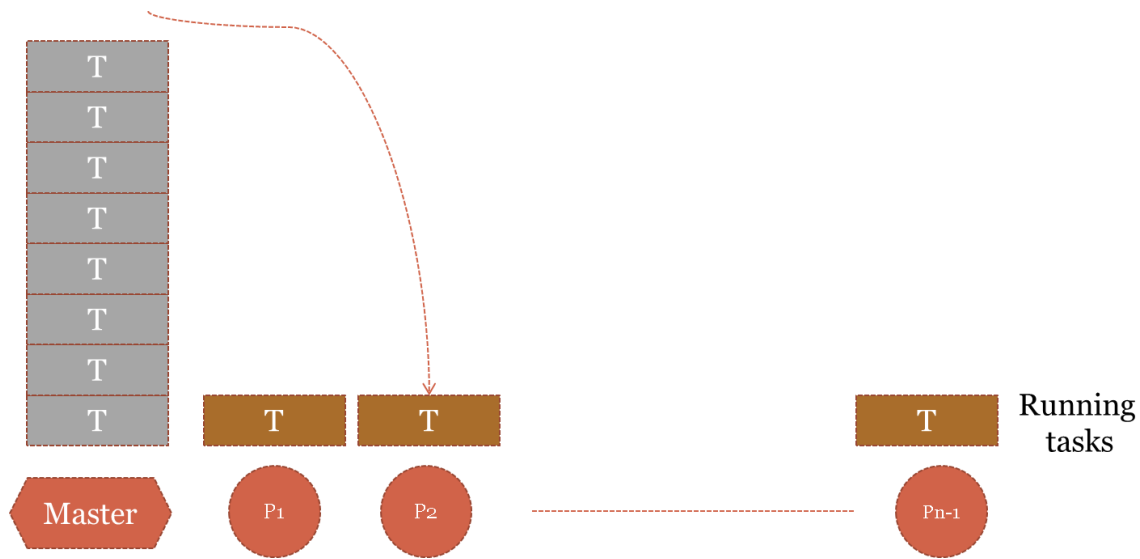


Figure 2.8: Master-Slave Scheduling: Master responds to the request of the slave

However a situation may arise where multiple processors finish their assigned tasks and send request to the master node for additional tasks concurrently. This would lead to contention and would require proper synchronization resulting in potential extra overhead as shown in Figure 2.9. Each of the requesting slaves needs to wait till the master handles the request of the slave. This problem increases dramatically with large number of processors when the chances for contention increase with number of processors.

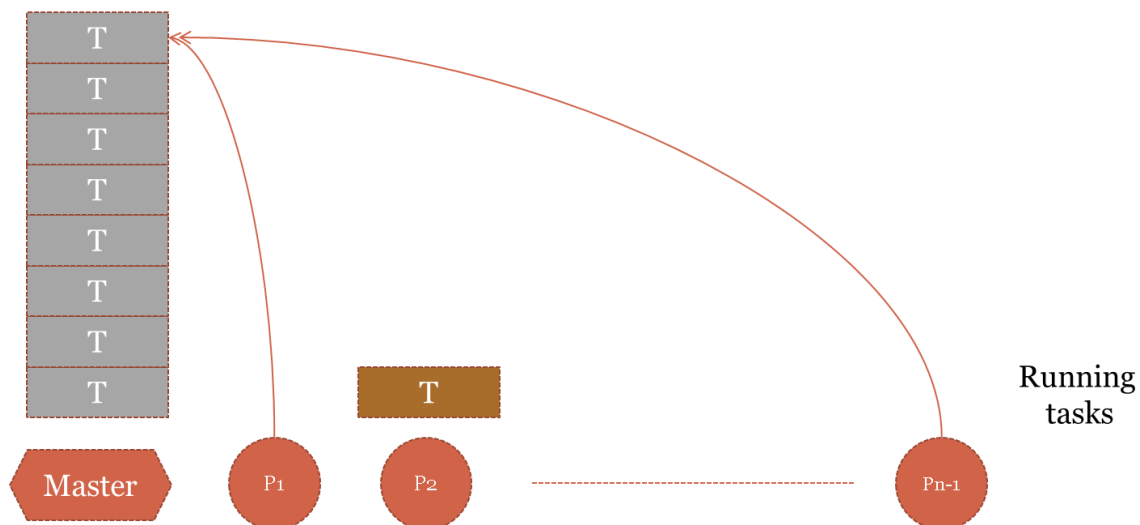


Figure 2.9: Master-Slave Scheduling: Multiple slave nodes request the master node leading

The contention could be alleviated by allocation more than one task to each processor in the beginning and each slave requesting additional tasks as soon as they finish their first task. In this way the slave would not need to wait to begin working on the next task till

the master responds. Further efficiencies could be obtained through developing a reallocation strategy of a pending task in the queue of another slave to a slave which is idle. Master-slave scheduling overcomes the limitations of static scheduling by providing better load balancing. It has low communication overhead. However, master-slave technique also has the disadvantage that the master processor does not perform any useful work and hence gets wasted. Also, if multiple slaves simultaneously make a request to the master, contention may arise.

## 2.3 Implementation

Even an initially balanced distribution of contingencies across the processors could become highly unbalanced with time. Therefore dynamic load balancing algorithms are employed to ensure minimum idle time for the processors and low communication overhead for the inter processor communicators and maximal resource utilization of each processor. In this work we have implemented the dynamic load balancing strategy for the parallel contingency analysis since the different contingencies have widely varying simulation times and a static allocation based strategy would lead to significant load imbalances. We report the results of the master-slave load balancing. We now briefly discuss the algorithms for static and dynamic load balancing approaches.

### 2.3.1 Static Load Balancing

Algorithm 1 in Figure 2.10 shows the pseudo code for the implementation of the static load balancing based parallel contingency analysis algorithm.

```
Input: A task list  $T$  and a processor list  $P$ 
Output: A static allocation of tasks to the processors
while True do
    foreach processor  $p$  in  $P$  do
        if  $T$  is empty then
            break;
        end
        Remove a task  $t$  from  $T$ ;
        Allocate  $t$  on  $p$ ;
    end
end
```

Figure 2.10: Algorithm 1: The Static Scheduling Algorithm

### 2.3.2 Dynamic Load Balancing

Algorithm 2 in Figure 2.11 below show the pseudo code for the dynamic load balancing based on the master-slave.

```
Input: A task list  $T$  and a slave processor list  $S$  and a Master node  $m$   
Output: A load balanced (best effort) allocation of tasks to processors  
//Initialization;  
foreach Slave Processor  $s$  in  $S$  do  
    if  $T$  is empty then  
        break;  
    end  
    Remove a task  $t$  from  $T$ ;  
    Assign  $t$  to  $s$ ;  
end  
Algorithm for Master node  $m$  ;  
while  $T$  is not empty do  
    Wait for the task request from a slave;  
    if a task request arrives from slave  $s$  then  
        Remove a task  $t$  from  $T$  ;  
        Allocate  $t$  to  $s$ ;  
    end  
end  
Algorithm for any slave node  $s$  ;  
while there is a task to be run do  
    Finish the task;  
    Request a task from  $m$ ;  
    if no task is available then  
        break;  
    end  
end
```

Figure 2.11: Algorithm 2: The Master-Slave Scheduling Algorithm

## 2.4 High Performance Computing at Iowa State University

Iowa State University has a supercomputer Cystorm consisting of 400 dual quad core nodes with AMD processors distributed through 12 racks. Thus there are a total of 3200 processors with a peak performance of 15.7 TF and data storage capacity of 44TB. All the results reported here are obtained by simulation on Cystorm supercomputer.

## 2.5 Test System

A large power system test case is selected which has 13,029 buses, 431 generators, 12,488 branches and 5,950 loads. In the simulation, each generator (GENROU) is provided with an exciter (IEEE1) and a governor (TGOV1) model. These are the standard PSSE models and the details of them could be found in PSSE manuals. The different contingencies are generated through bus faults on each of the bus and through a combination of bus fault followed by branch faults on connected branches and/or other branches or branch trips. However it is equally applicable for other combination of contingencies of N-k contingencies. The simulation time of the individual contingency varies from 15 seconds to 25 seconds depending on the number of events (bus fault, bus fault followed by line fault, duration of fault) and their severity in terms of number of cycles of fault duration.

## 2.6 Results and Discussion

Simulations results are reported for thousands of contingencies ranging from 10,000 to 30,000 contingencies. We use the wall clock time (in seconds) to report the results since this would reflect the actual operator time to get the analysis done. The results shown are for the master-slave based dynamic load balancing strategy for contingency analysis.

Table 2.1: Master Slave Scheduling: Simulation Time (in seconds)

Number of Contingencies	P=8	P=16	P=32
10000	18550	8595	4194
20000	35293	16469	7970
30000	52889	24508	12082

Table 2.1 shows the time in seconds required to simulate 10,000, 20,000 and 30,000 contingencies with 8, 16, and 32 processors. Given the large number of contingencies simulated, using only few processors would lead to huge computation time. For example, by interpolating the simulation time with 8 processors, simulating 30000 contingencies with master slave algorithm using a single processor will take nearly 5 days. Therefore it is important to appreciate the role that HPC would play in the secure and reliable operation of the stressed power grid. Table 2.2 shows the speedup result of the master-slave strategy as the number of processors increase.

Table 2.2: Speedup Values Compared to 8 Processors

Number of Contingencies	P=8	P=16	P=32
10000	1	2.16	4.42
20000	1	2.14	4.43
30000	1	2.16	4.38

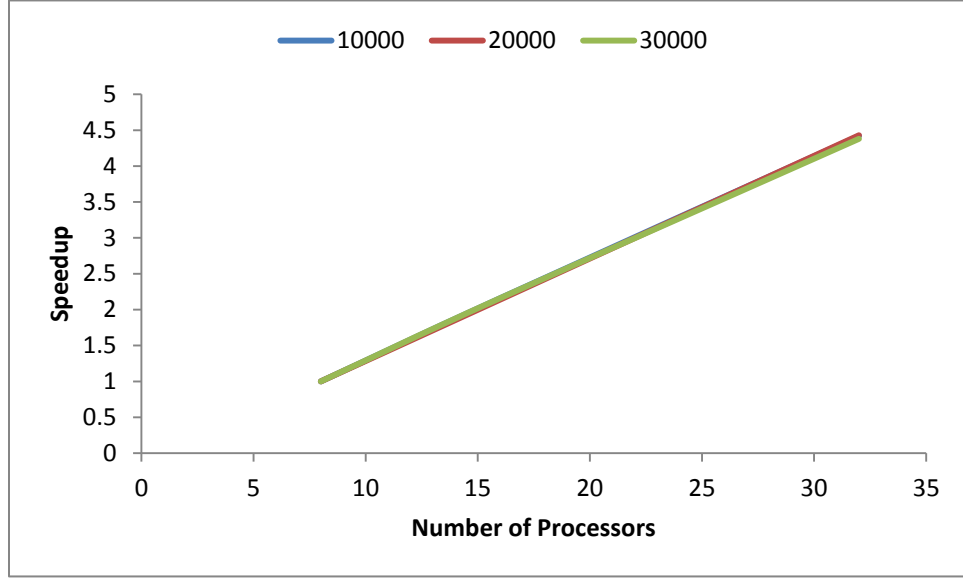


Figure 2.12: Speedup Compared to 8 Processors

Figure 2.12 shows the almost linear speed up as the number of processors increase.

## 2.7 Conclusion and Future Work

A master-slave based dynamic load balancing implementation is effectively done for massively parallel dynamic contingency analysis. A large number of contingencies (up to 30,000) of a large realistic system are simulated. The results show excellent scalability of the proposed algorithm and huge amount of time saving compared with simulating them on one processor. The huge computational saving could allow further exploration of the solution search space, and provides further aid to the operator to take appropriate action when required. However contention could arise on a single master with large number of processors. We are working on an extension of the tradition master-slave idea where more than one task is allocated to each processor in the beginning and each slave requests an additional task as soon as it finishes its first task and starts its queued task. In this way the slave would not need to wait to begin working on the next task till the master responds with a new task. This would alleviate the possible contention on the master. Further efficiencies will be obtained through developing a reallocation strategy of a pending task in the queue of another slave to a slave which is idle. Master-slave

scheduling overcomes the limitations of static scheduling by providing better load balancing. It has low communication overhead. However, master-slave technique also has the disadvantage that the master processor does not perform any useful work and hence gets wasted. To overcome this we are working on developing different load balancing strategies where each processor communicates with each other and does dynamic load balancing rather than a single master allocating tasks. These ideas are being pursued in another project.

### 3 Lyapunov Exponents for Failure Detection

---

#### 3.1 Motivation

For ensuring safe operation of power systems, short-term voltage and angular stability analysis are routinely performed by all energy management systems (EMS). Most existing methods of stability analysis require manual intervention and judgment rendering them unsuitable for real EMS, which operate large power systems with thousands of components. We are working in another funded project to develop failure detection logic and detect system instability utilizing Lyapunov exponent method (LEM).

#### 3.2 Approach and Computational Efficiency

Lyapunov exponents based methods have their roots in the ergodic dynamical theory for nonlinear dynamics. LEMs are used to measure the sensitive dependence on initial state. They measure the rate of divergence of trajectories which are initially infinitesimally separated. The LEM approach in summary utilizes the fact that the Lyapunov exponents of an unstable trajectory have at least one positive Lyapunov exponent which corresponds to the sensitive dependence feature. So, the largest Lyapunov exponent (LLE, also called maximum Lyapunov exponent) of an unstable trajectory must be positive. This indicates that the difference between initial conditions will expand in a particular direction along the trajectory. A negative value of LLE indicates that the difference will contract in a particular direction along the trajectory. This feature of LEM is utilized for failure detection and detecting impending system instability.

Due to its algorithmic procedure, LEM can be used without the need of any manual intervention. LEM could be used both with simulation responses and with PMU measurements to predict stability of the power system. While previous stability analysis approaches based on LEM use computationally inefficient “fixed timestep” integration method to solve power system modeling equations, we are developing algorithms which enable use of LEM with “variable timestep” integration method, thus achieving a magnitude order gain in computational efficiency. Our proposed method can be easily scaled to analyze a large number of contingencies for any system with large number of components.

## 4 Transient Stability Constrained Optimal Power Flow

---

### 4.1 Motivation and Literature Review of Transient Instability Constrained OPF

The optimal power flow (OPF) is used in the operation of power systems for reliability, security and economic efficiency. To better control the operation through dispatch, tremendous improvement has been made since the concept of OPF was first presented in the 1960s [16]. In this chapter, we describe a method to extend the OPF so that it also includes transient stability constraints on the rotor angle through the use of trajectory sensitivity.

Time domain simulation and direct method using energy functions [17, 18] are two widely used transient analysis tools. Time domain simulation is an effective and maybe the most straightforward way for transient stability assessment, but it suffers from its unsatisfactory time efficiency [19]. On the other hand, energy function has its advantage of being fast, but its accuracy and adaptability to complexity of large systems with switching actions prevents its wide application in industry.

Transient stability constrained OPF (TSCOPF) has been studied earlier in reference [20], which embeds the discretized integration process of time domain simulation into the constraints of optimal power flow, with stability limits of rotor angles among one of the constraints. Each iteration would require simulation of a contingency or many contingencies if multi-contingencies are considered. Reference [20-22] derived an equal area criterion based transient stability constraint by transforming rotor angle trajectories of a multi-machine system to the angle space of a single rotor angle trajectory of a one-machine infinite bus (OMIB). Reference [23] also used OMIB, along with differential evolution to search for optimal solution. Reference [24] used trajectory sensitivity for kinetic energy calculation and imposed energy related constraints to secure transient stability. Reference [25] calculated long term rotor angle trajectory sensitivities with respect to generation during peak hours, based on the day-ahead scheduled dispatch, and then transient stability constraints were added to economic dispatch problem.

This chapter introduces an efficient two-step TSCOPF using trajectory sensitivities, called TSCOPF-TS. The proposed TSCOPF-TS is based on time domain simulation. The simulation process in TSCOPF-TS is not embedded in the constraints or the objective function, but rather, it is performed external to the optimization problem, which allows specifically designed simulation tools to help improve the time efficiency of the proposed method. By using trajectory sensitivities, TSCOPF-TS avoids repetitive simulation for different dispatch during the problem solving process. Also, different from reference [26] that calculates trajectory sensitivities for many hours, TSCOPF-TS calculates trajectory sensitivities for only several seconds, which greatly reduces the analysis error due to uncertainties.

The well-structured TSCOPF-TS suggests that its time efficiency meets the speed requirement in industry for dispatch to prevent the potential instability caused by limited credible contingencies (multi-contingency). This is a very important step for its future application in industry.

## 4.2 Nomenclature

$x$	Power system state variables
$y$	Power system algebraic variables
$f(\cdot)$	Power system dynamic equations
$g(\cdot)$	Power system algebraic equations
$\Gamma$	Set of online generators
$\Gamma_n$	Set of online generators at bus $n$
$P_{Gi}$	Active power production of generator $i$
$N$	Set of buses
$P_{In}$	Active power injection at bus $n$
$P_{Dn}$	Active power consumption at bus $n$
$Q_{In}$	Reactive power injection at bus $n$
$Q_{Gn}$	Reactive power production at bus $n$
$Q_{Dn}$	Reactive power consumption at bus $n$
$P_{nm}$	Active power from bus $n$ to bus $m$
$Q_{nm}$	Reactive power from bus $n$ to bus $m$
$\Theta$	Set of buses connected to bus $n$
$V_n$	Voltage magnitude at bus $n$
$\theta_n$	Voltage angle at bus $n$
$V_j$	Voltage magnitude at bus $j$
$\theta_j$	Voltage angle at bus $j$
$G_{nj}$	Element $ij$ of the reduced conductance matrix
$B_{nj}$	Element $ij$ of the reduced susceptance matrix
$P_{Gi}^{\min}$	Minimum active power output of generator $i$
$P_{Gi}^{\max}$	Maximum active power output of generator $i$
$Q_{Gi}^{\min}$	Minimum reactive power output of generator $i$
$Q_{Gi}^{\max}$	Maximum reactive power output of generator $i$
$V_n^{\min}$	Minimum voltage magnitude at bus $n$
$V_n^{\max}$	Maximum voltage magnitude at bus $n$
$I_{nm}^{\min}$	Minimum current magnitude through line $nm$

$I_{nm}^{\max}$	Maximum current magnitude through line $nm$
$P_{nm}^{\max}$	Maximum power through line $nm$
COI	Center of inertia
$\bar{\delta}_i^c(k)$	Rotor angle (taking COI as reference) of generator $i$ at step $k$ during contingency $c$
$\bar{\delta}_{\min}$	Minimum rotor angle (taking COI as reference) of generator $i$
$\bar{\delta}_{\max}$	Maximum rotor angle (taking COI as reference) of generator $i$
$\theta_{ref}$	Voltage angle at reference bus
$\Omega_b$	Nominal frequency
$\omega_i$	Rotor speed of generator $i$
$E_i$	Emf magnitude of generator $i$
$P_{ei}$	Electrical power of generator $i$
$\Delta t$	Integration step size
$\frac{\partial \bar{\delta}_i}{\partial P_j}(t)$	Rotor angle (taking COI as reference) trajectory sensitivity with respect to generation
$P_j^*$	Dispatch of generator $j$ during the base case when the trajectory sensitivities are obtained
$M_i$	Inertia coefficient of generator $i$
$k$	Step series of discretized time domain simulation

### 4.3 Introduction to Trajectory Sensitivity

Figure 4.5 illustrates the procedure to perform TSCOPF-TS. From a high level overview, solving TSCOPF-TS requires two steps to be executed.

Trajectory sensitivity is a powerful tool for dynamic system analysis, and has been used in various areas such as control and power systems. This chapter will introduce the concept of trajectory sensitivity, including its efficient calculation and primary applications for arresting cascading events.

The structure of the power system model for dynamic security assessment is shown in Figure 4.1[27].

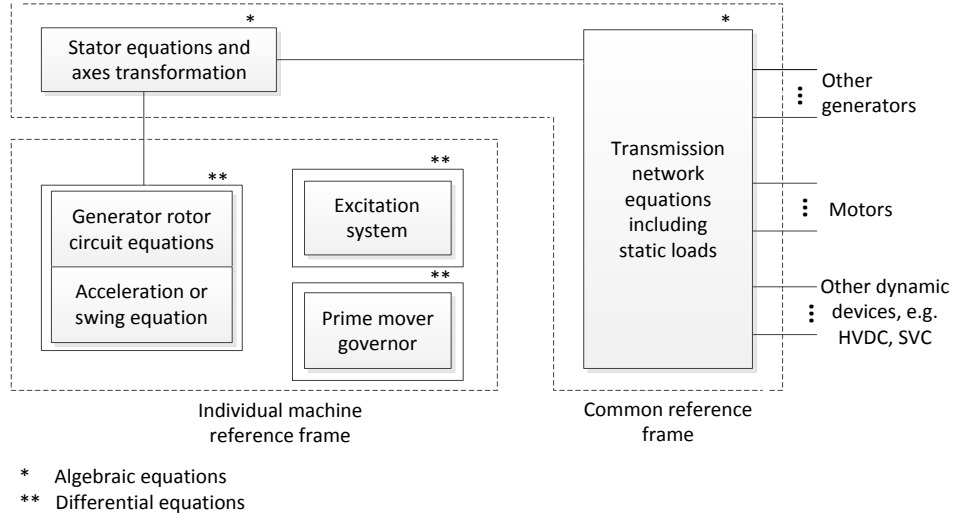


Figure 4.1: Structure of Power System Model for Dynamics Analysis

Mathematically, the dynamics of the power system can be expressed by a set (or several sets) of differential algebraic equations (DAE) as

$$\dot{x} = f(x, y, \lambda) \quad (4.1)$$

$$0 = g(x, y, \lambda) \quad (4.2)$$

where  $x$  is the vector of state variables such as rotor angles and rotor speed,  $y$  is the vector of algebraic variables such as voltages, and  $\lambda$  is the vector of model parameters such as generation levels, load levels and transmission line impedances;  $\lambda$  may also include simulation-specific parameters such as contingency clearing time.

If we take the first derivative of the DAE above with respect to any element of  $\lambda$  that is important to the system, we will get another set of DAE, as

$$\dot{x}_\lambda = f_x x_\lambda + f_y y_\lambda + f_\lambda \quad (4.3)$$

$$0 = g_x x_\lambda + g_y y_\lambda + g_\lambda \quad (4.4)$$

Equations (4.3) and (4.4) are augmentations of the original equations (4.1) and (4.2). Solving for (4.1-4.4) together will yield trajectories of all variable as well as their trajectories of first derivatives with respect to that particular parameter  $\lambda$ , and those additional trajectories are called *trajectory sensitivities*, i.e.

$$\frac{\partial x}{\partial \lambda}(t) = x_\lambda(t) \text{ and } \frac{\partial y}{\partial \lambda}(t) = y_\lambda(t)$$

Applications of trajectory sensitivity mainly lie in four areas:

- Dynamic trajectory estimation after parameter change
- Identifying critical parameter for power system stability
- System limit determination
- Control design

In the following, the first application will be introduced in detail, and the other three applications are briefly reviewed.

#### 4.3.1 Dynamic Trajectory Estimation after Parameter Change

It can be observed that trajectory sensitivity interprets the change of system variables ( $x$  and  $y$ ) with change of parameter  $\lambda$ . This feature significantly helps reduce the burden of power system time domain simulation. This is because time domain simulation is for a specific scenario, and if there is change of a parameter such as generation or load, simulation should be performed again. Trajectory sensitivity can estimate the resulting trajectories after the small parameter change. Suppose we have obtained a set of variable trajectories  $x^*(t)$ , and after that there is a slight change of parameter  $\Delta\lambda$ , then the resulting trajectories of variables can be estimated as

$$x(t) \approx x^*(t) + x_\lambda(t) \cdot \Delta\lambda . \quad (4.5)$$

Note that, estimation using equation (4.5) is actually based on Taylor's series expansion, that is

$$x(\lambda + \Delta\lambda) = x(\lambda) + \frac{x_\lambda}{1!} \Delta\lambda + \text{higher order terms} . \quad (4.6)$$

Equation (4.5) neglects higher terms. Because power systems are nonlinear systems, there will be estimation error, especially when the parameter change is comparably large.

In the following, several typical trajectory sensitivities will be introduced, with primary application to trajectory estimation. The New England 39-bus system is used for testing, and is illustrated in Figure 4.2. The disturbance is a fault at bus 3, cleared by tripping the line from bus 3 to bus 4, after 0.1 second. Trajectory sensitivities of rotor angles with respect to generator 30 are shown in Figure 4.3. Then generator 30 was shifted from 2.5 p.u. to 2.1 p.u., which is a 16% decrease. This shift was compensated by the swing bus (generator 39). Thus, there are changes of two parameters, i.e., the generation levels of generator 30 and of the swing bus. Since trajectory sensitivities are linear, they can be added up together, i.e.

$$r(t) \approx r^*(t) + r_{G30}(t) \cdot \Delta G_{30} + r_{G39}(t) \cdot \Delta G_{39} \quad (4.7)$$

where  $r^*(t)$  is base case generator 30 rotor angle trajectory;  $r_{G30}$  and  $r_{G39}$  are trajectory sensitivities of generator 30 rotor angle with respect to generation levels of generator 30 and 39, respectively;  $r(t)$  is generator 30 rotor angle after the generation shift.

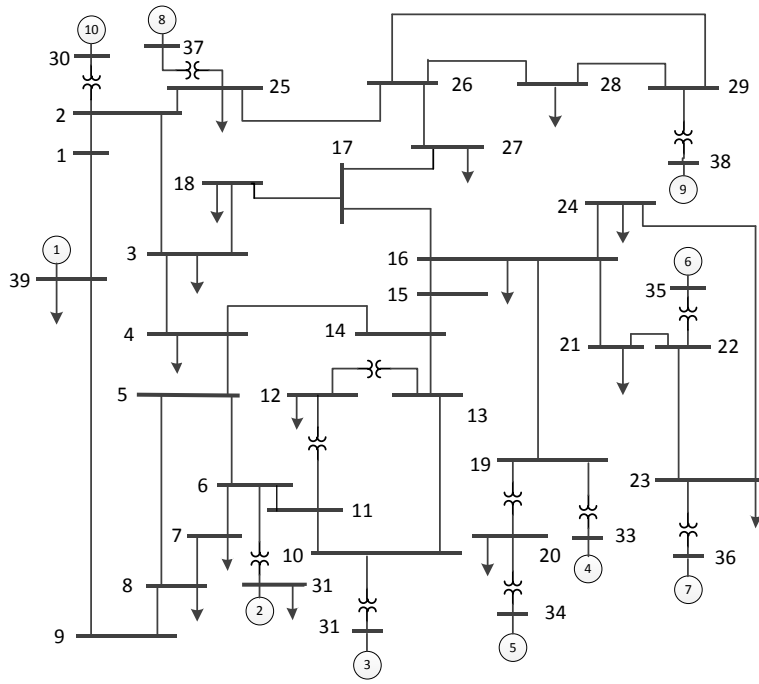


Figure 4.2: New England 39-Bus System

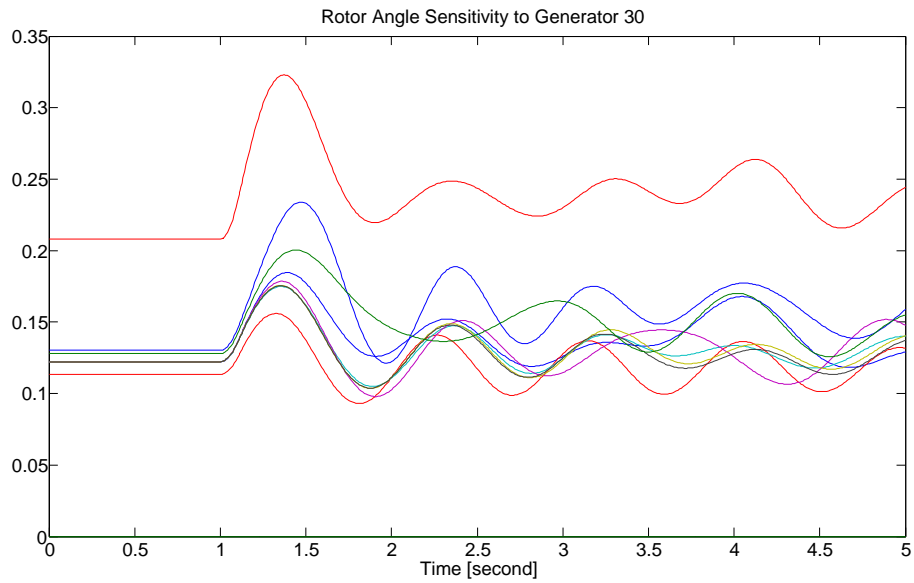


Figure 4.3: Rotor Angle Trajectory Sensitivity w.r.t. Generator 30

Figure 4.4 shows the base case generator 30 rotor angle trajectory, and the trajectories after generation shift, obtained through simulation and estimation. We can see Figure 4.4 that the estimated trajectory almost perfectly tracked the one obtained through time domain simulation. Therefore, the trajectory sensitivity shows good ability in dynamic estimation of rotor angle trajectories after a change of generation levels. This provides very useful information that will be used later in this report for the trajectory sensitivity based transient stability constrained optimal power flow.

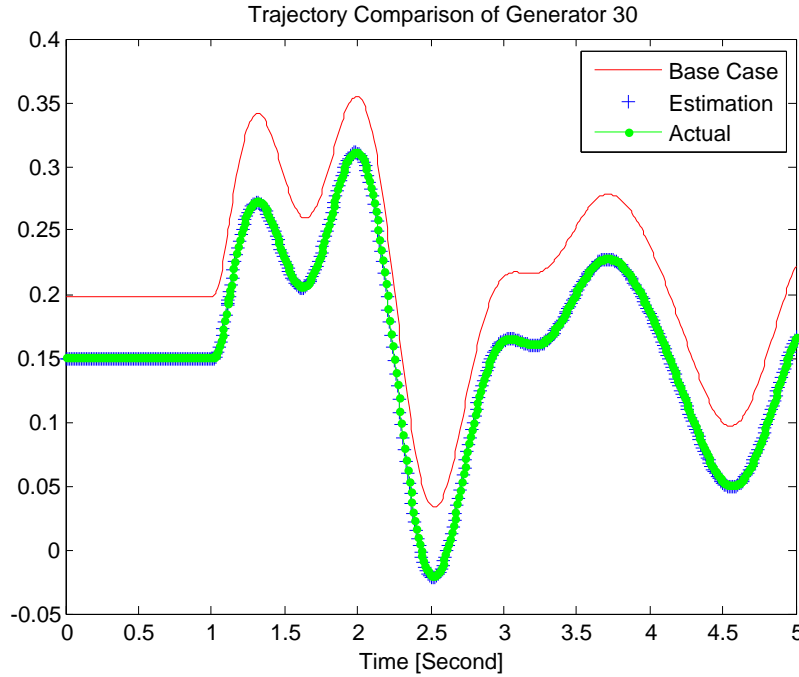


Figure 4.4: Trajectories of Generator 30 Rotor Angle

### 4.3.2 Other Applications of Trajectory Sensitivity

There are at least three other applications of trajectory sensitivities, described in what follows.

- Identifying critical parameters for power system stability

Reference [28] is a very typical example of using trajectory sensitivity to locate critical line, generator, and tripping time for a contingency that has happened. In the case of Nordel power grid disturbance of January 1, 1997, trajectory sensitivity gave very insightful analysis of the post contingency phenomena. It is the first application of trajectory sensitivity on large system.

- System limit determination

The estimation function of trajectory sensitivity helps us reduce the computational cost and thereby avoid repetitive simulations to find the stability limit for power system

operation [29, 30]. A typical example will be introduced later in this report, to find the critical clearing time for avoiding a violation of transient voltage dip criteria.

- Control design

Reference [20-22] provide typical examples to use trajectory sensitivity's estimation ability for control design. For a contingency that will cause instability, critical generators should be found for generation rescheduling. The goal is to stabilize the system through minimum dispatch with least cost. This work also used trajectory sensitivity to design transient stability constrained optimal power flow, discussed later in this report.

### 4.3.3 Efficient Calculation of Trajectory Sensitivities

Trajectory sensitivities can be calculated in an efficient way, if trapezoidal integral rule is used. The integration process for (4.1) and (4.2) using trapezoidal rule is

$$x^{k+1} = x^k + \frac{\eta}{2} [f(x^k, y^k) + f(x^{k+1}, y^{k+1})] \quad (4.7)$$

$$0 = g(x^{k+1}, y^{k+1}), \quad (4.8)$$

where  $k$  is time series index and  $\eta$  is integration time step. Or writing (4.7) and (4.8) in matrix form, we obtain

$$F(\phi^{k+1}) = \begin{bmatrix} \frac{\eta}{2} f(\phi^{k+1}) - x^{k+1} + \frac{\eta}{2} f(x^k, y^k) + x^k \\ g(\phi^{k+1}) \end{bmatrix} \quad (4.9)$$

where  $\phi^{k+1} = [x^{k+1}, y^{k+1}]$ . Newton iteration is used to solve  $F(\phi^{k+1})=0$  according to

$$\phi_{i+1} = \phi_i - F_{\phi}^{-1}(\phi_i) F(\phi_i) \quad (4.10)$$

where  $i$  denotes the Newton iteration step, and  $F_{\phi}$  is the Jacobian matrix of  $F$  with respect to  $\phi^{k+1}$ , i.e.

$$F_{\phi_i}^{k+1} = \begin{bmatrix} \frac{\eta}{2} f_{xi}^{k+1} - I & \frac{\eta}{2} f_{yi}^{k+1} \\ g_{xi}^{k+1} & g_{yi}^{k+1} \end{bmatrix}. \quad (4.11)$$

On the other hand, (4.3) and (4.4) can also be solved using trapezoidal iteration, i.e.

$$x_{\lambda}^{k+1} = x_{\lambda}^k + \frac{\eta}{2} [f_x^k x_{\lambda}^k + f_y^k y_{\lambda}^k + f_{\lambda}^k + f_x^{k+1} x_{\lambda}^{k+1} + f_y^{k+1} y_{\lambda}^{k+1} + f_{\lambda}^{k+1}] \quad (4.12)$$

$$0 = g_x^{k+1} x_{\lambda}^{k+1} + g_y^{k+1} y_{\lambda}^{k+1} + g_{\lambda}^{k+1}. \quad (4.13)$$

Or writing (4.12) and (4.13) in matrix form, we obtain

$$\begin{bmatrix} \frac{\eta}{2} f_x^{k+1} - I & \frac{\eta}{2} f_y^{k+1} \\ g_x^{k+1} & g_y^{k+1} \end{bmatrix} \begin{bmatrix} x_\lambda^{k+1} \\ y_\lambda^{k+1} \end{bmatrix} = \begin{bmatrix} -\frac{\eta}{2} (f_x^k x_\lambda^k + f_y^k y_\lambda^k + f_\lambda^k + f_\lambda^{k+1}) - x_\lambda^k \\ -g_\lambda^{k+1} \end{bmatrix}. \quad (4.14)$$

Solving this equation requires the factorization of the sparse matrix on the left-hand-side, which is the most time consuming part for linear equation solving but can be directly obtained from the last step of Newton iteration in (4.11). Therefore trajectory sensitivities can be calculated as above, with negligible computing time, compared to time domain simulation [31].

#### 4.4 High-Level Description of TSCOPF-TS

Figure 4.5 illustrates the procedure to perform TSCOPF-TS. From a high level, solving TSCOPF-TS requires two steps to be executed.

**Step 1:** Obtain transient stability constraints

**Step 2:** Solve OPF with transient stability constraints

The two steps are boxed with dashed lines in Figure 4.5.

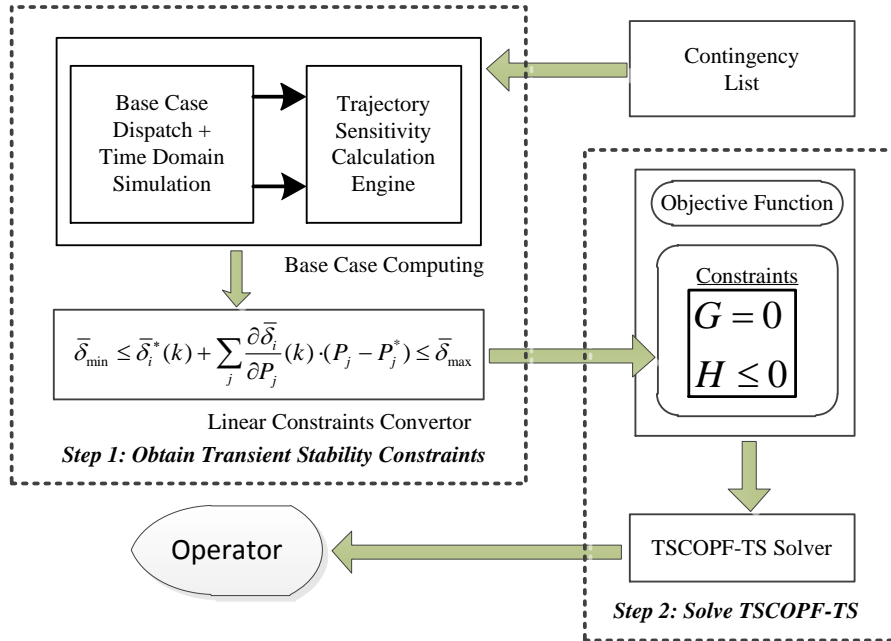


Figure 4.5: Flowchart of TSCOPF-TS

The following explains how the transient stability constraints are obtained and the formulation of TSCOPF-TS.

#### 4.5 Obtaining Transient Stability Constraints

Transient stability constraints are the key constraints in TSCOPF-TS to improve transient stability. They enforce limits on maximum rotor angles with the center of inertia (COI) as the reference frame, according to

$$\bar{\delta}_{\min} \leq \bar{\delta}_i^c(k) \leq \bar{\delta}_{\max}, \quad (4.1)$$

where  $\bar{\delta}_i^c(k)$  are series of discretized rotor angle at each time step  $k$ . The rotor angles are with respect to COI, which is

$$\text{COI}(k) = \frac{\sum_i M_i \delta_i(k)}{\sum_i M_i}. \quad (4.2)$$

For contingency  $c$ ,

$$\bar{\delta}_i^c(k) = \delta_i^c(k) - \text{COI}^c(k). \quad (4.3)$$

Obtaining those rotor angles needs time domain simulation. And even for the same contingency, different dispatch will result in different rotor angle trajectories for the same generator. To avoid repeated simulations for different dispatches, trajectory sensitivities of rotor angles with respect to generation ( $\partial \bar{\delta}_i / \partial P_j(t)$ ) are used. Once the rotor angle trajectories for base-case dispatch are obtained, the consequential trajectories after re-dispatch are calculated as

$$\bar{\delta}_i(k) = \bar{\delta}_i^*(k) + \sum_j \frac{\partial \bar{\delta}_i}{\partial P_j}(k) \cdot (P_j - P_j^*). \quad (4.4)$$

Then for a single contingency, the corresponding discretized transient stability constraints are

$$\bar{\delta}_{\min} \leq \bar{\delta}_i^*(k) + \sum_j \frac{\partial \bar{\delta}_i}{\partial P_j}(k) \cdot (P_j - P_j^*) \leq \bar{\delta}_{\max}, \quad (4.5)$$

which is an expanded expression of (4.1).

#### 4.6 Formulation of TSCOPF-TS

The formulation of TSCOPF-TS is as follows.

### A. Objective function

The objective function is generation cost; it may be linear, quadratic or piece-wise linear. A commonly used and quite representative one is the quadratic function as

$$\min f(P_{Gi}) = \sum_{i \in \Gamma} a_i + b_i P_{Gi} + c_i P_{Gi}^2 \quad (4.6)$$

### B. Power flow equations

Related equations include power balance

$$P_{In} = P_{Gn} - P_{Dn} = \sum_{m \in \Theta_n} P_{nm}(\cdot), \quad \forall n \in N \quad (4.7)$$

$$Q_{In} = Q_{Gn} - Q_{Dn} = \sum_{m \in \Theta_n} Q_{nm}(\cdot), \quad \forall n \in N \quad (4.8)$$

where

$$P_{Gn} = \sum_{i \in \Gamma_n} P_{Gi}, \quad \forall n \in N$$

$$Q_{Gn} = \sum_{i \in \Gamma_n} Q_{Gi}, \quad \forall n \in N$$

and power flow equations (AC power flow is considered, DC power flow can also be used here)

$$P_{In} = V_n \sum_j V_j \left[ G_{nj} \cos(\theta_n - \theta_j) + B_{nj} \sin(\theta_n - \theta_j) \right] \quad (4.9)$$

$$Q_{In} = V_n \sum_j V_j \left[ G_{nj} \sin(\theta_n - \theta_j) - B_{nj} \cos(\theta_n - \theta_j) \right] \quad (4.10)$$

### C. Operational limits

This includes generation limits

$$P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max} \quad (4.11)$$

$$Q_{Gi}^{\min} \leq Q_{Gi} \leq Q_{Gi}^{\max} \quad (4.12)$$

voltage limits

$$V_n^{\min} \leq V_n \leq V_n^{\max} \quad (4.13)$$

and line flow limits.

$$I_{nm}^{\min} \leq I_{nm} \leq I_{nm}^{\max} \quad (4.14)$$

$$P_{nm}^{\min} \leq P_{nm} \leq P_{nm}^{\max} \quad (4.15)$$

#### D. Steady state security constraints

These constraints are used for preventive control of steady state contingencies, with constraints similar to basic ones, except the power flow equations reflect contingency conditions [32]. The OPF with steady state security constraints is called the security constrained OPF (SCOPF).

#### E. Transient stability constraints

The transient stability constraints for first swing rotor angle stability are in (4.5)

#### F. Other Constraints

Another constraint to ensure angles are specified within a single 360 degree rotation follows:

$$-\pi \leq \theta_n \leq \pi, \quad \forall n \in N \quad (4.16)$$

$$\theta_{ref} = 0 \quad (4.17)$$

### 4.7 Case Study

#### 4.7.1 9-Bus System

Testing of the proposed TSCOPF-TS is initially performed using a 9-bus system, as shown in Figure 4.6.

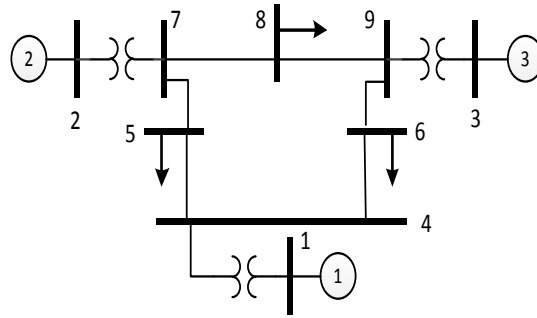


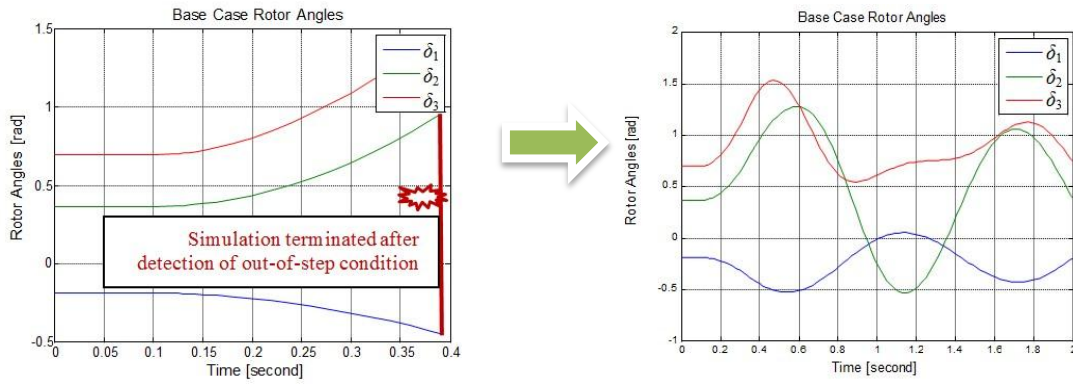
Figure 4.6: 9-Bus System

Only one contingency is considered in this analysis. This contingency is a fault at the bus 5, at the end of the transmission line from bus 5 to bus 7. It is cleared by tripping that line after 0.29 seconds. Table 4.1 compares the dispatch results with and without transient

stability constraints considered. We observe from Table 4.1 that generator 3 was decreased, compensated by generators 1 and 2. The cost increased from 5296.69 \$/MW to 5313.35 \$/MW (0.4%), due to adding the transient constraints. Figure 4.7 compares the rotor angle response of the three generators following the contingency, using the calculated dispatch with and without the transient constraints. We observe from Figure 4.7 that, the rotor angle stability was maintained after adding the transient stability constraints.

Table 4.1: Comparison Between w/ and w/o Transient Stability Constraints

	P1 (MW)	P2 (MW)	P3 (MW)	Cost (\$/MW)
W/	82.73	131.65	104.14	5296.69
W/O	89.80	134.32	94.19	5315.35



Rotor angles (a) w/o (b) w/ transient stability constraints

Figure 4.7: Rotor Angle Comparison between Scenarios

And it can be observed from Figure 4.7 (b) that, since we have set

$$\bar{\delta}_{\min}^c = -0.5 \text{ rad}$$

$$\bar{\delta}_{\max}^c = 1.5 \text{ rad} ,$$

the rotor angles are just within the pre-set angle range, with the small excess due to the interpolation error using trajectory sensitivities. The range of rotor angles should not exceed 180 degrees.

### 4.7.2 New England 39-Bus System

The New England 39-bus system is shown in Figure 4.2. Four disturbances are considered, summarized in Table 4.2.

Table 4.2: Scenario Description

Disturbance	Description
Disturb 1 (D1)	Fault at bus 3, cleared by tripping line from bus 2 to bus 3 after 0.1s
Disturb 2 (D2)	Fault at bus 16, cleared by tripping line from bus 16 to bus 21 after 0.1s
Disturb 3 (D3)	Fault at bus 17, cleared by tripping line from bus 17 to bus 27 after 0.1s
Disturb 4 (D4)	Fault at bus 26, cleared by tripping line from bus 26 to bus 28 after 0.1s

The four scenarios are grouped into six cases to test both single contingency situations and multi-contingency situations, as in Table 4.3.

Table 4.3: Case Descriptions

Case Studies	
Case A	D1
Case B	D2
Case C	D3
Case D	D4
Case E	D2 + D3
Case F	D1 + D2 + D3

The test was performed in the following environment:

Processor: Intel(R) Core(TM)2 Duo CPU E8500 @ 3.16GHz.

Installed memory (RAM): 4.00 GB (3.87 GB usable).

System Type: 64 bit Windows 7 Enterprise Service Pack 1.

Table 4.4 shows the simulation results.

Table 4.4: TSC-OPF Results on New England 39-Bus System

Generation	Base Case	Case A (D1)	Case B (D2)	Case C (D3)	Case D (D4)	Case E (D2+D3)	Case F (D1+D2+D3)
G1 (MW)	604.47	599.06	694.41	667.93	604.47	694.41	673.53
G2 (MW)	646.00	646.00	646.00	646.00	646.00	646.00	646.00
G3 (MW)	715.41	695.90	725.00	711.14	715.41	725.00	725.00
G4 (MW)	652.00	652.00	652.00	652.00	652.00	652.00	652.00
G5 (MW)	508.00	508.00	508.00	508.00	508.00	508.00	508.00
G6 (MW)	687.00	680.11	649.67	687.00	687.00	649.67	649.47
G7 (MW)	580.00	580.00	580.00	580.00	580.00	580.00	580.00
G8 (MW)	564.00	654.00	564.00	564.00	564.00	564.00	564.00
G9 (MW)	667.79	708.82	747.22	707.00	667.79	747.22	762.07
G10 (MW)	674.44	666.52	538.79	579.91	674.44	538.79	545.39
Total Cost (\$/hr)	<b>41941.3</b>	<b>41965.8</b>	<b>42228.7</b>	<b>42042.4</b>	<b>41941.3</b>	<b>42228.7</b>	<b>42236.1</b>
Cost Adjust.	0	+0.06%	+0.68%	+0.24%	0	+0.68%	0.7%
Time (s)	<b>0.18</b>	<b>0.28</b>	<b>0.23</b>	<b>0.22</b>	<b>0.24</b>	<b>0.40</b>	<b>0.40</b>
Time Adjust.	0	+55%	+27%	+22%	+33%	+122%	+144%
Binding*	N/A	Yes	Yes	Yes	N0	N/A	N/A

\* “binding” shows if the contingency is unstable at dispatch while not considering transient constraints.

There are several comments regarding the numerical results

- Adding more transient stability constraints cannot improve the economic benefits, as observed in the cost (\$/h) from Table 4.1 and Table 4.4. However, the transient stability has been maintained with less than 1% increase of cost. Considering the severity of transient instability, this trade-off is acceptable. The cost increase is caused by extra binding constraints, and this test intentionally selected some severe contingencies (3 of the 4 contingencies are binding) that raise the binding conditions.
- Only the first swing of rotor angles is considered, so the simulation time is

selected to be 2 seconds. Setting the fixed time step to 0.05s will generate about 40 integration steps. Imposing both upper and lower limit, there are about 80 equalities for each generator. For larger systems, the constraints increase with more generators.

- Transient stability constraints of each contingency include about 80 rotor angle inequalities for each generator in each contingency; and the number of constraints for each contingency increases only with more generators. Compared to the optimization time without contingency, there is over 100% time increase when considering 3 contingencies, but there is no significant time increase from the 2-contingency case to the 3-contingency case.

#### **4.8 Summary of TSCOPF**

This chapter has used trajectory sensitivity for TSCOPF to improve the time efficiency. Repetitive simulation for different dispatch is avoided during the problem solving process. TSCOPF-TS forms a two-step algorithm. Analysis shows that computational burden brought by transient stability constraints is far less than that from steady state security constraints. Tests on a 9-bus system and the New England 39-bus system show that the performance of TSCOPF-TS for multi-contingency dispatch is satisfactory.

The simulations described in this section were run using Matlab. We are currently coding a C++ version of the optimization software which is expected to run significantly faster than the Matlab code.

## **Part IV**

# **Dynamic Security Assessment Processing System and Application to Low Probability Events**

**James McCalley, Project Leader**

**Lei Tang, Graduate Student**

**Siddhartha Kumar Khaitan, Research Associate**

**Iowa State University**

## 5 Addressing Low Probability Events

---

### 5.1 Motivation

For the past several decades, power systems have experienced more frequent and more severe high consequence events, resulting in large blackouts. The severe impact is manifested in increased economic loss, electric power interrupted and number of people affected. Blackouts are typical low probability high consequence events. This raises the question of how to address those low probability high consequence events that cause high risk in power system operations.

We have carefully reviewed the major blackouts that have occurred since 1965. Table 5.1 lists the number of notable major blackouts around the world [1]. The definition of “notable” means

- The blackout must not be planned by the service provider
- The blackout must affect at least 1000 people and last at least one hour
- There must be at least 1 million (person  $\times$  customer hours) of disruption

We observe from Table 5.1 that, since 1965, the number of major blackouts increases each year.

Table 5.1: Number of Notable Major Blackouts Around the World Since 1965

Year	1965- 1995 (30 years)	1996- 2000 (5 years)	2001- 2005 (5 years)	2006- 2010 (5 years)	2011 (1 year)
Number of Major Blackouts	13	16	27	77	18

In Table 5.2 and Table 5.3, we have summarized the 30 most severe blackouts that have occurred since 1965. These tables were updated from similar ones given in [4, 33].

Table 5.2: Summary of Extreme Major Blackouts Since 1965

Location	Date	GW lost	Duration	People affected	Approximate cost
US-NE	11/09/1965	20	13 hours	30 million	
US-NE	07/13/1977	6	22 hours	3 million	300 million
France	12/19/1978	30	10 hours		
West Coast	12/22/1978	12.35		5 million	
Sweden	12/22/1982	>7	5.5 hours	4.5 million	
Brazil	12/27/1983	15.762			
Brazil	08/18/1985	7.793			
Hydro Quebec	04/18/1988	18.5			
US-West	01/17/1994	7.5			
Brazil	12/13/1994	8.63			
US-West	12/14/1994	9.336		1.5 million	
Brazil	03/26/1996	5.746			
US-West	07/02/1996	11.743		1.5 million	
US-West	07/03/1996	1.2		small number	
US-West	08/10/1996	30.489		7.5 million	1 billion dollars
San Francisco	12/08/1998	1.2	8 hours	1 million	
Brazil	03/11/1999	25	4 hours	75 million	
Brazil	05/06/1999	2			
India	01/01/2001	12	13 hours	220 million	107 million
Rome	06/26/2003	2.15		7.3 million	
US-NE	08/14/2003	62	1-2days	50 million	4-6 billion
Denmark/Sweden	09/13/2003	6300	6.5 hours	5 million	
Italy	09/28/2003	27	19.5 hours	57 million	
Croatia	12/01/2003	1.27			2.5 million
Greece	07/12/2004	9	3 hours	5 million	
Moscow/Russia	05/24-25/2005	2.5	> 6 hours	4 million	
Java/Bali	08/18/2005	2.7		100 million	
European Blackout	11/04/2006	6.4	1 hour	15 million	
Brazil/Paraguay	11/10/2009	1.4	4.5 hour	87 million	

Table 5.3: Analysis of Extreme Major Blackouts Since 1965

<b>Location</b>	<b>Date</b>	<b>Number of outaged elements</b>	<b>Time between initiating and secondary, pre- collapse events</b>	<b>Cause of secondary, pre-collapse Events</b>
US-NE	11/09/1965	N-1	Few minutes	Proper protection operation
US-NE	07/13/1977	N-2	20-45 minutes	Lightening
France	12/19/1978		> 30 minutes	Proper protection operation
West Coast	12/22/1978	N-1	Fast	Primary and secondary protection & communication failure
Sweden	12/22/1982	N-2	50 seconds	Proper protection, under frequency LS failure
Brazil	12/27/1983	N-1	9-10 minutes	Simultaneous tripping of 7 circuits and transformers
Brazil	08/18/1985	N-2		Protection failure (SPS setting)
Hydro Quebec	04/18/1988	N-3	2-3 seconds	Communication failure followed by load shedding protection failure
US-West	01/17/1994	N-2	Fast	Proper protection operation
Brazil	12/13/1994	N-1		Proper protection operation
US-West	12/14/1994	N-1	40-52 seconds	Inefficient protection, loss of synchronism
Brazil	03/26/1996	N-1		Proper protection operation
US-West	07/02/1996	N-1	20 seconds	Proper protection operation, relay misoperation
US-West	07/03/1996	N-1	Fast	Relay misoperation
US-West	08/10/1996	N-1	5-7 minutes	Protection failure
San Francisco	12/08/1998		16 seconds	No load protection, delayed remote protection
Brazil	03/11/1999	> N-6	> 30 seconds	Proper protection operation

Table 5.3: Analysis of Extreme Major Blackouts Since 1965 (Continued)

Brazil	05/06/1999	Many		Inadvertent protection operation
India	01/01/2001		13 hours	
Rome	06/26/2003			High load, low generation reduction in import
US-NE	08/14/2003	N-1	More than 2 hours	Proper protection operation
Denmark/Sweden	09/13/2003	N-1	5 minutes	Switching device breaks, proper protection operation
Italy	09/28/2003	N-1	25 minutes	Unsuccessful reclosing, loss of synchronism, dynamic interaction leading to voltage collapse
Croatia	12/01/2003	N-1	30 seconds	Protection failure
Greece	07/12/2004	N-1	10 minutes	Proper protection operation
Moscow/Russia	05/24-25/2005		> 12 hours	6 lines from HV substation tripped due to faults and overloading
Java/Bali	08/18/2005			
European Blackout	11/04/2006	Many	30 minutes	Proper protection operation
Brazil/Paraguay	11/10/2009			87 million

The following observations are based on Table 5.3.

- Initiating events can be N-1 contingencies and also N-k ( $k \geq 2$ ) contingencies. N-k contingencies are usually line or bus faults followed by nearby protection failures.
- About 50% of the blackouts were fast; taking only several seconds to collapse after the initial event occurred. The remaining 50% of the blackouts involved slow processes that took several minutes to several hours to collapse after the initial event occurred.
- If the blackout was one of the 50% characterized as “slow,” it involved cascading outages where transmission circuits and generators were tripped one after another which finally led to collapse due to the weakened system topology.

Reference [27] proposed a framework that used dynamic event trees to store potential cascading sequences and design corresponding remedial actions. An extended term time domain simulator was used to simulate offline most low probability high consequence contingencies as potential trigger incidents, based on the prediction of the upcoming operating scenarios in the next many hours or days. Initiating event identification and probability calculation has been well addressed in references [2]. Work in this report will extend this work and also that of [4] in developing the ability within a dynamic security assessment processing system (DSAPS) to address low probability events.

## 5.2 Overview

Our approach within DSAPS to handle low probability events addresses them via two modes: anticipatory mode and emergency mode. Both modes utilize a high-speed extended-term time domain simulator (HSET-TDS) that models fast (e.g., machine, exciter, stabilizer, and governor) slow (e.g., boilers, tap changers, load ramping, automatic generation control) dynamics as well as generator protection systems.

The two modes are illustrated in Figure 5.1 [34] and described in the following two subsections.

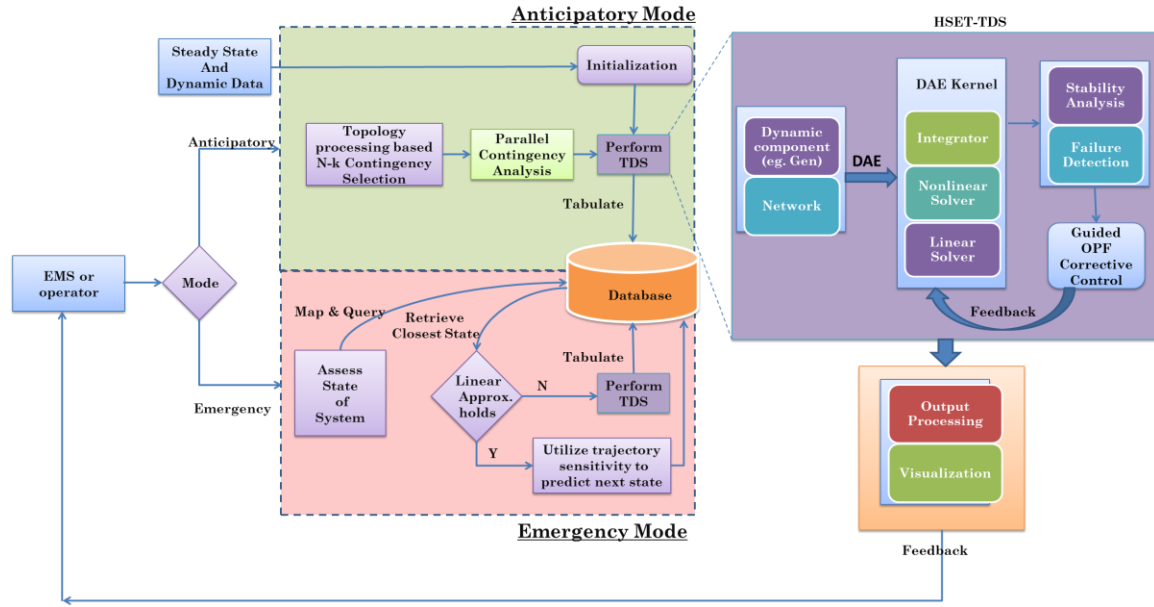


Figure 5.1: Handling Low Probability Events within DSAPS

### 5.2.1 Anticipatory Mode

In the anticipatory mode, DSAPS continuously assesses low probability initiating events in order of decreasing probability, computing system response for each event. The goal of anticipatory computing is to cover as much of the event-probability space as possible within a particular window of computing time. A failure detection facility checks simulation results as the simulation is progressing and stops it when unacceptable system performance (failure) is recognized. If failure is detected, an optimal corrective action is identified by a guided optimal power flow (GOPF) where embedded intelligence selects the type of corrective action, based on the identified failure conditions, which become the decision variables within the optimal power flow (OPF). The amount of corrective action is then determined by the OPF.

Results of this process are archived in a database. These results include the initiating event, the initial operating conditions, the failure, the corrective action, and the amount of corrective action. Should a low-probability event occur in the system, the database is searched to find the event, and then the operating conditions used in the simulation are

compared to the operating conditions when the event occurred. If the event is found, and if the simulated operating conditions are near to those of the actual operating conditions, then the corrective action is retrieved and displayed to the operational personnel for use as decision-support in responding to the event.

The basic philosophy of the anticipatory mode is to prepare and revise, track and defend.

The failure detection and corrective action determination is an automation of certain aspects of procedures used to design special protection systems (also known as remedial action schemes). The anticipatory mode is illustrated in

Figure 5.2.

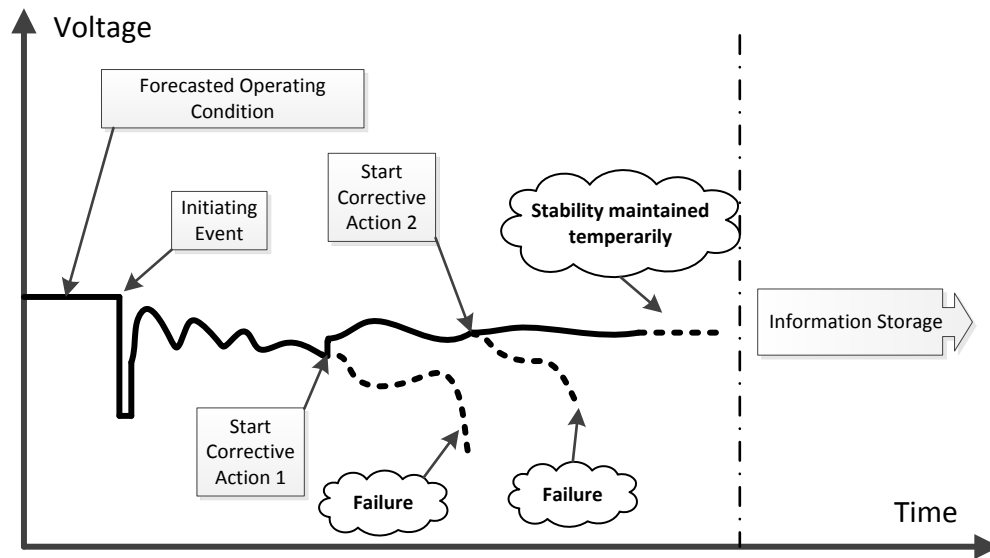


Figure 5.2: Illustration of Anticipatory Mode

### 5.2.2 Emergency Mode

DSAPS is available in an emergency mode where it is run following initiation of a severe disturbance. In this mode, DSAPS computes corrective control actions based on direct integration or previously computed linear trajectory sensitivities if the current state of the system is within the region of accuracy of the closest retrieved system state. In case it performs direct integration, the results are archived. At the same time the emergency mode is initiated, DSAPS searches the archives for the low probability event that occurred. If that event is found, and if the corresponding archived operating conditions are within the region of accuracy of the actual pre-event operating conditions, the corresponding archived corrective actions can be provided to the operational personnel for use as decision support. If the event is not found, or if the corresponding archived operating conditions are not within the region of accuracy of the actual pre-event operating conditions, then corrective actions for use as decision support are obtained when the emergency mode simulation is completed, at which time they are presented to

the operational personnel. Thus, emergency mode simulation must be faster than real time to be useful.

## 6 Design and Operation of a High Speed Extended-Term Time Domain Power System Simulator (HSET-TDS)

---

### 6.1 Introduction

Modern power systems have greatly grown in their sizes to cater to the increasing demands of electricity. Hence, modeling these power systems for getting insights into their characteristics has become a real challenge. Nonetheless, modeling of power systems is extremely important, since this enables the designers to gain insights into their functioning and propose innovations for continuously improving them. In the absence of efficient simulators, many design ideas (e.g., stability analysis) for improving power systems of tomorrow are being tested using simulators which run the power systems of yesterday. The slow simulation speed also forces the developers to perform simulations in an offline manner. Thus, the inherent computation inefficiency of the existing simulators prohibits the researchers from modeling the power systems in full detail. To model highly detailed power systems of today, the simulators should be easily extensible and provide flexibility to achieve a fine balance between the modeling accuracy and simulation speed. However, many of the existing simulators do not offer this flexibility. Thus, lack of high speed simulators which also provide detailed modeling, has remained a critical bottleneck in realistic validation of the design ideas.

The Dynamic Security Assessment Processing System (DSAPS) described previously requires very fast simulation capability. To address this, in this chapter we describe HSET-TDS, a high speed power system simulator.

Design of differential-algebraic-equation (DAE) numerical solvers requires designer decision at several levels including hardware, programming language, integration method, nonlinear solver and linear solver. We view these decisions hierarchically as illustrated in Figure 6.1, with broader, structural design decisions represented at the top. We desire to identify the combination of choices for these five hierarchical levels so as to achieve the most effective computational efficiency for power system time domain simulation. The description of HSET-TDS in this chapter is one of several evolving steps to achieve this.

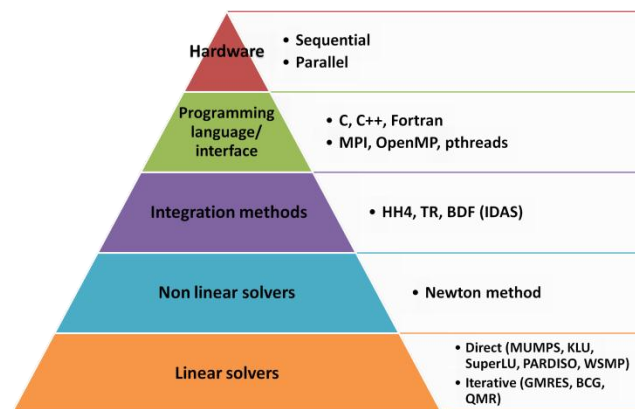


Figure 6.1: Synergistic View of Power System Time Domain Simulator Design

To help the user explore the design options and also exercise trade-off between required modeling accuracy and simulation speed, HSET-TDS provides different models of power system components such as governors, exciters, generators and numerical algorithms, namely linear, nonlinear and integration solvers. HSET-TDS provides a cross-platform simulation framework, and has been designed using the well-known object-oriented programming (OOP) approach. HSET-TDS utilizes the structural correspondence between the physical components (e.g., generators, buses, etc.) and the classes of the simulation model to enable translation of real-world relationships between the components into the simulation framework. Thus, the simulator code closely mimics the structure of the power system domain. By virtue of this feature, the simulator can be developed and more simulator functionalities can be easily added. For data plotting and visualization, HSET-TDS provides output in a form which is compatible with most plotting programs such as MATLAB, gnuplot and matplotlib (python).

We discuss several design features of HSET-TDS which help in providing high simulation speed while also enabling detailed modeling of power systems. HSET-TDS has been validated against commercial software packages, namely PSSE and DSA Tools. Further, using the best among different classes of numerical algorithms, HSET-TDS outperforms PSSE software in computational efficiency.

We are currently working on HSET-TDS to scale it to analyze a large number of contingencies. We plan to develop it into a parallel simulator with capabilities of both functional and domain decomposition based parallelization. Using this, HSET-TDS would be ported to state-of-the-art HPC (high performance computing) platforms. Future versions of HSET-TDS would also allow extended-term simulations.

## 6.2 Literature Review

Simulation is an important research tool and is used extensively for evaluation and verification of proposed design ideas. For purpose of study, we broadly divide the power system simulators in two categories, namely commercial software and research-grade software. Among the commercial software, Power World [42], Digsilent Power Factory [43], ETAP (Electrical Transient Analyzer Program) [44], EuroStag [45], PSSE [46], Simpow [47] and CYME [48], etc., are well-known simulators. The examples of research/educational software are PSAT [49], EST [50], UWPFLOW [51], VST [52]. Although the commercial simulators model the power system in much more detail than the research-level simulators; due to their being closed source, they do not provide full flexibility of experimentation and prototyping.

PSAT is a Matlab toolbox, which stands for power system analysis toolbox. PSAT can perform continuation power flow and optimal power flow analysis. Further, it can also perform small signal stability analysis and time domain simulations. It uses Matlab for optimizing its performance and Simulink for providing graphical representation. TRELSS (Transmission Reliability Evaluation of Large-Scale Systems) is a simulator used for power network reliability analysis using enumeration of generation and transmission contingencies [53, 54]. TRELSS has been used to analyze and simulate cascading failures. TRELSS simulates cascading outages of lines, transformers and generators due to overloads and voltage violations in large AC networks. TRELSS models islanding and

protection control groups. Further, it also ranks the cascading outages in order of severity. PSSE (Power System Simulator for Engineering) is a tool for simulating and analyzing power system performance. It provides capabilities to analyze power flow and conduct extended term dynamic simulation, network reduction, transfer limit analysis, etc.

UWPFLOW is a tool which calculates local bifurcations related to system limits or singularities in the system Jacobian in power system modeling. VST [52] is a Matlab-based voltage stability toolbox. It is designed to analyze bifurcation and voltage stability in power systems. VST is built upon the theoretical foundation of bifurcation theory and provides flexibility to model load flow, analyze small-signal and transient stability. In addition, NEPLAN [55], PCFLO [56], COMREL [57] are other tools which are used in power system simulations.

Also, several power system toolboxes are based on Matlab, such as MatPower (Matlab Power System Simulation Package) [58], MatEMTP (Electromagnetic Transients Program in Matlab) [59], PST (power system toolbox') [60], PAT (Power Analysis Toolbox for Matlab/Simulink) [61], SPS (SimPowerSystems) [62], EST (Educational Simulation Tool), etc.

HSET-TDS is research software and is developed in C++. This is in contrast with most other simulators/toolboxes, which are based on Matlab. We discuss the computational efficiency of C++ and Matlab later.

### 6.3 Application and Architecture View

Figure 5.1 shows the overall architecture of the simulator. HSET-TDS operates in two modes, namely the anticipatory mode and the emergency mode. In normal scenarios, HSET-TDS operates in anticipatory mode, where it analyzes many contingencies and stores the results or suggested corrective actions in a database. Through anticipatory computing, HSET-TDS tries to cover as much of the event probability space as possible within a given time duration.

In the case of a severe disturbance, HSET-TDS switches to emergency mode. In this mode, HSET-TDS queries the database. If the current state of the system is within the region of accuracy of the closest retrieved system state, then the corrective control action for the disturbance can be computed by the result retrieved from the table. Otherwise, HSET-TDS analyzes the contingency, computes the corrective action and archives the result.

HSET-TDS is capable of simulating a wide variety of systems. HSET-TDS has been tested with many different test systems of different sizes ranging from 9 bus system to 13029 bus systems with more than 2000 generators. HSET-TDS has been compared against commercial software for both computational efficiency and accuracy.

HSET-TDS can be used for a variety of studies such as stability analysis, finding critical clearing time of contingencies, providing situational awareness for operators. In energy management systems, HSET-TDS can be used as a decision support tool for the operational personnel in the event of high-consequence disturbances. HSET-TDS can help the power system operator in dynamic security assessment by predicting both possibility of failures and designing appropriate corrective actions for them.

## **6.4 Modeling and Design View**

### **6.4.1 Mathematical Modeling**

A set of differential equations describes the dynamic behavior of the power system components which includes the generator, exciter, governor, and dynamic loads like induction motor. The differential equations of generator, exciter and governor and other dynamic components like induction motor, power system stabilizer together form differential equations. Similarly all the algebraic equations of the network, the generator and the governor form the algebraic equations of the power system. The differential and algebraic equations together (DAE) define the electromechanical state of power system at any instant of time. A system of differential equations are referred to be a stiff system if they have both slow varying and fast dynamics present in them. If the generator dynamic models include the sub transient equations (fast varying components) and slow rotor dynamics, then the DAE of the power system becomes stiff. The integration method which is used to solve the DAE of the power system should be capable of handling stiff systems.

### **6.4.2 Strategy**

The effectiveness of solving the DAE of the power system depends on the numerical algorithms and the type of generator models used. The DAE of the power system can be solved either by the partitioned solution strategy (also called alternating solution strategy) or the simultaneous solution approach. In the partitioned solution approach or alternating solution strategy, for a given time step, both the differential equations and the algebraic equations are solved separately. The solution of the differential equations  $x$ , is used as a fixed input while solving the algebraic equations and similarly the solution of the algebraic equations  $y$ , is used as a fixed input while solving the differential equations. This process is repeated iteratively until the solution converges. In the case of simultaneous or combined solution approach, the differential equations are discretized using an integration method to form a set of nonlinear equations which are then combined with the algebraic equations to form a single set of nonlinear equations and solved by Newton method as discussed below.

The partitioned solution approach is suited for simulations that cover a shorter time interval. This method allows a number of simplifications like partial matrix inversion, triangular factorization to speed up the solution. However, if proper care is not taken in handling these simplifications it will lead to large interfacing errors. In the case of simultaneous approach, Newton's method is usually employed to solve the set of algebraic equations. It ensures no convergence problem even if large integration step length is used with stiff systems like power system. Since both differential and algebraic equations are solved together; there are no interfacing problems. This method also allows for large and variable time steps which is not feasible for partitioned strategy. In this research simultaneous approach is adopted which is suitable for both short-term and long-term dynamic simulations.

Power system simulators must be capable of modeling both fast and slow dynamics and adaptively change time steps of integration since the power system dynamic response to

disturbances is decided not only by fast dynamics of its machines, but also by the action of slow processes such as tap changers and load dynamics. To capture the multi-scale dynamics, the components could be modeled to different degrees of detail according to the phenomenon studied. As the complexity of the models increases, the dimension of the DAEs and the computation resources required to solve them also increase. Therefore, it is a common practice to perform model reduction to gain computational efficiency. However, model reduction also leads to loss of accuracy and very fast transients. Therefore, power system simulator design involves achieving a fine balance between modeling accuracy and simulation efficiency. This also requires very efficient numerical algorithms for enabling near real-time simulations.

To fulfill the modeling needs to different degrees of accuracy, HSET-TDS provides a library of generator models (2<sup>nd</sup>, 3<sup>rd</sup>, 4<sup>th</sup>, 6<sup>th</sup> and GENROU) exciter models (IEEET1 and IEEEEX1) and governor (TGOV1, GAST and HYGOV) models [63]. HSET-TDS uses an integration algorithm to solve the differential equations. The nonlinear equations are solved using Newton methods. For solution of linear equations, which form a major part of the solution time, efficient sparse linear solver routines are used. In the following, we discuss the numerical algorithms in more detail.

#### **6.4.3 Integration Method**

There have been several efforts to develop robust and fast numerical integration methods to deal with stiffness and ensure A-stability [64-68]. Different numerical integration schemes differ in their convergence, order, stability, and other properties. In HSET-TDS we have implemented several different explicit and implicit integrators such as backward differentiation formula (BDF), Trapezoidal, Euler, etc. These integrators can be operated in fixed time step mode or variable time step mode. By varying the minimum and maximum step size of integration, the user/developer can exercise a trade-off between accuracy and simulation speed. The time steps are chosen in a manner to minimize the local truncation error. The goal is to choose largest time step meeting the error condition. Optional routines are provided to find the consistent initial values. HSET-TDS provides the flexibility to study voltage, angle or speed response at a desired precision.

#### **6.4.4 Nonlinear Solver**

The nonlinear equations, formed as a result of differential equations discretization and algebraic equations of the network and the components, are solved iteratively by the Newton method at each time step. Newton methods have quadratic convergence. However it is a common practice to keep the Jacobian constant over a large number of time steps to alleviate the need to refactorize it at every time step. These are called modified or inexact Newton methods. These maintain fast convergence rate. Inexact Newton methods with line search allows larger initial time step and allows greater robustness and flexibility.

#### **6.4.5 Linear Solvers**

There have been significant advances in the field of sparse linear solvers. Linear solvers can be categorized into direct linear solvers and the iterative linear solvers. There are a

number of algorithms available for both the direct and the iterative solvers. However not all algorithms are suitable for all applications like in power system computations. The choice of the algorithm depends on the problem at hand and the numerical characteristics of the Jacobian involved. Some of the characteristics that impact the choice of the methods are a) diagonal dominance (b) numerical stability (c) symmetry (d) conditioning (e) structure (banded, tridiagonal, etc.) and so on. In this research we focus on the direct linear solvers. HSET-TDS allows an easy interface to any sparse linear solver routine. Currently HSET-TDS is interfaced only with the direct linear solvers, namely UMFPACK, KLU and PARDISO [40, 41, 69]. Section 6.7 compares the computational efficiency of HSET-TDS for these linear solvers.

In contrast to a dense solver algorithm, which involves two phases, a typical sparse solver consists of four distinct phases, which are as follows.

- a. The ordering step which minimizes the fill-in and exploits special structures (for example block triangular form).
- b. The analysis step or symbolic factorization which determines the nonzero structures of the factors and creates suitable data structures for the factors.
- c. Numerical factorization for computing the factor matrices.
- d. The solve step which performs forward and/or backward substitutions.

We next present a brief overview of the sparse direct solvers which are implemented in HSET-TDS (KLU, UMFPACK and PARDISO).

KLU is a sparse linear solver package written in C for solving sparse unsymmetric linear systems of equations. KLU was originally designed for circuit simulation applications. KLU is found to be the most efficient linear solvers for power system dynamic simulations on the serial platform. UMFPACK consists of a set of ANSI/ISO C routines for solving unsymmetric sparse linear systems using the multifrontal method [70]. UMFPACK offers an easy interface to different programming languages. HSET-TDS uses UMFPACK version 5.6.0. The PARDISO package is an easy to use software for solving large sparse symmetric and unsymmetric linear systems of equations in both serial and parallel platforms. PARDISO algorithm utilizes Level-3 BLAS update for performance. Non symmetric permutations and preprocessing speed up the factorization process in PARDISO.

#### **6.4.6 Simulation Modeling**

HSET-TDS is a cross-platform simulator which can be used with both Windows and Linux operating systems. The design of HSET-TDS has been done using object-oriented programming (OOP) approach and modeling language chosen is C++. With object-oriented programming approach, each power system component is modeled as an object, which contains data fields, methods and also specifies their interactions. For each component, the corresponding class definition specifies the properties, attributes and behaviors of the object. The attributes of the component define all the properties of the object and the behaviors determine the manner in which components interact with other components or conduct different functions. HSET-TDS uses a driver class which binds different components together. This class acts as a user-interface for the developer. Thus,

by the virtue of using OOP, HSET-TDS maintains a direct one-to-one correspondence between components of the power system domain and classes of the simulation model.

HSET-TDS leverages several other features of OOP paradigm also. A feature of OOP is that it allows composition, which means that new classes can be designed by composing the existing classes. As an example, Figure 6.2 presents a conceptual illustration of a power system.

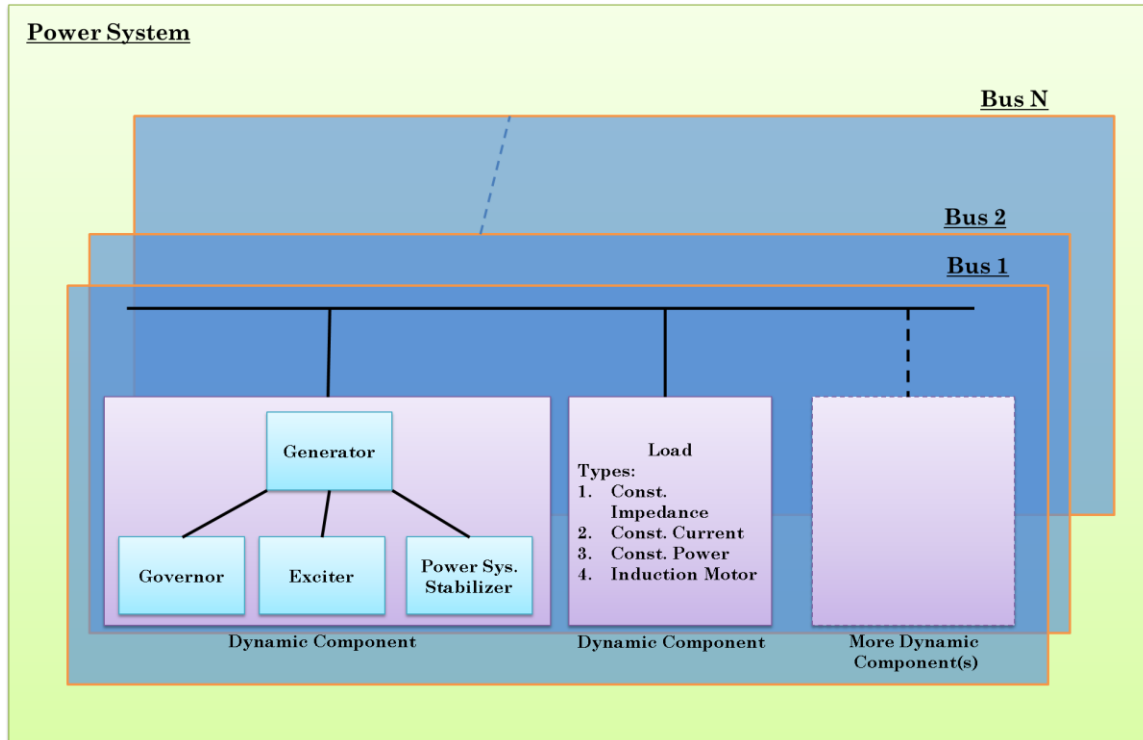


Figure 6.2: Conceptual Illustration of the Power System

The power system is composed of different dynamic components. The equivalent modeling of the power system using OOP approach is shown in Table 6.1. The BusModel is composed of variables from different classes such as generator node, exciter node, governor node, etc. Further, the BusModel itself can be part of another class such as PowerSystem which models the entire power system. This hierarchical structure allows easy and accurate modeling.

Table 6.1: Example Composition in HSET-TDS

```
class BusModel
{
    GeneratorNode genNode;
    ExciterNode excNode;
    GovernorNode govNode;
    LoadNode loadNode;
    BreakerNode breakerNode;
    BranchNode branchNode;
};
```

HSET-TDS is modular, and hence developments in one component affect other the simulation model of other components in only minimal way. This is because different components interact with each other through well-defined interfaces ("methods") and not through direct data reference. Thus, a change in the internal modeling of a component does not change its interface to the external environment.

The inheritance feature of OOP allows code-reuse for development of new classes by inheriting the functionalities of the existing classes. HSET-TDS uses inheritance to aggregate the common features of different components in a base class. Further, by using the base class to derive new classes, the efforts required for developing the common code is avoided and the new classes can contain the specific implementation of the methods or also have extra features.

Another feature of OOP is that it promotes "encapsulation". This implies that all the properties of the component (class) are set within the definition of the class itself and is not externally defined. Thus, the any need to understand or revise the properties of the component is located in the single place. Also remaining components (classes) can be modeled without being concerned with the internal design of the given component. Furthermore, encapsulation facilitates easy creation of multiple instances of the same component, since each class contains all of the required functionalities of the object. Thus, simulation modeling becomes easier. In HSET-TDS, interaction among multiple components happens using "message passing" feature of OOP. The communication between objects takes place through well-defined function interfaces.

HSET-TDS uses industry standard data input. Due to this, HSET-TDS does not require pre-processing of power system specification (raw) files. Inside the simulation model, HSET-TDS allocates components using dynamic memory allocation (C++ new command) and not static memory allocation. Thus, HSET-TDS can simulate power systems with any number of components without requiring tuning of parameters, etc.

To provide robust interface to the user, HSET-TDS uses an InputRecord class. This class ensures that the simulation parameters provided by the user are valid. Further, it also sets meaningful default parameters when such information is not provided.

HSET-TDS uses an event-driven simulation approach to model different events happening in the power system. HSET-TDS maintains a single global event queue, which is sorted by the time of the occurrence of the event. At each simulation time increase, this queue is checked to see for the occurrence of the events. An event describes its time period of occurrence, the component(s) which it affects and its characteristics. Using this approach, either occurrence or clearing of faults in single or multiple components can be easily modeled. For this, HSET-TDS models each fault as an event. For example, a fault in a component is modeled as an event occurring at a given time in that component. Corresponding to each event, the event handling mechanism in HSET-TDS takes suitable actions. The events in HSET-TDS could be branch fault, branch fault clearing, bus fault, bus fault clearing, branch trip, generator trip or a combination of these.

In summary, HSET-TDS models the components and other non-physical aspects (e.g., constraints, behaviors, etc.) of the power system using well-established design philosophy of OOP. Thus, HSET-TDS makes the task of extending the simulator and adding new features easy and also provides significant benefits.

## **6.5 HSET-TDS Efficiency**

The time evolution of the power system is modeled by the DAEs. US has three major interconnections namely the Eastern, Western and the ERCOT. Each of them have large number of electric power components. As an example, the Eastern Interconnection uses power system with around 30,000 buses and 5,000 generators. Assuming 25<sup>th</sup> order differential equations for modeling each of the generator, governor and exciter combinations and other associated controls like PSS, voltage regulators, the dynamic simulation model will have more than hundred thousand differential algebraic equations. Further, with increased emphasis on simulating N-*k* contingencies, the number of contingencies that need to be simulated and analyzed are extremely large. Thus, computational efficiency is the major bottleneck. HSET-TDS achieves efficiency in two aspects, namely by choice of numerical solvers and by choice of simulation platforms. We now discuss each of them in more detail.

### **6.5.1 Algorithmic Efficiency**

HSET-TDS allows easy interface with many different sparse linear solvers, and by selecting the best among them the computational efficiency can be significantly enhanced. HSET-TDS uses variable time step integrator to gain further computational efficiency. As for linear solver, HSET-TDS uses KLU which is nearly twice as fast as other linear solvers like PARDISO and UMFPACK.

### **6.5.2 Simulation Platform Efficiency**

C++ is a compiled language, while Matlab is an interpreted language. Thus, for typical programs interpreted languages are slower than the compiled languages. Therefore, due

to the inherent speed limitations of Matlab, the Matlab-toolbox based programs (power simulators) provide limited simulation speed. On the other hand, since HSET-TDS uses C++, it can take advantage of several optimizations and debugging features provided by the compiler. Visualization in HSET-TDS is achieved through exporting the data in tabular format which can be easily rendered through plotting software such as SigmaPlot, Matlab, and gnuplot. The decoupling of the simulator and the visualization tool ensures that the requirements of plotting do not affect the simulation efficiency.

## 6.6 HSET-TDS Validation Results

To establish the accuracy and estimate the modeling error and error in solution strategy HSET-TDS is compared against commonly available commercial software packages namely DSA Tools (TSAT) and PSSE.

We a large power system test case is selected which has 13,029 buses, 431 generators, 12,488 branches and 5,950 loads. In the simulation, each generator (GENROU) is provided with an exciter (IEEET1) and a governor (TGOV1) model. These are the standard PSSE models and the details of them could be found in PSSE manuals. The contingency selected has a bus fault at bus 13179 starting as time 1.1 seconds is cleared at 1.2 seconds. Following this there is a branch fault on 13181-13182, at time 5.5 seconds and it is cleared at time 5.7 seconds. The simulation is performed for 20 seconds.

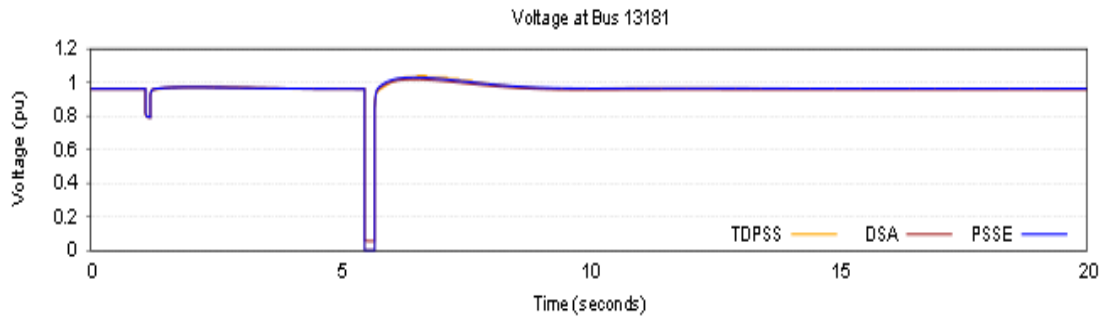


Figure 6.3: Voltage Profile

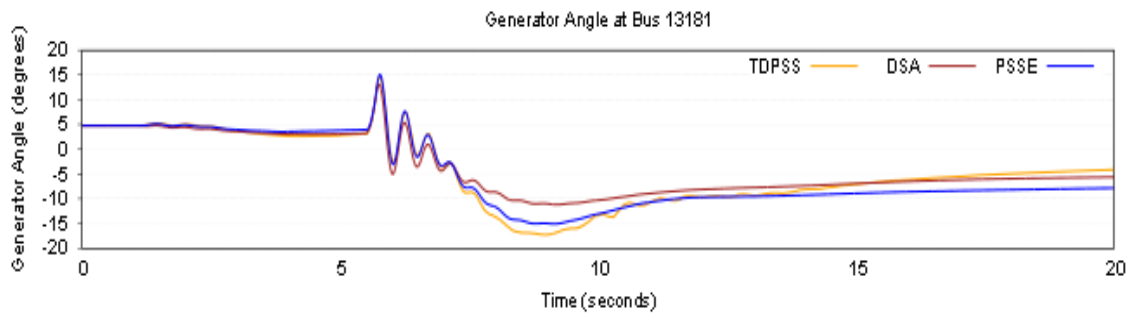


Figure 6.4: Generator Angle Profile

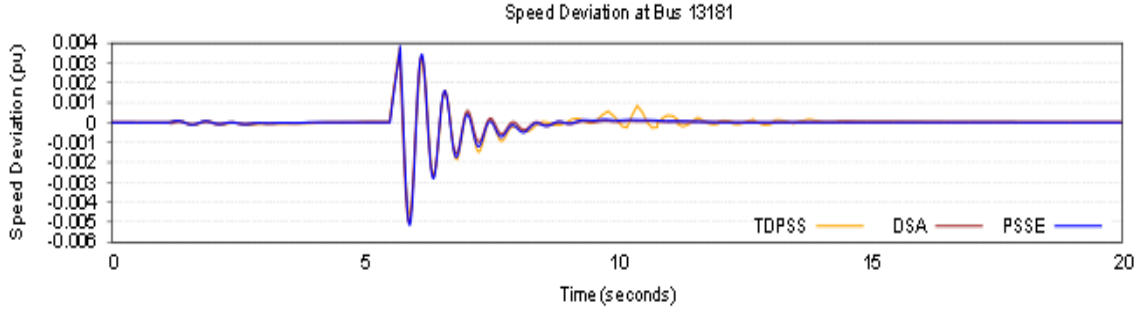


Figure 6.5: Speed Deviation Profile

Figure 6.3 - Figure 6.5 show the simulation plots for bus voltage, generator angle and speed deviation on bus 13181 using HSET-TDS, PSSE and DSA tools. The response obtained using HSET-TDS matches closely with those obtained using commercial software packages like PSSE and TSAT. These results confirm the accuracy of HSET-TDS. From the simulation plots it is important to note that there are slight differences in the responses obtained from PSSE and TSAT using the same models and input file for the model description.

## 6.7 Performance Comparison

Linear solvers play an important role in the solution of the DAEs of the power system. The linear system solution sub problem forms a large part in the solution process and thus has an important role in computational efficiency. In this section, we compare the simulation time taken by HSET-TDS for different linear solvers on two different contingencies. The first contingency is a bus fault at bus 13179 starting as time 1.1 seconds is cleared at 1.2 seconds. Following this there is a branch fault on 13181-13182, at time 5.5 seconds and it is cleared at time 5.7 seconds. The second contingency has a bus fault at bus number 26 (which is a generator bus) at time 1.1 seconds, which is cleared at 1.2 seconds. The simulation is run for 20 seconds in both cases.

The simulation time by HSET-TDS using different linear solvers for the two contingencies are shown in Table 6.2

Table 6.2: Speed Comparison with Different Linear Solvers

	Simulation Times(seconds)			Speedup	
	KLU	UMFPACK	PARDISO	UMFPACK/KLU	PARDISO/KLU
1	19.5	46.5	34	2.4	1.7
2	11.3	23.7	21	2.1	1.9

KLU outperform the other two linear solvers and compared to UMFPACK, KLU is almost twice as fast as UMFPACK and compared to PARDISO, KLU provides over 1.7 times speedup.

Table 6.3: Speed Comparison with PSSE

	Simulation Times (seconds)		Speedup
	TDPSS	PSSE	PSSE/TDPSS
1	19.5	39	2
2	11.3	38.6	3.4

HSET-TDS is also compared against PSSE for the above mentioned contingencies. The results are shown in Table 6.3. As shown in the results HSET-TDS can provide over 3 times speedup over PSSE. This confirms the prospects of HSET-TDS as an enabling technology in power system dynamic simulation study. Further HSET-TDS can be easily extended to state-of-the-art HPC platforms as opposed to Matlab based or legacy code based simulation software.

## 6.8 Further Development

HSET-TDS is being further developed to improve the simulation speed and enabling more detailed modeling. Towards simulation speed enhancements, we are motivated by Amdahl's law [71] which dictates that efforts for speedup should focus on the component which is most heavily used, since that will most significantly improve the overall performance of the simulator. Based on this, we are working on utilizing parallel linear solvers such as use of multithreading in linear solvers (e.g., PARDISO), MPI based linear solvers (e.g., MUMPS, Superglue) and developing sparse linear solvers for solving ill-conditioned matrices on graphics hardware (GPUs). We are working towards parallelizing contingency analysis since they are fairly independent of each other. This requires development of novel load balancing techniques for maximum resource utilization since simulation times vary greatly.

Computational efficiency: In terms of mathematical modeling improvements we are working to develop efficient algorithms to reduce the stiffness of the DAE of the power system to gain computational efficiency. This can also be viewed in terms of developing effective model reduction strategies. In terms of numerical algorithms employed in power system, we are working towards algorithmic developments of adaptive time stepping and adaptive integration schemes to simulate multi scale and multi component power system. In another funded research project we are further developing IDAS [12], which is a linear multistep variable order variable time step (VOVTS) BDF integrator. Parallel linear solvers to facilitate functional decomposition will be implemented. The software is also intended to perform multi contingency analysis in parallel on different HPC platforms with fine load balancing strategies.

Component modeling: Regarding development activity in detailed modeling, we are working towards extending HSET-TDS capabilities to perform long term simulation through the modeling of automatic generation control, boilers, tap changers and load ramping. Future efforts would also be directed towards energy storage modeling.

Protection system modeling: Of particular importance in this long term simulation study is the role of generator protection especially under abnormal conditions. Along with coordinating with the system to avoid misoperation the protective relays must protect the

generating plant damage which would otherwise seriously hamper restoration and cause huge economic loss. Therefore our future extensions would build protections logic and its operation into the software.

Corrective action determination: Intelligence would be built into the software to selected guided corrective action strategy in the event of a contingency to avert an impending system instability.

## References

---

- [1] K. Morison, W. Lei, and P. Kundur, "Power system security assessment," *Power and Energy Magazine, IEEE*, vol. 2, pp. 30-39, 2004.
- [2] Q. Chen and J. D. McCalley, "Identifying high risk N-k contingencies for online security assessment," *Power Systems, IEEE Transactions on*, vol. 20, pp. 823-834, 2005.
- [3] Q. Chen, "The probability, identification, and prevention of rare events in power systems " PhD, Iowa State University, Ames, 2004.
- [4] S. Khaitan, "On-line cascading event tracking and avoidance decision support tool " PhD, Iowa State University, Ames, 2008.
- [5] R. C. Green, L. Wang, M. Alam, and C. Singh, "Intelligent and parallel state space pruning for power system reliability analysis using MPI on a multicore platform," 2011, pp. 1-8.
- [6] R. C. Green, L. Wang, and M. Alam, "High performance computing for electric power systems: Applications and trends," 2011, pp. 1-8.
- [7] V. Jalili-Marandi, Z. Zhou, and V. Dinavahi, "Large-Scale Transient Stability Simulation of Electrical Power Systems on Parallel GPUs," *Parallel and Distributed Systems, IEEE Transactions on*, pp. 1-1, 2011.
- [8] Y. Zhang, "Par-psse: Software for pairwise statistical significance estimation in parallel for local sequence alignment," *International Journal of Digital Content Technology and its Applications (JDCTA)*, vol. 6, pp. 200-208, 2012.
- [9] A. Agrawal, S. Misra, D. Honbo, and A. Choudhary, "Parallel pairwise statistical significance estimation of local sequence alignment using Message Passing Interface library," *Concurrency and Computation: Practice and Experience*, 2011.
- [10] N. Berger, "GPUs in experimental particle physics," *Bulletin of the American Physical Society*, vol. 57, 2012.
- [11] S. Mittal and Z. Zhang, "Integrating sampling approach with full system simulation : Bringing together the best of both," presented at the IEEE International Conference On Electro/Information Technology (EIT), Indianapolis, USA, 2012.
- [12] S. Mittal, S. Gupta, and S. Dasgupta, "System Generator: The State-of-art FPGA Design tool for DSP Applications," presented at the Third International Innovative Conference On Embedded Systems, Mobile Communication And Computing (ICEMC2 2008), Global Education Center, India, 2008.
- [13] S. Mittal and Z. Zhang, "EnCache: Improving Cache Energy Efficiency Using a Software-Controlled Profiling Cache," presented at the IEEE EIT, 2012.
- [14] Z. Huang, Y. Chen, and J. Nieplocha, "Massive contingency analysis with high performance computing," 2009, pp. 1-8.

- [15] I. Gorton, Z. Huang, Y. Chen, B. Kalahar, S. Jin, D. Chavarría-Miranda, D. Baxter, and J. Feo, "A high-performance hybrid computing approach to massive contingency analysis in the power grid," 2009, pp. 277-283.
- [16] L. Tang and J. McCalley, "Next Generation Online Dynamic Security Assessment," Power Systems Electric Research Center 2011.
- [17] L. Yuan and J. D. McCalley, "Risk-based optimal power flow and system operation state," in *Power & Energy Society General Meeting, 2009. PES '09. IEEE*, 2009, pp. 1-6.
- [18] W. Hongye, C. E. Murillo-Sanchez, R. D. Zimmerman, and R. J. Thomas, "On Computational Issues of Market-Based Optimal Power Flow," *Power Systems, IEEE Transactions on*, vol. 22, pp. 1185-1193, 2007.
- [19] T. Athay, R. Podmore, and S. Virmani, "A Practical Method for the Direct Analysis of Transient Stability," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-98, pp. 573-584, 1979.
- [20] D. Gan, R. J. Thomas, and R. D. Zimmerman, "Stability-constrained optimal power flow," *Power Systems, IEEE Transactions on*, vol. 15, pp. 535-540, 2000.
- [21] R. Zarate-Minano, T. Van Cutsem, F. Milano, and A. J. Conejo, "Securing Transient Stability Using Time-Domain Simulations Within an Optimal Power Flow," *Power Systems, IEEE Transactions on*, vol. 25, pp. 243-253, 2010.
- [22] Y. Yue, J. Kubokawa, and H. Sasaki, "A solution of optimal power flow with multicontingency transient stability constraints," *Power Systems, IEEE Transactions on*, vol. 18, pp. 1094-1102, 2003.
- [23] A. Pizano-Martinez, C. R. Fuerte-Esquivel, and D. Ruiz-Vega, "A New Practical Approach to Transient Stability-Constrained Optimal Power Flow," *Power Systems, IEEE Transactions on*, vol. PP, pp. 1-1, 2011.
- [24] H. Yulong and L. Mingbo, "Transient stability constrained optimal power flow based on trajectory sensitivity, one-machine infinite bus equivalence and differential evolution," in *Power System Technology (POWERCON), 2010 International Conference on*, 2010, pp. 1-6.
- [25] W. Sun, D. Z. Fang, and S. Yuan, "Sensitivity-Based Approach for Optimal Power Flow with Transient Stability Constraints," in *Energy and Environment Technology, 2009. ICEET '09. International Conference on*, 2009, pp. 267-270.
- [26] X. Ye and K. Chongqing, "Transient-constrained dynamic economic scheduling using trajectory sensitivity," in *Power and Energy Society General Meeting, 2011 IEEE*, 2011, pp. 1-7.
- [27] Q. Chen and J. D. McCalley, "Operational defence of cascading sequences," in *Power and Energy Society General Meeting, 2011 IEEE*, 2011, pp. 1-8.
- [28] I. A. Hiskens and M. Akke, "Analysis of the Nordel power grid disturbance of January 1, 1997 using trajectory sensitivities," *Power Systems, IEEE Transactions on*, vol. 14, pp. 987-994, 1999.

- [29] T. B. Nguyen, M. A. Pai, and I. A. Hiskens, "Sensitivity approaches for direct computation of critical parameters in a power system," *International Journal of Electrical Power & Energy Systems*, vol. 24, pp. 337-343, 2002.
- [30] T. B. Nguyen, M. A. Pai, and I. A. Hiskens, "Direct computation of critical clearing time using trajectory sensitivities," in *Power Engineering Society Summer Meeting, 2000. IEEE*, 2000, pp. 604-608 vol. 1.
- [31] I. A. Hiskens and M. A. Pai, "Trajectory sensitivity analysis of hybrid systems," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 47, pp. 204-220, 2000.
- [32] R. C. Degenoff, W. Neugebauer, C. H. Saylor, and S. L. Corey, "Security constrained optimization: an added dimension in utility systems optimal power flow," *Computer Applications in Power, IEEE*, vol. 1, pp. 26-30, 1988.
- [33] J. McCalley. Operational defense of power sysetm cascading outages [Online]. Available: [ewh.ieee.org/conf/tdc/JDMCascadingT&D042308.pdf](http://ewh.ieee.org/conf/tdc/JDMCascadingT&D042308.pdf)
- [34] S. Khaitan and J. McCalley, "High Performance Computing for Power System Dynamic Simulation," in *High performance computing in power and energy systems*, ed: Springer.
- [35] S. K. Khaitan, J. D. McCalley, and C. Qiming, "Multifrontal Solver for Online Power System Time-Domain Simulation," *Power Systems, IEEE Transactions on*, vol. 23, pp. 1727-1737, 2008.
- [36] F. Chuan, J. D. McCalley, and T. Jianzhong, "A Numerical Solver Design for Extended-Term Time-Domain Simulation," *Power Systems, IEEE Transactions on*, vol. PP, pp. 1-1, 2012.
- [37] L. Petzold, S. Li, Y. Cao, and R. Serban, "Sensitivity analysis of differential-algebraic equations and partial differential equations," *Computers & chemical engineering*, vol. 30, pp. 1553-1559, 2006.
- [38] S. K. Khaitan and J. D. McCalley, "A Class of New Preconditioners for Linear Solvers Used in Power System Time-Domain Simulation," *Power Systems, IEEE Transactions on*, vol. 25, pp. 1835-1844, 2010.
- [39] S. Khaitan, J. McCalley, and M. Raju, "Numerical methods for on-line power system load flow analysis," *Energy Systems*, vol. 1, pp. 273-289, 2010.
- [40] K. Stanley and T. Davis, "KLU: a'Clark Kent'sparse LU factorization algorithm for circuit matrices," 2004.
- [41] O. Schenk and K. Gärtner, "Solving unsymmetric sparse systems of linear equations with PARDISO," *Future Generation Computer Systems*, vol. 20, pp. 475-487, 2004.
- [42] T. Overbye. Available: <http://www.powerworld.com/>.
- [43] DIgSILENT. Available: <http://www.digsilent.de/>
- [44] ETAP. Available: <http://etap.com/index.htm>
- [45] Eurostag. Available: <http://www.eurostag.be/>

- [46] Siemens. Available: <http://www.siemens.com>
- [47] H. Fankhauser, T. Adielson, K. Aneros, A. A. Edris, L. Lindkvist, and S. Torseng, "SIMPOW—a digital power system simulator," *Reprint of ABB Review*, p. 1990, 1990.
- [48] CYME. Available: <http://www.cyme.com/>
- [49] F. Milano, "An open source power system analysis toolbox," *Power Systems, IEEE Transactions on*, vol. 20, pp. 1199-1206, 2005.
- [50] C. D. Vournas, E. G. Potamianakis, C. Moors, and T. Van Cutsem, "An educational simulation tool for power system control and stability," *Power Systems, IEEE Transactions on*, vol. 19, pp. 48-55, 2004.
- [51] C. A. Canizares and F. Alvarado, "UWPFLOW: continuation and direct methods to locate fold bifurcations in AC/DC/FACTS power systems," *University of Waterloo*, 1999.
- [52] S. Ayasun, C. O. Nwankpa, and H. G. Kwatny, "Voltage stability toolbox for power system education and research," *Education, IEEE Transactions on*, vol. 49, pp. 432-442, 2006.
- [53] M. Beshir, T. Cheng, and A. Farag, "Comparison of two bulk power adequacy assessment programs: TRELSS and COMREL," 1996, pp. 431-437.
- [54] M. Kumbale, T. Rusodimos, F. Xia, and R. Adapa, "TRELSS: A Computer Program for Transmission Reliability Evaluation of Large-Scale Systems," *Vol*, vol. 2, pp. 3833-1, 1997.
- [55] K. Kahle and C. G. S. Division, "The NEPLAN software package: a universal tool for electric power systems analysis," 2002.
- [56] W. Grady, "PCFLO User's Manual," ed: Univ. Texas Austin [Buy Via Ask\* IEEE], 2000.
- [57] P. M. Associates, *Composite system reliability analysis program comrel users manual*, 1991.
- [58] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *Power Systems, IEEE Transactions on*, pp. 1-8, 2011.
- [59] J. Mahseredjian and F. Alvarado, "Creating an electromagnetic transients program in MATLAB: MatEMTP," *Power Delivery, IEEE Transactions on*, vol. 12, pp. 380-388, 1997.
- [60] J. H. Chow and K. W. Cheung, "A toolbox for power system dynamics and control engineering education and research," *Power Systems, IEEE Transactions on*, vol. 7, pp. 1559-1564, 1992.
- [61] K. Schoder, A. Hasanovic, and A. Feliachi, "PAT: a power analysis toolbox for MATLAB/Simulink," *Power Sy*
- [62] H. Le-Huy and P. Brunelle, "Design and implementation of a switched reluctance motor generic model for simulink simpowersystems," 2005.

- [63] S. K. Khaitan and J. D. McCalley, "High Performance Computing in Power and Energy Systems," in *POWSYS*, ed: Springer, 2012, pp. 43-69.
- [64] J. Astic, A. Bihain, and M. Jerosolimski, "The mixed Adams-BDF variable step size algorithm to simulate transient and long term phenomena in power systems," *Power Systems, IEEE Transactions on*, vol. 9, pp. 929-935, 1994.
- [65] J. Sanchez-Gasca, R. D'Aquila, W. Price, and J. Paserba, "Variable time step, implicit integration for extended-term power system dynamic simulation," 1995, pp. 183-189.
- [66] K. E. Brenan, S. L. Campbell, S. L. V. Campbell, and L. R. Petzold, *Numerical solution of initial-value problems in differential-algebraic equations* vol. 14: Society for Industrial Mathematic stems, *IEEE Transactions on*, vol. 18, pp. 42-47, 2003.s, 1996.
- [67] M. Berzins and R. Furzeland, "An adaptive theta method for the solution of stiff and nonstiff differential equations," *Applied numerical mathematics*, vol. 9, pp. 1-19, 1992.
- [68] F. L. Alvarado, R. H. Lasseter, and J. J. Sanchez, "Testing of trapezoidal integration with damping for the solution of power transient problems," *Power Apparatus and Systems, IEEE Transactions on*, pp. 3783-3790, 1983.
- [69] T. A. Davis, "Algorithm 832: UMFPACK V4. 3---an unsymmetric-pattern multifrontal method," *ACM Transactions on Mathematical Software (TOMS)*, vol. 30, pp. 196-199, 2004.
- [70] S. K. Khaitan, J. D. McCalley, and Q. Chen, "Multifrontal solver for online power system time-domain simulation," *Power Systems, IEEE Transactions on*, vol. 23, pp. 1727-1737, 2008.
- [71] G. M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," 1967, pp. 483-485.