Online Dynamic Security Assessment Using Phasor Measurement Unit and Forecasted Load

by

Qiushi Wang

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2017 by the
Graduate Supervisory Committee:

George Karady, Chair
Anamitra Pal
Keith Holbert

ARIZONA STATE UNIVERSITY

May 2017

ABSTRACT

On-line dynamic security assessment (DSA) analysis has been developed and applied in several power dispatching control centers. Existing applications of DSA systems are limited by the assumption of the present system operating conditions and computational speeds. To overcome these obstacles, this research developed a novel two-stage DSA system to provide periodic security prediction in real time. The major contribution of this research is to develop an open source on-line DSA system incorporated with Phasor Measurement Unit (PMU) data and forecast load. The pre-fault prediction of the system can provide more accurate assessment of the system and minimize the disadvantage of a low computational speed of time domain simulation.

This Thesis describes the development of the novel two-stage on-line DSA scheme using phasor measurement and load forecasting data. The computational scheme of the new system determines the steady state stability and identifies endangerments in a small time frame near real time. The new on-line DSA system will periodically examine system status and predict system endangerments in the near future every 30 minutes. System real-time operating conditions will be determined by state estimation using phasor measurement data. The assessment of transient stability is carried out by running the time-domain simulation using a forecast working point as the initial condition. The forecast operating point is calculated by DC optimal power flow based on forecast load.

To my parents

ACKNOWLEDGMENTS

I want to express my great gratitude to my advisor and mentor, Dr. George Karady, for his support, encouragement, guidance and patience throughout my master degree. He taught me not only the essential method for research, but also a positive attitude towards life and learning.

I thank my committee members, Dr. Anamitra Pal and Dr. Keith Holbert, for their time and valuable comments. I also want to thank Dr. Vijay Vittal for his guidance on my research and support on my educational path.

I'm grateful to my parents and boyfriend, who provide their generous love and support to me. I also thank all my friends for their company and support during these years.

I appreciate the financial support from *US-AID* through project *US-PCASE Exchange Program.*

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

LIST OF SYMBOLS

E_1	Field voltage value,1 in p.u.
E_2	Field voltage value,2 in p.u.
F	Shaft output ahead of reheater in p.u.
K_a	Regulator gain (continuous acting regulator) in p.u.
K_e	Exciter self-excitation at full load field voltage in p.u.
K_f	Regulator stabilizing circuit gain in p.u.
P_{max}	Maximum turbine output in p.u.
R	Turbine steady-state regulation setting or droop in p.u.
$S(1.0)$	Machine saturation at 1.0 p.u. voltage in p.u.
$S(1.2)$	Machine saturation at 1.2 p.u. voltage in p.u.
$SE(E_1)$	Saturation factor at $E1$
$SE(E_2)$	Saturation factor at $E2$
T_1	Control time constant (governor delay) in s
T_1	Servo time constant in s
T_1	Steam reheat time constant in s
T_1	Steam valve bowl time constant in s
T_2	Hydro reset time constant in s
T_a	Regulator time constant in s
$T_d0\prime$	d axis transient open circuit time constant in s
$T_d0\prime\prime$	d axis subtransient open circuit time constant in s
T_e	Exciter time constant in s
T_f	Regulator stabilizing circuit time constant in s
$T_q0\prime$	q axis transient open circuit time constant in s
$T_q0\prime\prime$	q axis subtransient open circuit time constant in s
T_r	Regulator input filter time constant in s
V_{RMAX}	Maximum regulator output, starting at full load field voltage in p.u.

V_{RMIN}	Minimum regulator output, starting at full load field voltage in p.u.
r_a	Armature resistance in p.u.
x_d	Unsaturated d axis synchronous reactance in p.u.
x_d'	Unsaturated d axis transient reactance in p.u.
x_d''	Unsaturated d axis subtransient reactance in p.u.
x_l	Leakage or Potier reactance in p.u.
x_q	Unsaturated q axis synchronous reactance in p.u.
x_q'	Unsaturated q axis transient reactance in p.u.
x_q''	Unsaturated q axis subtransient reactance in p.u.
AEP	American Electric Power
COA	Center of Angles
D	Machine load damping coefficient
DMAPE	Average Daily Mean Absolute Percentage Error
DSA	Dynamic Security Assessment
H	Inertia constant in s
ICSEG	Illinois Center for a Smarter Electric Grid
MIP	Mixed Integer Programming
NERC	North American Reliability Council
OPF	Optimal Power Flow
PDC	Phasor Data Concentrator
PMU	Phasor Measurement Unit
Rated Power (MVA)	Machine-rated MVA
Rated Voltage (kV)	Machine-rated terminal voltage in kV
SE	State Estimation
WAN	Wide Area Network
WECC	Western Electricity Coordinating Council
WLS	Weighted Least Squares

Chapter 1

INTRODUCTION

1.1 Background

1.1.1 Requirements of a successful electrical power service

Electric power has become a major portion of energy supply all over the world after industrial revolution. As the major power supply to a wide range of customers, the qualification of power systems operation has a significant influence on both the happiness of individual's daily life and the economic development of the society. A successful power system operation requires the ability of power supply to provide reliable and economic service to customers.

One requirement for the reliability of the power system is to supply loads with adequate power without interruption. A major achievement for power systems, after hundreds of years of development, is the large area interconnected transmission systems constructed to deliver adequate electric energy to loads. Another important requirement for the reliable power system operation is to maintain power system security, which highlights the capability of power systems to maintain voltage and frequency within certain limitations under disturbances.

Power system security, to be more specific, is the ability of the bulk power electric system to withstand sudden disturbances such as electric short circuits or unanticipated loss of system components [1]. This definition is defined by the North American Reliability Council (NERC), and has been widely accepted in both industry and academic research. A joint working group, by the CIGRE Study Committee 38 and

the IEEE Power System Dynamic Performance Committee, clarified two aspects of security analysis in [2].

A secured power system operation requires that the system, when a sudden disturbance occurs, will:

i) survive the transient period and move into an acceptable steady-state condition, and

ii) in the new steady-state condition, all the voltages and frequency of the power system are within established limits

The latter analysis, to meet steady-state requirement of the post-disturbances system, is the subject dealing with static security analysis. The former concern of transient analysis is called the dynamic security analysis.

1.1.2 Dynamic security assessment system

Dynamic security assessment (DSA) system is one important portion of security assessment system for predicting power system transient stability [3]. This is because operators in the control center need real time automatic transient stability analysis to facilitate decision making. Figure 1.1 illustrates the relationship between power system security analysis and the DSA system.

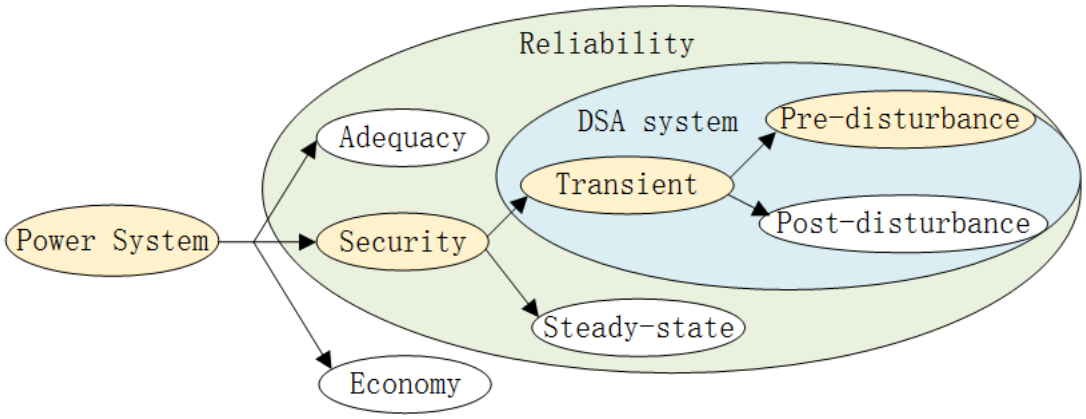

Figure 1.1: Power System Operation Requirements

A DSA system mainly deals with transient stability problems. On-line transient stability studies can be performed both before or after the fault. Post-fault analysis is relatively easier because disturbances happen randomly most of the time. A large disturbance can be any major change in loads, a fault on the transmission line, or a failure in a piece of equipment. Knowing fault conditions will effectively accelerate the computational speed by reducing the complexity of the problem. However, being able to predict endangerments for power systems pre-fault is necessary and crucial for preventing possible failure during operation.

1.1.3 Phasor measurement unit

Phasor measurement unit (PMU) is a device which has the capability to provide accurate synchronous phasor measurements of voltages and currents in power systems. According to IEEE standard for Synchrophasor Measurements for Power Systems [4], the frequency error for PMU is limited within 0.005 Hz, while the precision of the relative voltage angle is 0.02 electrical degrees. Furthermore, PMUs can communicate with Phasor Data Concentrator (PDC) to upload and collect data for a wide area network (WAN). Therefore, PMUs have the capability to provide accurate phasor measurements for the calculation of system states in WAN.

In a modern power system, sophisticated monitoring systems, along with control and protection systems, have been applied to diminish the impact of unexpected disturbances in the system. Because using PMU data can reduce the computational burden as well as improve the accuracy of calculated results, many countries around the world have been building PMU-based monitoring systems. For example, about 2,000 commercial-grade PMUs were installed by 2015 in the U.S. [5]. The number of PMU installations is still increasing.

1.2 Literature Review

The concept of the dynamic security assessment (DSA) system was first brought into sharp concern in late 1960s-early 1970s when the first well-known massive power system failure happened on 9th November 1965 in the United States [6]. The failure, reported as the Northeast Power Failure, affected 30 million people in the U.S. and Canada, and lasted in some areas for as much as 13 hours. The initial cause of the interruption was a mis-operation of a backup protection relay on a weak transmission line between America and Canada. That relay was set too low and disconnected a line with a heavy load. As a result, the load was shifted to the remaining four lines, which triggered the overload protection successfully for each of the four lines and initiated the power failure.

After carefully studying and reporting the stages and causes of the failure, a work group from federal power commission pointed out the necessity of developing a dynamic security assessment system to provide rapid security check for modified system configurations. An essential function of DSA systems is to determine system capability to remain synchronous when severe disturbances happen. Real-time analysis for transient stability of a system can largely assist operators decision making process to prevent possible failures.

Since then, an extraordinary amount of contributions has been made toward the goal of providing real-time transient stability prediction and assessment for power systems. Among all these efforts, there are three fundamental approaches to conduct transient stability analysis. These three methods are numerical integration methods [7, 8, 9, 10], energy function methods [11, 12, 13, 14], and dynamic state estimation (SE) [15, 16, 17]. Besides the three fundamental approaches, probabilistic methods

[18, 19, 20], and data mining methods [21, 22, 23, 24, 25] are developed to improve the accuracy and computational speed of the fundamental approaches.

The energy function method is also known as the second (direct) method of Lyapunov. It is a well accepted method and had been put into practices of DSA systems in 1980s for both off-line and on-line studies [3].

The energy function for individual machines is a developed version of the Kimbarks two area criterion. The model assumes that the system is represented by a classical model and the damping is negligible. Therefore, the equation of motion for the multi machine system is given in the form of (1.1) in [7],

$$M_i\dot{\omega}_i = P_{mi} - P_{ei} - \frac{M_i}{M_T}P_{coi} \tag{1.1}$$

where

ω_i is the angular frequency for generator i

P_{mi} is the mechanical power for generator i

P_{ei} is the electrical power for generator i

M_i is the respective generator inertial constant

M_T is the system inertial constant

P_{coi} is the total accelerating power. For a n-bus system, it can be calculated by:

$$P_{coi} = \sum_{j=1}^{n}(P_{mj} - P_{ej}) \tag{1.2}$$

The energy function for each generator can be expressed as:

$$V_i = V_{KEi} + V_{PEi} \tag{1.3}$$

$$V_{KEi} = \frac{M_i}{2}\omega_i^2 \tag{1.4}$$

$$V_{PEi} = -\int_{\theta_{ai}}^{\theta_{bi}}(P_{mi} - P_{ei} - \frac{M_i}{M_T}P_{coi})\mathrm{d}\theta_i \tag{1.5}$$

We will need to calculate the derivative of V_i with respect to time along the trajectories of the system. The derivative of V_i is evaluated to determine the capability for the system to withstand a major disturbance. For the classical system neglecting damping, the system will remain stable with a disturbance only if the solution exists for (1.6) and the rotor angle is within the requirement of (1.7).

$$\dot{V}_i = 0 \tag{1.6}$$

$$0 < \theta_i < \frac{\pi}{2} \tag{1.7}$$

Power system analysts perform off-line simulation of the system based on energy functions and determine the critical clearing time of a fault, which is the edge of the system to remain stable under this fault.

The energy function method has been performing an important role in assessing first-swing transient stability. However, this model fails to provide accurate transmission limitations to constrain real-time operation. The assumption of constant generator main field-winding flux linkage causes inaccurate performance of this model for a longer period analysis. Also neglecting the damping power makes the model inaccurate for a large system with relatively weak ties.

The dynamic state estimator concentrates more on monitoring and predicting system states after fault occurrence. The dynamic state estimator examines the states of generators in real time and determines whether the system can successfully withstand a disturbance after it occurs. The dynamic state estimator for power system is developed based on Kalman Filter, in which the state of a system is the sum of previous states and weighted measurements [26].

Given a dynamic system

$$x(k+1) = x(k) + (\triangle t)r, \qquad x(0) = x_0 \tag{1.8}$$

With the associated measurement system

$$z(k) = Hx(k) + v(k), \qquad k = 0, 1, \ldots, \frac{\tau}{\triangle t} \tag{1.9}$$

where, $\triangle t$ is the sampling period

r is the maximum rate-of-change vector

τ is the period of operation.

The estimate $\widehat{x}(k)$ is given by (1.10) in [27]

$$\widehat{x}(k+1) = \widehat{x}(k) + W(k+1)[z(k+1) - H\widehat{x}(k)] \tag{1.10}$$

The dynamic state estimator has the capability to detect a severe fault condition and potential risks after a major disturbance. However, the performance of a dynamic state estimator largely depends on the selection of the measurement redundancy, meter accuracy, and the value of W matrix. The total cost of the monitoring and communication system will be a big investment. Even if the system is fully set up, Kalman gain W can to be obtained by Kalman filter [28]. Therefore, the dynamic state estimator is suitable for post-fault DSA studies.

The probabilistic methods take into account both the configuration transition rates and the conditional probabilities of security transitions [18]. This method studies the probability distribution of the time to insecurity for a large number of different disturbances. These disturbances may occur at different locations and with different clearing schemes. The security assessment for these disturbances above are based on the solution from numerical integration. The large number of training cases of numerical integration will add heavy computational burden to system. As a result, the probabilistic methods are more of a planning type and have served as a learning stage for other approaches.

With the application of PMU, a large amount of data needs to be collected and dealt with. Data mining technology can help to speed up the analysis by designing

proper models for the system. One application of data mining methods in power systems is to predict the system transient stability based on the value of system measurements and states. Because the model of security condition of power system is a discrete target variable, the predictive model for system security is chosen to be classification, which includes decision tree (DT) [29, 23, 30], artificial neural network (ANN) [21, 31, 24, 25], and supporting vector machine (SVM) [32, 33].

Although data mining methods have the capability to provide fast prediction of system security, certain concerns limit the practical application of data mining technologies in power systems. The predictive model designing process relies significantly on the knowledge and experience of the designer, which may be hard to control. In addition, the training and testing of the data mining models need a large data set of system behavior from off-line study or historical records. Storage of the large amount of data increases the cost and information security risk. Also, the real-time communication of a large amount of data remains a practical obstacle.

Numerical integration methods are applicable solutions of the differential equations. All the commercial software, for transient stability analysis today, are based on different numerical integration methods.

Compared to other models for the study of transient stability, numerical integration provides the most accurate prediction results, which is a valuable reference to assist operational decision making. Although the large computational burden of the numerical integration methods slowed the application of it in on-line DSA systems, the recent development of parallel algorithms [8, 34] and distributed methods [35, 9] has made real time simulation feasible for wide area network.

A Task Force of the Computer and Analytical Methods Subcommittee of the Power Systems Engineering Committee defined several research areas of parallel computation as applied in power systems in [8]. Among all the power system problems,

the transient stability problem, which is presented by nonlinear differential-algebraic equations, had attracted special attention in research because of the inherent need for on-line analysis to predict system security. Furthermore, the group summarized two well-developed computational algorithms, including vector and array processor and distributed processor, for parallel computation. This fundamental work demonstrated the feasibility of on-line time-domain simulation for transient stability problems.

Following the work in [8], Huang and Chen developed a distributed dynamic security assessment (DDSA) system to perform real time transient analysis for power systems. The architecture of the DDSA system implied two levels of parallelism, namely multi-case parallelism and network-wise parallelism. The test result of the IEEE 39-bus test system demonstrated the feasibility of on-line application for DSA system using distributed time domain simulation.

CIGRE working group C4.601 identified state of the art applications in on-line DSA systems and described worldwide practical installation of these systems in 2004 [36]. Figure 1.2 shows the installations in service and under development around the world. These on-line applications of DSA systems demonstrate the practicability of the utilization of time-domain simulation in transient stability study. For example, the DSA server developed in Nemmco (Australia) has the capability to process 55 contingencies for a 2100-bus 300-generator model within 10 minutes [36].

The implication of real-time DSA system can be categorized into two groups. One group is post-fault DSA system, which is designed for monitoring and detecting faults and analysis whether the fault will cause system failure after detecting the fault. In places like Australia, Brazil, and Japan, for example, time domain simulation for transient stability analysis will be triggered, only if a fault was detected in the system. This architecture is computationally efficient. However, these systems cannot help to prevent system failure before major disturbances occur.

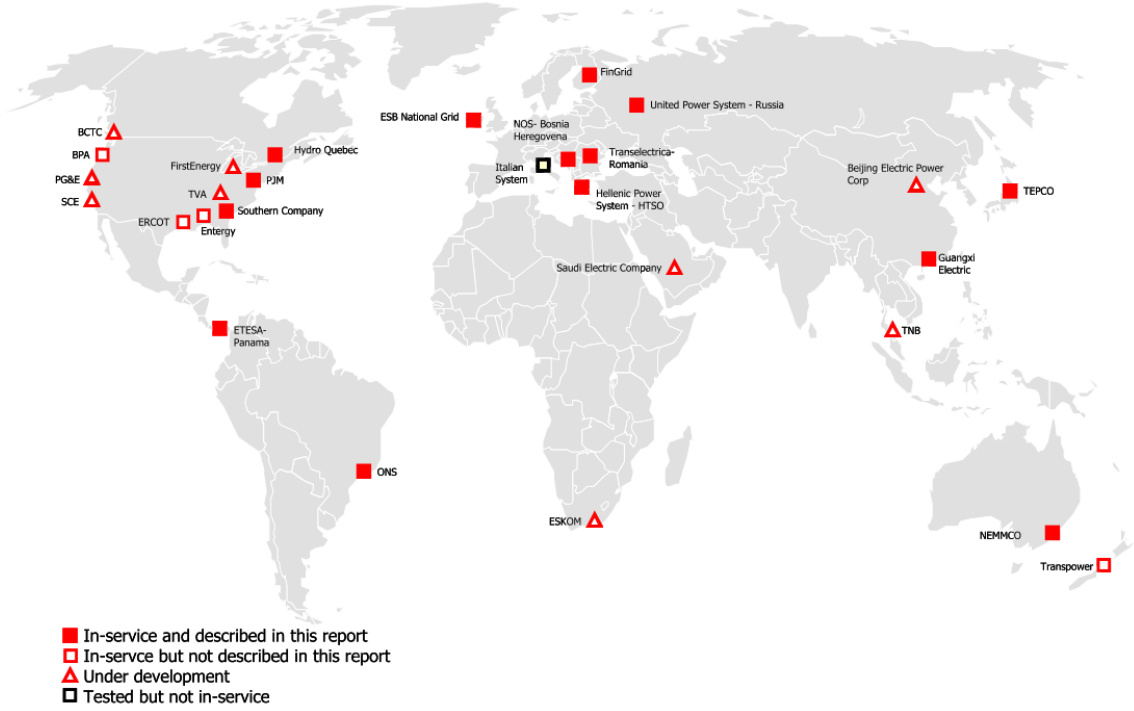

Figure 1.2: Reported On-line DSA Installation [36]

It is not enough to implement only post-fault DSA system to ensure system security. Recall the blackout in August 2003. The failure of fault detection system missed operator's opportunities to take proper actions to prevent the subsequent faults[37]. A post-fault DSA system will not work if the system failed to detect a failure. On the contrary, a DSA system with pre-fault prediction [10] is able to alert operators to the potential risk before the system experiences a fault, so that operators are able to take preventive actions to improve reliability.

Pre-fault prediction and precautions are necessary for reliable operation. However, the majority of previous research focuses on post-fault transient stability analysis, while little focus is placed on pre-fault analysis of the future network conditions.

PJM system control center finished the installation and test for a pre-fault DSA system in 2006 [10]. Figure 1.3 shows key components in the DSA system in PJM. The system takes measurements from SCADA system to calculate system states, and

performs pre-fault stability analysis to evaluate how well the system can withstand credible contingencies.

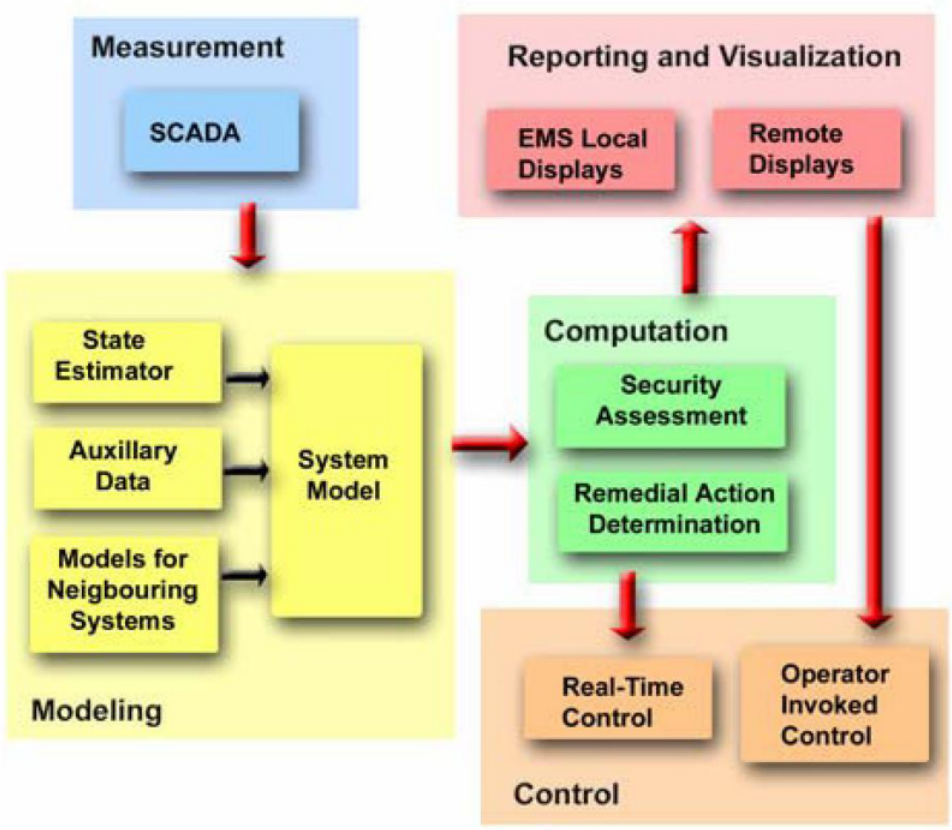

Figure 1.3: Key Components in DSA System in PJM[10]

Xu at el. proposed an intelligent system (IS) method to deal with pre-fault DSA systems in [25]. The system uses pre-fault steady-state power system variables as input to determine the risk of the system, and time-domain simulated contingency data to train model. When it is well trained, the system can determine the potential fault very quickly.

The performance of existing pre-fault DSA systems can be improved by including demand change in the future system with economical consideration. Very little work has been done on real-time pre-fault DSA systems that incorporate load forecast.

1.3 Motivation and Scope of the Research

Today, power systems are facing fundamental changes, in terms of system configuration, on both supply and demand sides. On the supply side, the large fossil fuel based synchronous machines are being replaced by renewable resources based generators. On the demand side, a growing number of distributed generators and power electronic devices have been being applied and connected to the distribution system. What is more, the utilization of advanced communication and automation systems increases the complexity of the power systems.

These trends place two challenges for the security operation of power system. First, the diversity and complexity growth increases the dimension of the model. Second, the load growth pushes the system to operate close to secure limitation. As a result, the security risk has been higher than ever.

Renewable energy has been developed to replace conventional plants driven by fossil fuel. According to Western Electricity Coordinating Council (WECC), the net generation of solar and wind in 2015 in the Western Interconnection is up to 7% [38]. The participation factor of wind and solar is continuously growing. However, many of the renewable sources show an inherently intermittent characteristic. This characteristic makes the power demand of conventional plants more variable. This phenomenon is named as the "Duck Curve", as shown in Figure 1.4 by the California Independent System Operator (CAISO) [39].

The "Duck Curve" plots actual and estimated net load of conventional plant versus time on a particular winter day. It shows that, in 2020, the midday demand will decrease by about 50% of the actual load while the peak demand will increase. The dual effect of the midday demand decrease and the peak demand increase will cause significant demand drop and steep ramp during midday hours.

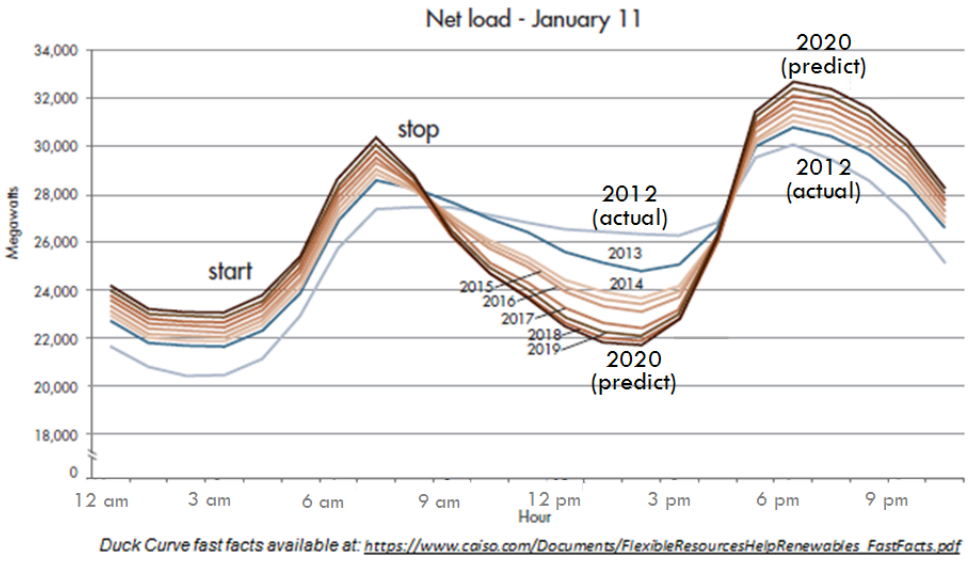

Figure 1.4: CAISOs 2013 Illustration of the "Duck Curve [39]

To ensure the reliability under the changing system configuration, on-line transient stability analysis, as a function of the DSA system, must have the capability to provide accurate predictive security assessment. On the other hand, the demand dropping of net load relaxes the limitation for both voltage and power transfer. Therefore, chances increase for the system to operate at the forecast operating point based on economic dispatch.

However, the existing DSA system lacks the ability to capture the predictable changes in the system. As a result, overwhelming operating margins will be set by the existing DSA system during the demand decrease, while part of severe endangerments cannot be determined during the steep ramp. In order to overcome the shortcomings of the existing DSA system, a new two-stage DSA system is developed in this thesis.

The primary objective of this thesis is to develop a new DSA system to provide more accurate and economic security assessment compared to the existing DSA systems. As a DSA system, pre-fault time-domain simulation runs periodically in a half-hour interval to evaluate how well the system can withstand disturbances. Be-

fore the simulation, states of the corresponding system operating points should be accessible. If any endangerments are detected, the system should report an alarm and generate plots for the unsecured cases.

The main contribution of this research is to prove the possibility and necessity of incorporating phasor measurement unit (PMU) and forecast load data with DSA systems. Predicted working points, instead of current working points, are first used in a DSA system to provide predictive pre-fault security assessment.

The new DSA system runs state estimation using data from PMUs to obtain the system operating condition. The utilization of PMU in optimal places results in a more accurate state estimation.

A significant difference between the existing DSA system and the proposed DSA system lies in the distinct selection of initial states for transient stability analysis. Instead of using current operating states, the new DSA system uses half-hour predicted operating states to initialize transient stability analysis. Using predicted operating states enables the system to provide predictive and more accurate assessment for the system in the near future.

An approach to calculating the predicted system states is proposed in this thesis. The input data for the calculation is the forecast load. The forecast load data, together with the cost of each generation, will be fed into DC optimal power flow (OPF). The results of DC OPF will be trusted as predicted operating states and will be used to initialize the transient stability analysis in the new DSA system.

1.4 Thesis Outline

Chapter 1 introduces the background, related work, motivation and scope of this research of on-line DSA systems. A brief introduction of PMU and forecasted load data is also presented.

Chapter 2 explains the methodology of this work, including optimal PMU placement, PMU-based linear state estimation, state forecast using forecasted load data based on DC optimal power flow, and criteria for power system DSA systems.

Chapter 3 describes the program developed for the proposed two-stage dynamic security assessment system. The method to formulate the matrix in MATLAB, the interface between Python and DSA Tools are provided in this chapter. Detailed data for the simulation platform, IEEE 118-bus test system static and dynamic model, and forecasted load profile are also included.

Chapter 4 illustrates the performance of the new DSA system with simulation results. A brief economic benefits analysis is also included in this chapter.

Chapter 5 concludes all important outputs from the study and suggests other steps that can be considered for future work.

Chapter 2

DYNAMIC SECURITY ASSESSMENT SYSTEM

2.1 Introduction to the Dynamic Security Assessment System

DSA system plays an important role in power system control room for operational decision making and remedial control processes. The computational framework for current practice of DSA system includes two stages, as shown in Figure. 2.1 .

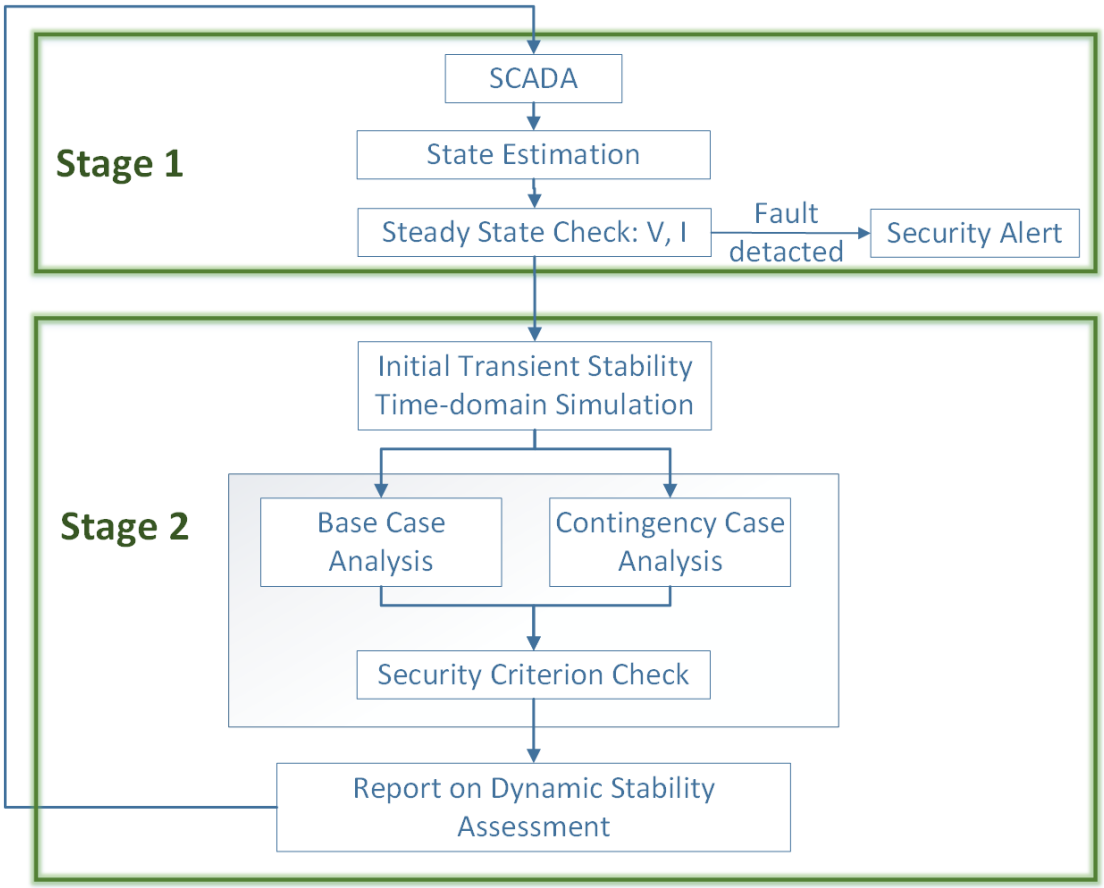

Figure 2.1: Computational Framework of Current Practice of DSA System

Power systems operate under the presence of random disturbances, so simply assuming steady-state operating conditions will introduce large prediction errors and

may cause DSA failure. Therefore, in the first stage, DSA system will calculate and check the steady state operation condition of the system. The state estimator calculate system status, including voltage magnitude and angle for each bus, using measurement records from the Supervisory Control and Data Acquisition (SCADA) system. The measurements stored in the SCADA system are not synchronous and contain voltage magnitude of each bus, real and reactive power of each generator and load. After the state estimator calculates the states of the present working point, the DSA system will check whether the system is operating within pre-defined operation limitations. These limitations include voltage limits, thermal limits, and power output limits of generators. Alert signals will be sent when violations are detected for the above limits.

The second stage will evaluate the transient stability behavior of the system for pre-defined contingencies. Stage two will be performed only after the system is determined as steady state operating condition within limitations in stage one. Time domain simulation for transient stability, based on numerical integration, will be performed to determine whether the pre-fault working point is secure or not.

Existing DSA systems assume that the load of a power system does not have ramp changes, and the security behavior of the system will remain almost the same in a short period. Therefore, the current working point can represent the future working point which is desired to study. As a result, the existing DSA system uses present working point to initialize the time-domain simulation for power system transient stability studies. During the time-domain simulation of the system behavior, a list of contingencies will be added to the system to calculate accurate generator rotor angle after disturbances. The rotor angle will be evaluated to determine whether system can withstand the disturbance.

The DSA system will generate a security report at the end of an assessment cycle.

A full DSA assessment cycle is determined by the computational speed of the transient stability analysis, and is typically about 30 minutes to 1 hour.

2.2 Proposed Dynamic Security Assessment System

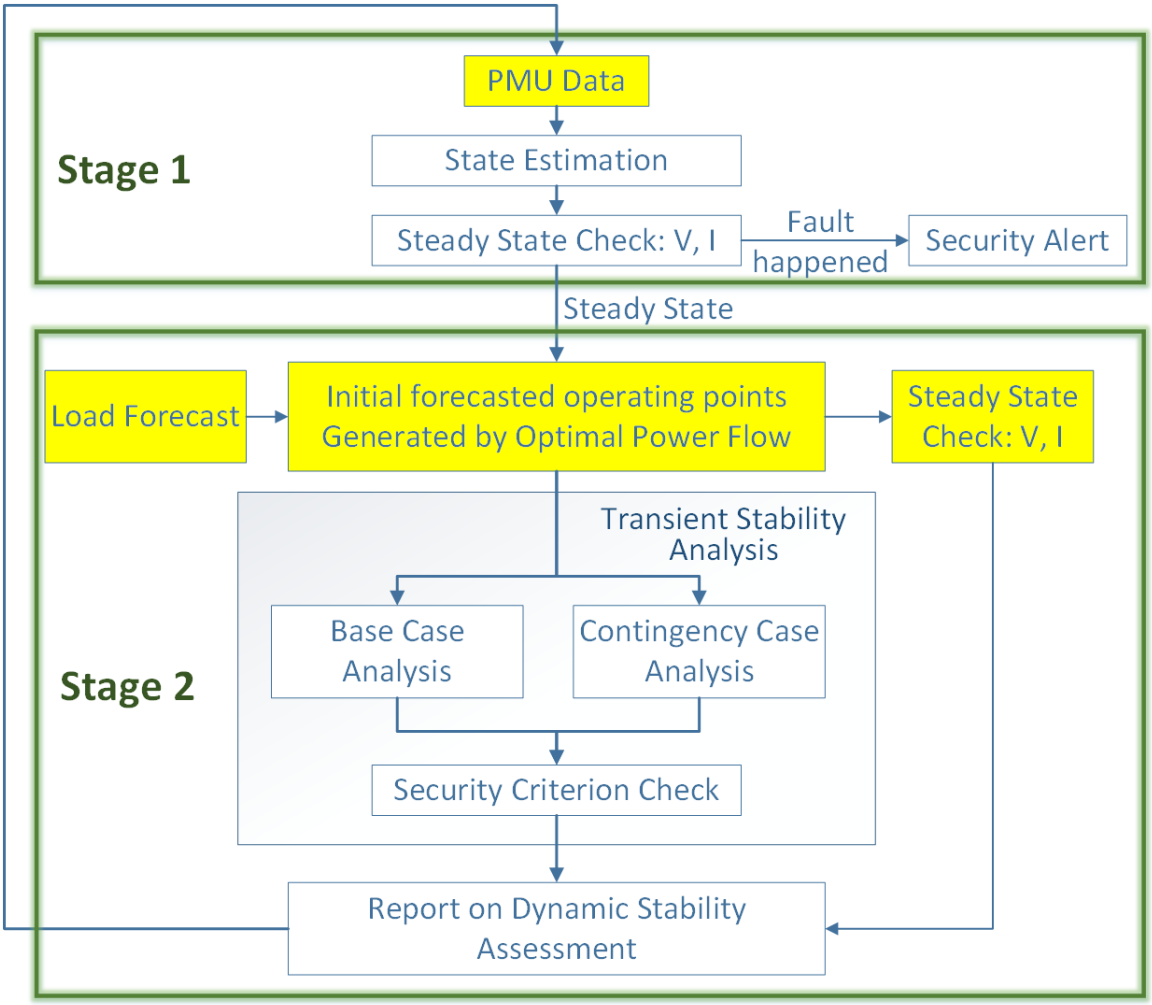

Figure 2.2: Computational Framework of Current Practice of DSA System [40]

As discussed in the motivation section in Chapter 1, the load demand of the power system will increase continuously, while the slope of the load demand will be more steep. These changes will cause the assumption of similar working points to be invalid. As a result, a large error will appear in the prediction result. To improve

the performance of the existing DSA system, a new DSA scheme is proposed in this research. Figure 2.2 shows the computational framework of the proposed DSA system.

Instead of non-synchronous measurements from SCADA system, synchronous phasor measurements from PMUs are used to perform the state estimation in the proposed DSA system. PMU measures the voltage phasors of the bus and present phasors along the line branches. The measurements from PMU allows linear state estimation, which will provide faster and more accurate state result.

The method for transient stability analysis remains time-domain simulation in the new DSA system after considering the speed and accuracy. A summary of advantages and disadvantages for the five transient stability analysis methods introduced in Chapter 1 are listed in Table 2.1. Time-domain simulation can provide accurate prediction results with acceptable computational burden.

Table 2.1: Comparison of the Transient Stability Analysis Methods

Method	Advantages	Disadvantages	Pre-fault On-line DSA?
Time-domain Simulation	Accurate	Computational Burden	Yes
Energy Function	Fast	Only for first-swing transient; not accurate	Yes
Probabilistic Methods	Take consideration of risk probability	Heavy computational burden	For long-term system planning
Dynamic SE	Most accurate	Need real-time PMU data	For post-fault on-line DSA
Data Mining Methods	Fast	Need support of a large set of data base, and real-time PMU data	Yes

Another improvement in the proposed DSA system is to use the forecast operating point, instead of the current operating point, to initialize the time domain simulation

of transient stability studies. The forecast operating point is obtained by performing Optimal Power Flow (OPF) studies. An example is examined below to illustrate the difference between the existing DSA system and the proposed DSA system.

Assume a typical load profile as shown in Figure 2.3. The typical load profile is obtained based on the National Grid load profile in the United Kingdom on July 24, 2016. It is assumed that the present operating point is 7:30 am, indicated by the green X in the figure. The transient stability behavior of interest is at the future operating point of 8:00 am, which is represented by the black X. The analysis period between the present operating point and the future operating point is 30 minutes. The existing DSA system uses the present operating point at 7:30 am to analyze the behavior of the system at future operating point of 8:00 am.

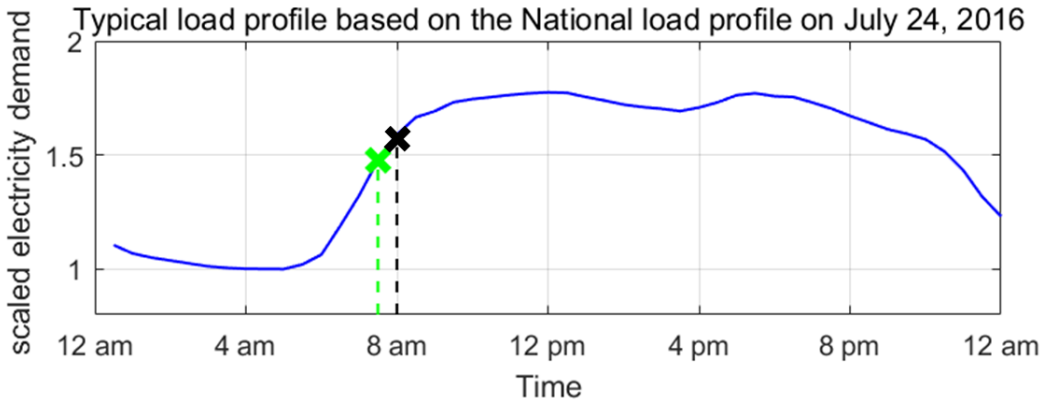

Figure 2.3: Present Working Point Illustrated on a Typical Load Profile [41]

On the contrary, the proposed DSA system uses forecast operating point to analyze system behavior at the future operating point. As shown in Figure 2.4 it uses a forecast operating point for 8:00 am to analyze the future operating point of 8:00 am.

The forecast operating point is based on the forecast load. Hippert et al. pointed out that it is relatively easy to obtain forecasts with about 10% mean absolute percent error, benefited from the development of weather forecasts and artificial intelligence techniques [42]. According to [43], the average daily mean absolute percentage error

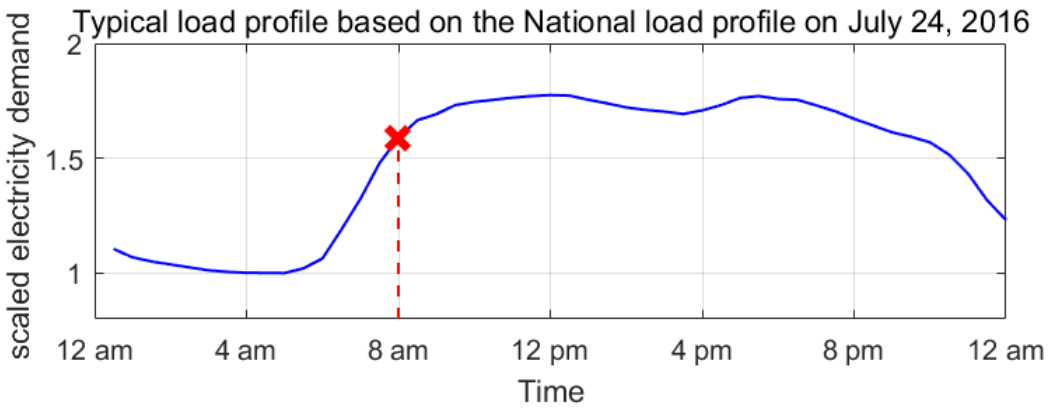

Figure 2.4: Forecast Working Point Illustrated on a Typical Load Profile

(DMAPE) for the short-term load forecast of PJM Electricity Markets in the United States is less than 1.61%.

Therefore ideal load assumption can be made to simplify the analysis. In this research, we assume ideal short-term load forecast without error. Furthermore, it is assumed that all the operating points were calculated by OPF without any changes made by operators. Therefore, the forecast operating points are the same as the actual operating points of the system.

Applying forecast operating point to initialize the transient stability study will improve the performance of the DSA system in two aspects. First, the security prediction of a system will become more accurate if the load forecast and the forecast operating point is accurate enough. Second, using a predicted working point to predict system behavior can relax the requirements of computational speed because the period between the present working point and the future working point provides a time window to perform the analysis.

2.3 PMU-based Linear State Estimation

Static state estimation in a power system refers to the procedure of calculating the voltage phasor at all of the system buses at a given point of time [40]. In a power

system with m measurement and n states, the relationship between the measurement vector $\hat{z}$ and the system states x_i can be expressed as a function of $h_i(x)$ and error e_i:

$$\hat{z} = \begin{bmatrix} h_1(x_1, x_2, \ldots, x_n) \\ h_2(x_1, x_2, \ldots, x_n) \\ \vdots \\ h_m(x_1, x_2, \ldots, x_n) \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} = \hat{h}(\hat{x}) + \hat{e} \tag{2.1}$$

where

$\hat{h}^T = \big[h_1(x), h_2(x), \ldots, h_m(x)\big]$

$h_i(x)$ is the relationship function between measurement z_i and the state vector $\hat{x}$

$\hat{x}^T = \big[x_1, x_2, \ldots, x_n\big]$

$\hat{e}^T = \big[e_1, e_2, \ldots, e_n\big]$.

For a given set of measurements, the best estimate of system states can be obtained by solving the Weighted Least Squares (WLS) optimization problem:

$$\begin{aligned} \text{min.} \quad & \sum_{i=1}^{m} W_i i_i^r \\ \text{s.t.} \quad & z_i = h_i(x) + r_i, i = 1, 2, \ldots, m \end{aligned} \tag{2.2}$$

where

$r_i = z_i - E(z_i)$, which is the residual of measurement z_i

$W_{ii} = \sigma_i^{-2}$, which is the weight inversely related to the assumed error variance σ_i^2 for that measurement.

In the application of WLS estimator in power systems, it is common to assume the measurement errors are independent and the expected value of $E(e_i)$ of the measurement error e_i is zero. Therefore, the minimization problem of (2.2) in power systems

is equivalent to minimizing the following objective function:

$$\begin{aligned} J(x) &= \sum_{i=1}^{m} (z_i - h_i(x))^2 / R_i i \\ &= \big[z - h(x)\big]^T R^{-1} \big[z - h(x)\big] \end{aligned} \tag{2.3}$$

Where:

$R = diag\{\sigma_1^2, \sigma_2^2, \ldots, \sigma_m^2\} = E\big[e \cdot e^T\big]$.

In the existing DSA system, measurements of the system are real and reactive power injection at buses. These measurements have a nonlinear relationship with system states. The minimum solution for the objective function in (2.3) locates the point where the derivative of the objective function is equal to zero:

$$\begin{aligned} g(x) &= \frac{\partial J(x)}{\partial x} \\ &= -\big[\frac{\partial h(x)}{\partial x}\big]^T \big[R\big]^{-1} \big[z - h(x)\big] \\ &= 0 \end{aligned} \tag{2.4}$$

Define the measurement Jacobian H:

$$H(x) = \big[\frac{\partial h(x)}{\partial x}\big] \tag{2.5}$$

Each element in the measurement Jacobian matrix is a nonlinear function of voltage magnitudes and voltage angles at the corresponding bus and adjacent buses. The calculation of these elements in $H(x)$, which will vary in every iteration, adds a heavy computational burden to the state estimator.

In the proposed DSA system, PMU-based state estimation is implied to improve the computational speed and accuracy. It is assumed that the power system has enough PMUs with enough channels. The measurement record from PMU contains bus voltage phasor at its associated bus and branch current phasors along all branches connected to the associated bus. A linear relationship between measurements and

system states can be established based on the general positive sequence two-port π -model for transmission lines shown in Figure 2.5.

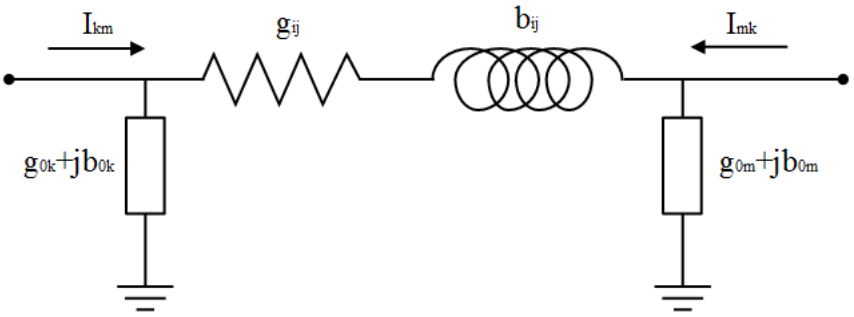

Figure 2.5: Equivalent Circuit for A Transmission Line

The relationship between PMU measurements and the system status is linear, because PMU records and calculates voltage and current phasors. The relationship between current phasor measurements and system states can be derived by a nodal equation [44].

$$\begin{bmatrix} \vec{V_k} \\ \vec{V_m} \\ \vec{I_{km}} \\ \vec{I_{mk}} \end{bmatrix} = \begin{bmatrix} M_B \\ Y_f \end{bmatrix} \begin{bmatrix} \vec{V_k} \\ \vec{V_m} \end{bmatrix} + \begin{bmatrix} e_{Vk} \\ e_{Vm} \\ e_{Ikm} \\ e_{Imk} \end{bmatrix} \tag{2.6}$$

where

$M_B = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$, which is the incidence matrix indicting PMU connections

$Y_f = \begin{bmatrix} Y_{km} + Y_{k0} & -Y_{km} \\ -Y_{km} & Y_{km} + Y_{m0} \end{bmatrix}$, which is the admittance matrix relating measured currents to system states.

The entries of the new Jacobian matrix are constant numbers depending on the system configuration. For linear state estimation, the minimized objective as (2.3)

can be formulated as follows:

$$\begin{aligned} J &= (z - Hx)^T W (z - Hx) \\ &= z^T W z - x^T H^T W z - z^T H W x + x^T H^T W H x \end{aligned} \tag{2.7}$$

Taking the derivative of the objective function in (2.7) results in:

$$\frac{\partial J}{\partial x} = -2 z^T W H + 2 x^T H^T W H = 0 \tag{2.8}$$

Therefore, the best estimation of the system states x is:

$$x = \left[(H^T W^T H)^{-1} H^T W^T \right] z \tag{2.9}$$

where

$\left(H^T W^T W H\right)^{-1} H^T W^T$ is the pseudo inverse of $H^T W^T$

2.4 Optimal PMU Placement

The objective of the PMU placement problem is to guarantee observability of the system with a minimum number of PMU installations. An integer programming based method is formulated as (2.10)-(2.11) to solve the PMU placement problem [45].

$$\text{min.} \sum_{i=1}^{n} w_i \cdot x_i \tag{2.10}$$

$$\text{s.t.} f(X) \succeq \widehat{b} \tag{2.11}$$

where

w_i is the cost of the PMU installation at bus i

$\widehat{b}$ is an $n \times 1$ vector whose entries are commonly all ones

X is a binary vector indicating PMU placement.

The entries of the binary vector X are defined as follows:

$$x_i = \begin{cases} 1, & \text{if a PMU is installed at bus i} \\ 0, & \text{otherwise} \end{cases} \tag{2.12}$$

It is assumed that the flow measurement and the zero injection are ignored for the original system. The system connection configuration can be expressed in a binary incidence matrix A, whose entries are as follows:

$$A_{i,j} = \begin{cases} 1, & \text{if } i = j \text{ or } i \text{ and } j \text{ are connected} \\ 0, & \text{otherwise} \end{cases} \tag{2.13}$$

To obtain full observation of a system, each bus in the system should have at least one connection to the PMU installation bus. Therefore, the $f(X)$ matrix is a matrix which indicates the connection conditions between each bus and PMU installation. $f(X)$ can be obtained from the production of the binary incidence matrix A and the binary PMU placement matrix X:

$$f(X) = AX \tag{2.14}$$

The procedure for building the $f(X)$ matrix will be illustrated using IEEE 9-bus system as an example. The IEEE 9-bus test system is shown in Figure 2.6.

The A matrix for the 9-bus system is:

$$A = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{2.15}$$

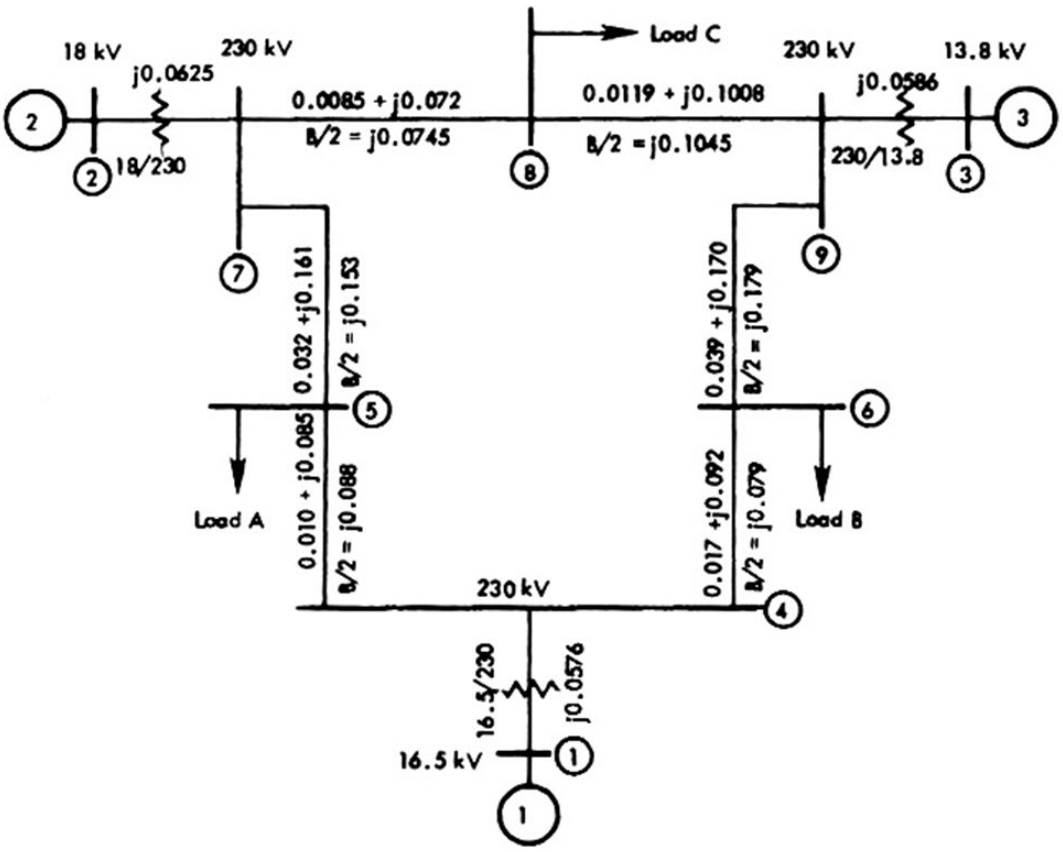

Figure 2.6: IEEE 9-bus Test System

The inequality constraint is:

$$f(X) = AX = \begin{cases} x_1 + x_4, & \geqslant 1 \\ x_2 + x_7, & \geqslant 1 \\ x_3 + x_9, & \geqslant 1 \\ x_1 + x_4 + x_5 + x_6, & \geqslant 1 \\ x_4 + x_5 + x_7, & \geqslant 1 \\ x_4 + x_6 + x_9, & \geqslant 1 \\ x_2 + x_5 + x_7 + x_8, & \geqslant 1 \\ x_7 + x_8 + x_9, & \geqslant 1 \\ x_3 + x_6 + x_8 + x_9, & \geqslant 1 \end{cases} \tag{2.16}$$

Taking the constraint at bus 1 as an example, $x_1 + x_4 \geqslant 1$ indicates that at least one PMU should be installed at either bus 1 or bus 2 to obtain system observation for bus 1. The optimization result shows that at least three PMUs should be installed at bus 4, 7, and 9.

2.5 Forecast System States by DC Optimal Power Flow

The proposed DSA system needs the knowledge of forecast system states to perform transient stability analysis, as demonstrated in the computational framework of the proposed DSA system in Section 2.2. The calculation of the forecast working point can be formulated as an optimal power flow (OPF) problem. The OPF problem solves the power flow of the system with consideration of economic generation dispatch. Based on which solution type the system uses to deal with the power flow problem, the OPF problem can be divided into two main categories: AC OPF and DC OPF.

For an n-bus system, the DC OPF problem can be formulated as [46]:

$$\begin{aligned}
\text{min.} \quad & \sum_{i=1}^{n} F_i(P_{geni}) \\
\text{s.t.} \quad & g_P(\theta, P_g) = B_{bus}\theta + P_{bus,shift} + P_d + G_{sh} - C_g P_g = 0 \\
& h_f(\theta) = B_f\theta + P_f, shift - F_{max} \preccurlyeq 0 \\
& h_f(\theta) = -B_f\theta - P_f, shift - F_{max} \preccurlyeq 0 \\
& P_{geni}^{min} \preccurlyeq P_{geni} \preccurlyeq P_{geni}^{max} \\
& \theta_{geni}^{min} \preccurlyeq \theta_{geni} \preccurlyeq \theta_{geni}^{max}
\end{aligned} \tag{2.17}$$

where

P_{geni} is the desired real power generation of generator i,

$F_i(P_{geni})$ is the cost function of generator i,

C_g is the generator connection matrix,

G_{sh} is the approximated amount of power consumed by the constant shunt elements,
F_{max} is the vector of flow limits,
P_{geni}^{min} and P_{geni}^{max} are the lower bound and upper bound of the capacity of real power generation for generator i.

In this research, DC OPF is applied in the DSA scheme because the performance of OPF is not of critical interest in the scope of this study. Compared to AC OPF, DC OPF can avoid nonlinear or non-convex problems, which are common in AC OPF. The method of OPF can be replaced in further studies.

The forecast working points depend on both forecast load and the cost curve of each generator. The short-term forecast loads are available from load forecasting companies, while the cost curve of each generator is reported from generation companies. It is assumed that there is no human interface to the power generation of the system. Therefore, the results of DC OPF are trusted as forecast operation points of the system.

2.6 Transient Stability Security Criterion

Transient stability, as described in Chapter 1, is the ability for a power system to withstand severe disturbances, such as a fault on the transmission facilities. During a disturbance, a low frequency oscillation of generator angle δ will be superimposed on the synchronous speed ω_0. The change of the generator angle δ can be solved from the equation of motion, or swing equation, as (1.1) in Chapter 1.

The Center Of Angle (COA) is defined as:

$$\delta_{COA} = \frac{\sum_{j=1}^{n} \delta_j H_j}{\sum_{j=1}^{n} H_j} \tag{2.18}$$

where

δ_i is the respective internal generator rotor angle,

H_i is the respective generator's inertia time constant.

The COA in a defined region can be calculated form the simulation result of generator rotor angles. A variable is defined $\triangle\delta$ to represent the difference between rotor angle of generator i and the COA of a region:

$$\triangle \delta_j = \delta_j - \delta_{COA} \tag{2.19}$$

In most fault conditions, the generator's rotor angle will accelerate due to the sudden reduction of power demand. The acceleration rate of different generator differs depending on the electrical distance between generators and the fault location. In a WAN system with fault, the COA will accelerate slightly, while the generator near the fault will experience larger rotor acceleration rate. The relationship between COA and relative rotor angle of the generator near the fault is illustrated in Figure 2.7.

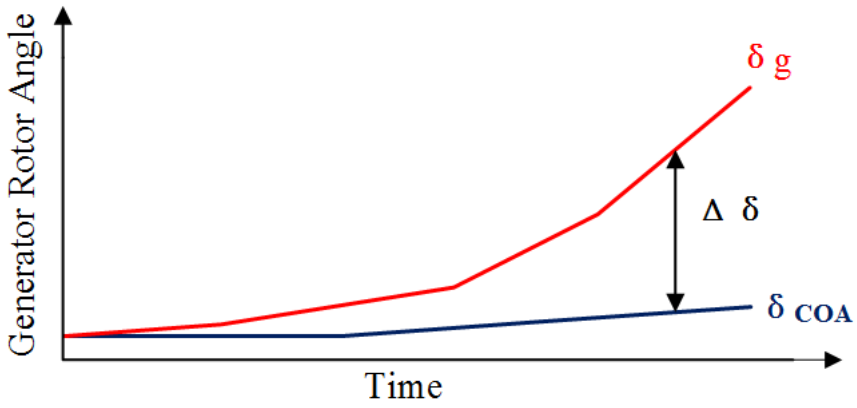

Figure 2.7: Typical Behavior of Generator Rotor Angle During A Fault

A threshold can be set as a criterion of system insecurity based on the rotor angle difference $\triangle\delta_j$. The threshold commonly used to check $\triangle\delta$ for stability is 60 degrees for accelerating conditions and -65 degrees for decelerating conditions [47].

Chapter 3

ON-LINE DSA SYSTEM DEVELOPMENT

This chapter describes step-by-step how to build an on-line DSA system. The development of an on-line DSA system has four tasks:

a. getting measurements records and running state estimation program;

b. initiating and running transient stability time-domain simulation;

c. checking system states to filter out insecure cases; and

d. realizing automation of the DSA system.

The computational procedure and corresponding software involved in each task of DSA systems are illustrated in Figure 3.1.

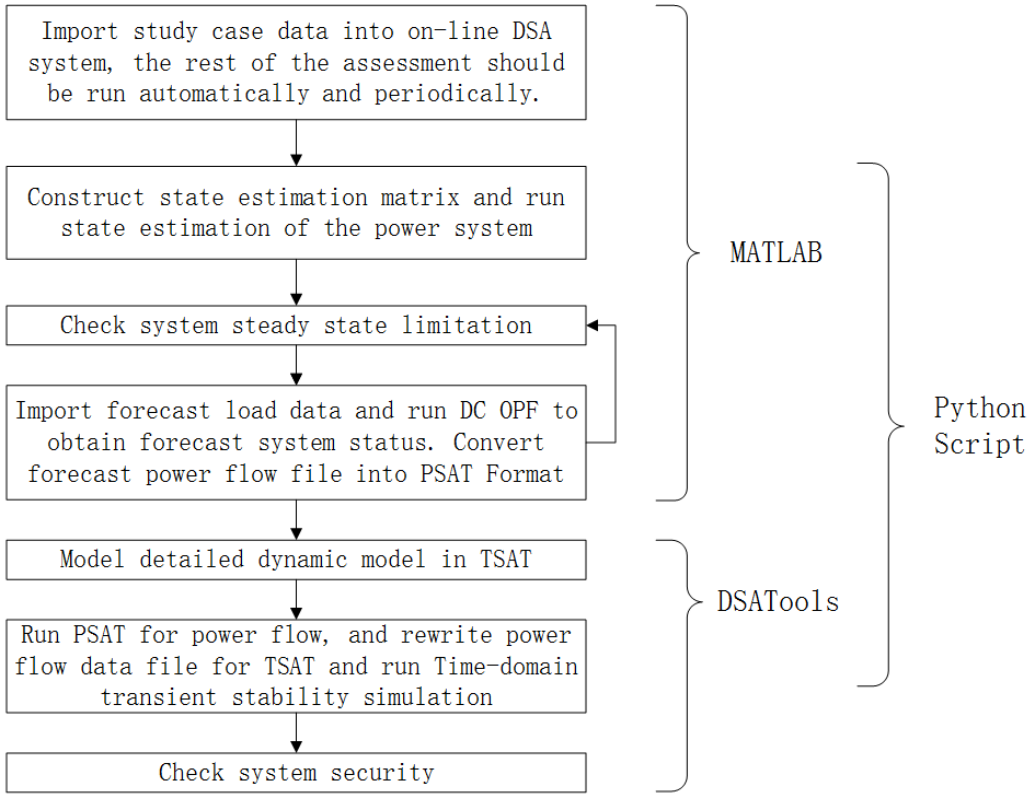

Figure 3.1: DSA Program Flow Chart

In this research, functions of state estimation and DC OPF are programmed in MATLAB 2016b. Some of the MATLAB code is based on the MATPOWER open source package. The transient stability analysis part is conducted by TSAT, which is one part of DSATools package. System security check and automation are realized based on Python 2.7. The main program of Python will call MATLAB function, call TSAT time-domain simulation, deal with simulation results,and prepare security reports. All the software is working on a 32-bit Windows 7 platform.

3.1 PMU-based State Estimation Program in Matlab

3.1.1 Introduction to MATPOWER Package

MATPOWER is an open source package coded in MATLAB M-files. The package provides solutions for power system network configuration, power flow and optimal power flow problems [48].

The format for power flow data in MATPOWER is mpc stored in a .mat file. A function "psse2mpc(casename)' is provided in MATPOWER to convert the PSS/E format .raw data file into MATPOWER format .mat data file. After converting the power flow file into mpc file, the number of buses n_b and branches n_l of the system can be obtained by using 'size()' command. Power system branch impedance, real and reactive power injection to a bus are available by using 'idx_bus' and 'idx_brch' commands.

The system $n_b \times n_b$ bus admittance matrix Y_{bus} relates the complex nodal current injection I_{bus} and the complex node voltage V:

$$I_{bus} = Y_{bus}V \tag{3.1}$$

Two $n_l \times n_b$ branch admittance matrices, Y_f and Y_t, are defined in MATPOWER. The two branch admittance matrices relate bus voltages to current injections at the

from bus and *to* bus respectively:

$$I_f = Y_f V_f \tag{3.2}$$

$$I_t = Y_t V_t \tag{3.3}$$

Taking the equivalent transmission line pi-model, as shown in Figure 2.5, as an example, assume that the series impedance is Z_{km}; the shunt admittance is $\frac{Y_{km}}{2}$. The current injection at node k is:

$$\begin{aligned} I_f &= \frac{V_k - V_m}{Z_{km}} + \frac{Y_{km}}{2} V_k \\ &= \left(\frac{1}{Z_{km}} + \frac{Y_{km}}{2}\right) V_k + \left(-\frac{1}{Z_{km}}\right) V_m \end{aligned} \tag{3.4}$$

$$\begin{aligned} I_t &= \frac{V_k - V_m}{Z_{km}} - \frac{Y_{km}}{2} V_k \\ &= \left(\frac{1}{Z_{km}} - \frac{Y_{km}}{2}\right) V_k + \left(-\frac{1}{Z_{km}}\right) V_m \end{aligned} \tag{3.5}$$

The system connection matrix indicates the connections between each branch and buses. C_f and C_t represents connections for *from* bus and *to* bus respectively. The rows of the connection matrix represents branches in the same order as defined in power flow data file. The column of the matrix represents buses from bus 1 to bus n_b.

3.1.2 PMU Placement

One important step before PMU-based state estimation is to place PMU properly in the system to obtain full observation. The method to place PMU was described in Section 2.4. The connection matrix A is obtained by searching non-zero entries in system admittance matrix Y_{bus}.

The optimization problem as formulated in (2.10)-(2.11) is a Mixed Integer Programming (MIP) problem, where the decision variables are constrained to be integer

values. A mixed-integer linear programming (MILP) solver *intlinprog* in MATLAB can be applied to deal with this problem.

3.1.3 PMU-based State Estimation

PMU-based state estimation is coded based on the linear state estimation method introduced in Section 2.3. The inputs of the program, in this research, are PMU measurement records of voltage phasors at the PMU-associated buses and branch current phasors along all branches connected to the associated bus.

Take the IEEE 9-bus system, as shown in Figure 2.6, as an example. The optimum PMU installation locations for the IEEE 9-bus system are at bus 4, bus 7 and bus 9, so the input of the linear state estimation program should be:

$$\left[\vec{V_4}, \vec{V_7}, \vec{V_9}, \vec{I_{45}}, \vec{I_{46}}, \vec{I_{75}}, \vec{I_{78}}, \vec{I_{96}}, \vec{I_{98}}, \vec{I_{41}}, \vec{I_{72}}, \vec{I_{93}}\right]^T \tag{3.6}$$

PMU placement matrix, X, and system branch-bus connection matrices, C_f and C_t, are used to construct the admittance matrix that indicates connections between PMU associated measurements and system states in MATLAB. The PMU placement matrix is the integer decision variable as described in the previous Subsection 3.1.2.

The admittance matrix can be divided into two parts. The upper part of the admittance matrix indicates relationship between voltage measurements and system states. It can be obtained by the following steps [44]:

a. defining an $n_b \times n_b$ identity matrix I,

b. taking out rows corresponding to the PMU placement matrix.

The lower part of the matrix represents the relationship between current measurements and system states. The procedure to obtain this part of matrix is:

a. adding C_f and C_t together to obtain system branch-bus connection matrix C;

b. determining rows to take for the next step by finding the location of non-zero

entries for CX, which is the row number of branches associated with at least one PMU;

c. taking out the rows determined in step b from system branch-*from*-bus.

After the determination of the two parts of the admittance matrix for linear state estimation, the system state can be calculated using pseudo-inverse function *pinv* provided by MATLAB.

The MATLAB code to obtain optimal PMU placement and to perform PMU-based state estimation is included in Appendix A

3.2 Time-domain Transient Stability Simulation in DSATools

TSAT is one part of the DSATools software package developed by Powertech Labs Inc. The DSATools software package includes four tools: PSAT, VSAT, TSAT, and SSAT. The on-line DSA system developed in PJM is based on this package [10].

TSAT is the tool specially used for transient stability simulation and study. It can perform detailed time-domain simulation for large and complex power system models. The process can be automated with Python Script interface.

To perform a transient stability simulation of a system, four types of data files are needed as inputs. They are: power flow data file, dynamic data file, contingency data file and monitor data file. One TSAT case can contain several scenarios. Different contingencies can be evaluated in different scenarios.

The TSAT case is stored in a file with the file extension type .tsa. One can create a TSAT case through the human-machine interface, or by creating a .tsa file and including all the input data files into it.

3.3 Automation of DSA system in Python

An interface between MATLAB and DSATools (PSAT, TSAT) have been developed in this research using Python Script. Since DSATools is a 32-bit version software, to avoid compiling problems, the Python version interfacing DSATools should be 32-bit. However, the MATLAB is a 64-bit version, so the Python version for MATLAB should be 64-bit. The computational flow chart of the automation process in Python is shown in Figure 3.2.

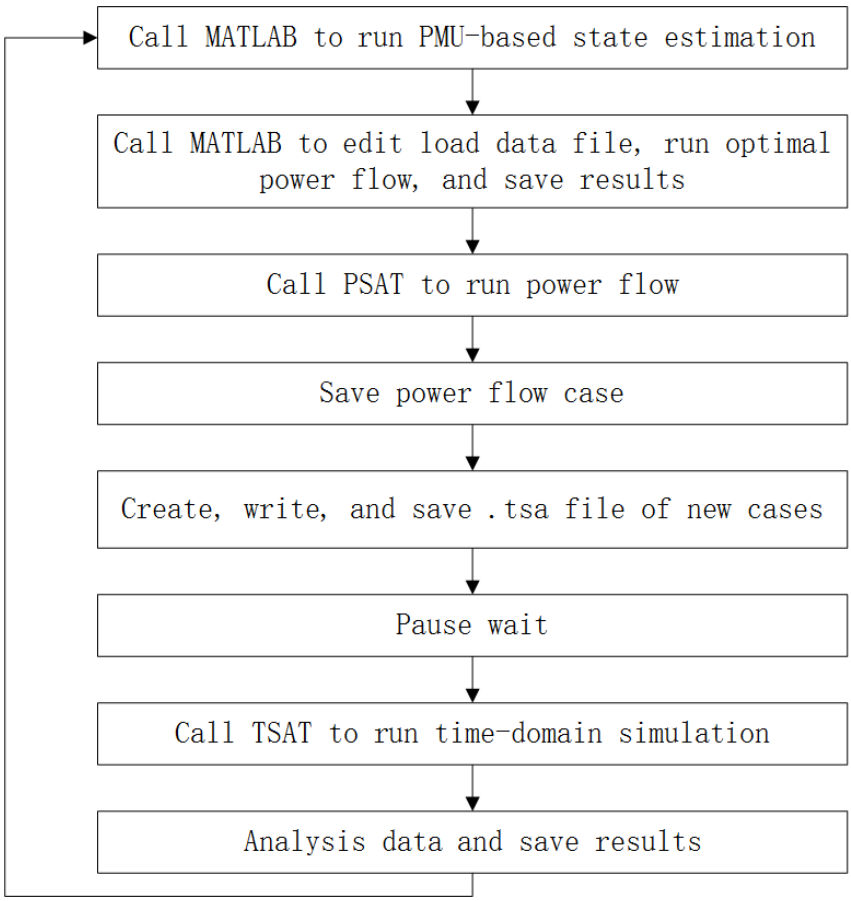

Figure 3.2: Python Automation Flowchart

Python will first call MATLAB to run state estimation. Then Python will update forecast load and call MATLAB to run optimal power flow. The generation information form OPF is exported from MATLAB and be updated in the .raw power flow file

by Python. The prepared raw power flow file is PTI format. Python will call PSAT to run power flow again and save the power flow results as .pfb format. Then Python will create, edit, and save .tsa file for transient stability simulation. A pause wait is added here because editing and saving .tsa file need a small amount of time. After this, Python will call TSAT to run the simulation based on the .tsa file created. The results are then analyzed to determine system stability.

The code for program automation is included in Appendix B

3.3.1 Call Matlab Function

MATLAB provides an MATLAB Engine API package for Python to call MATLAB as a computational engine. To import MATLAB engine, it needs to be registered using *cmd* as administrator. The commands to register MATLAB engine are:

```
cd C:\path direction\R2015a\extern\engines\python
setup.py install
```

After installing and registering MATLAB engine, Python can call a MATLAB function using the following commands:

```
eng=matlab.engine.start_matlab()
eng.addpath(r'file direction',nargout=0)
```

3.3.2 Call PSAT and TSAT Function

PSAT provides a Python script environment to automate the process of power flow study. Commands are provided to edit system configuration as well as to evaluate results. The commands to run PSAT program should be saved in a Python file.

Python can call PSAT and TSAT to run using Windows *subprocess* package. The

commands for PSAT to run Python automation file without PSAT window popping up are:

```
import subprocess
startupinfo = subprocess. STARTUPINFO()
startupinfo.dwFlags = subprocess.STARTF_USESHOWWINDOW
startupinfo.wShowWindow = subprocess.SW_SHOWMINNOACTIVE
p = subprocess.Popen('PSAT_Path python_automation_script_path python '.sta
p.wail()
```

A batch file is provided in TSAT to enable external running of transient stability simulation. To call TSAT to run the batch file, subprocess package is used as following:

```
import subprocess
p = subprocess.Popen('file_direction/Tsat/bin/tsat_batch_'.
    tsa_file_direction', stdin=subproces.PIPE, stdout=
    supprocess.PIPE, shell=True)
p.wail()
```

The simulation result of TSAT is stored in a .dll file. To get access to the .dll file, it should be registered using *cmd* with administration:

```
regsvr32 D:\PATHNAME\Tsat\bin\ResultScript.dll
```

The results can be obtained in Python with help of win32com.client package. The command to read the results are as following:

```
import win32com.client
reader = win32com.client.Dispatch("ResultScript.BinReader")
```

3.3.3 Editing File Data

Creating and editing a new file is relatively easy in Python. To add line in the file, an easy way is to save all lines of the file in an array *filelines*. Then, use the following command to insert the lines into a plain file.

```
file = open(filepath, 'w')
file.writelines(filelines)
file.close()
```

The way to change elements in a written file is to read the file, make the modifications, and write it out to a new file. This procedure is used when updating the real and reactive power of the load file.

The code to change load data in MATLAB and Python is included in Appendix D and Appendix E.

Chapter 4

CASE STUDY: IEEE 118-BUS SYSTEM

4.1 IEEE 118-bus Test System

To further understand the benefits of the proposed DSA system, performance of both the existing and the proposed DSA systems were examined on the IEEE 118-bus test system, as shown in Figure 4.1. The 118-bus system is a simplified model of the American Electric Power (AEP) System as of December 1962 [49]. The system contains 19 generators, 35 synchronous condensers, 177 lines, 9 transformers, and 91 loads. The static model and dynamic model of the system will be described below.

4.1.1 Static modeling

The test power flow row data for IEEE 118-bus system is available on the Illinois Center for a Smarter Electric Grid (ICSEG) Power Cases website [49]. The generator cost data, for calculating DC OPF, is based on the cost data of the 118-bus system in Matpower package.

Assume that the actual net load of the 118-bus system follows the same feature as the estimated net load for 2020 from the CAISO "Duck Curve", shown in Figure 4.2. This means the actual load of the IEEE 118-bus system is scaled by the same factor as the "Duck Curve" in half hour intervals. Half-hourly load factor data of the "Duck Curve" are shown in Table 4.1. Note that net load of the system will experience a steep ramp between 5 pm to 10:30 pm and decrease quickly after 10:30 pm.

The proposed DSA system will predict a forecast operating point by running DC OPF with forecast load data. Since ideal load forecast is assumed, the forecast

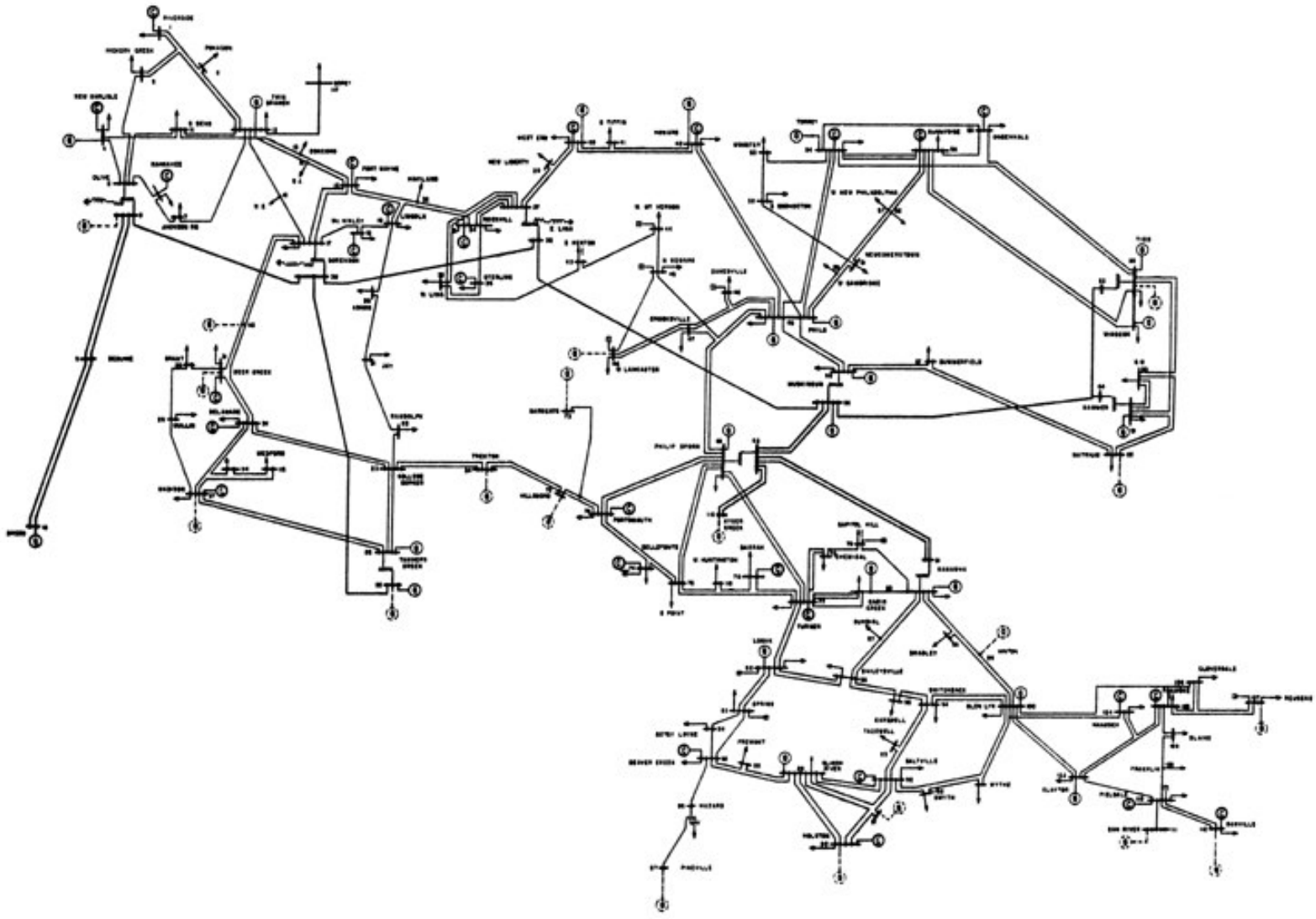

Figure 4.1: IEEE 118-bus Test System Configuration [49]

Table 4.1: Scaled Estimated Net Load Data for 2020 CAISO "Duck Curve"

Time	Scale Factor	Time	Scale Factor
12 am	1.73	12:00 pm	1.05
12:30 am	1.67	12:30 pm	1.00
1:00 am	1.64	1:00 pm	0.99
1:30 am	1.60	1:30 pm	0.97
2:00 am	1.57	2:00 pm	0.97
2:30 am	1.54	2:30 pm	0.97
3:00 am	1.53	3:00 pm	0.98
3:30 am	1.51	3:30 pm	1.00
4:00 am	1.52	4:00 pm	1.03
4:30 am	1.53	4:30 pm	1.06
5:00 am	1.54	5:00 pm	1.14
5:30 am	1.54	5:30 pm	1.23
6:00 am	1.55	6:00 pm	1.43
6:30 am	1.56	6:30 pm	1.62
7:00 am	1.57	7:00 pm	1.81
7:30 am	1.58	7:30 pm	1.99
8:00 am	1.56	8:00 pm	2.06
8:30 am	1.55	8:30 pm	2.12
9:00 am	1.46	9:00 pm	2.09
9:30 am	1.37	9:30 pm	2.05
10:00 am	1.29	10:00 pm	1.99
10:30 am	1.20	10:30 pm	1.93
11:00 am	1.15	11:00 pm	1.86
11:30 am	1.09	11:30 pm	1.78

*Source: Energy Storage and the California "Duck Curve" [50]

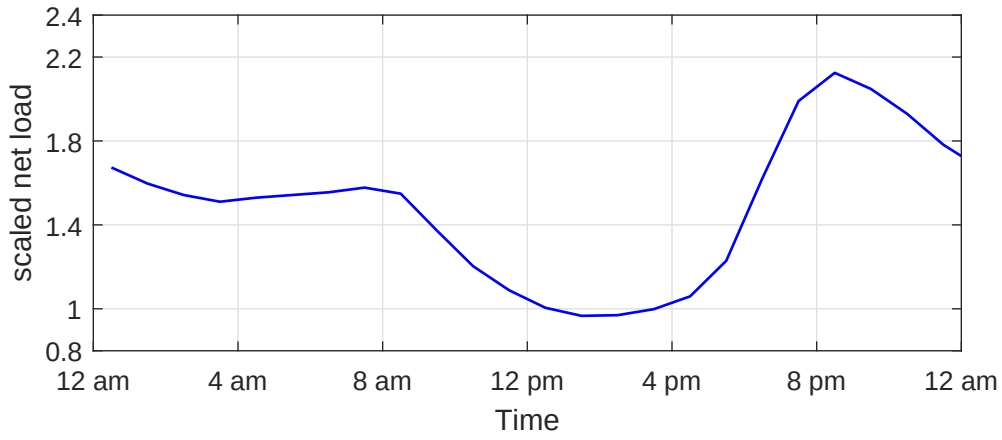

Figure 4.2: Estimated Net Load Curve for 2020 from CAISO "Duck Curve"

load data are the same as the actual load half an hour later. To eliminate impacts introduced by other factors, the present working point is also calculated by DC OPF for the existing DSA system but with input data of present actual load data.

4.1.2 Dynamic modeling

In the dynamic model of the 118-bus system, all generators, exciters, and governors were built with detailed dynamic models. The parameters of generators, exciters, and governors are based on the data listed in Table 4.2, Table 4.3, and Table 4.4, respectively. Based on the parameters given in those tables, the model selected for generator is *'DG0S4'*; for exciter is *'EXC1*, for governor is *GOV8*. The dynamic file used for case study is included in Appendix F.

In the contingency file, the base scenario evaluates the system without fault, while sub-scenarios run simulations for three-phase faults on buses and line sections. Only three-phase faults are evaluated in this research because these faults are the most severe and always cause the worst operating condition in a system.

Simulation time is 10 seconds for a three-phase bus fault, which will exist for 5 cycles. Simulation time is 16 seconds for a three-phase line fault which is placed

Table 4.2: IEEE 118-bus Modified Test Generator Data

Parameter	Sync. Gen	Sync. Gen	Sync. Gen	Sync. Gen	Sync. Gen	Sync. Gen	Sync. Gen	Sync. Gen	Sync. Gen
Unit No.	10 69 80	12	25 49 100	31 26 87	54 46 111	59 103	65 61	66	89
Rated Power (MVA)	590	125	330	410	75	100	233	512	835
Rated Voltage (kV)	22	15.5	24	13.8	13.8	20	24	20	
H(s)	2.319	4.748	3.006	3.704	6.187	4.985	4.122	2.631	2.6419
D	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000
r_a(p.u)	0.0046	0.004	0.000	0.0019	0.0031	0.0035	0.0016	0.004	0.0019
x_d(p.u)	2.110	1.220	1.950	1.7668	1.050	1.180	1.569	1.700	2.183
x_q(p.u)	2.020	1.160	1.920	1.7469	0.980	1.050	1.548	1.650	2.157
x'_d(p.u)	0.280	0.174	0.317	0.2738	0.185	0.220	0.324	0.270	0.413
x'_q(p.u)	0.490	0.250	1.120	1.0104	0.360	0.380	0.918	0.470	1.285
x''_d(p.u)	0.215	0.134	0.200	0.2284	0.130	0.145	0.249	0.200	0.339
x''_q(p.u)	0.215	0.134	0.200	0.2284	0.130	0.145	0.249	0.200	0.339
x_l(p.u)	0.155	0.0078	0.199	0.1834	0.070	0.075	0.204	0.160	0.246
T'_{d0}(p.u)	0.5573	8.970	0.9754	0.8418	1.0748	1.100	1.0614	0.6035	5.690
T'_{q0}(p.u)	0.1371	0.500	0.875	0.8676	0.1102	0.1086	0.8895	0.1367	1.500
T''_{d0}(p.u)	0.0246	0.033	0.0473	0.035	0.0267	0.0277	0.0336	0.0556	0.041
T''_{q0}(p.u)	0.0272	0.070	0.0134	0.035	0.0358	0.0351	0.0381	0.0319	0.144
$S(1.0)$(p.u)	0.079	0.1026	0.082	0.2632	0.100	0.0933	0.0987	0.090	0.134
$S(1.2)$(p.u)	0.349	0.432	0.290	0.5351	0.3928	0.4044	0.303	0.400	0.617

Table 4.3: IEEE 118-bus Modified Test Exciter Data

Parameter	Sync. Gen	Sync. Gen	Sync. Gen	Sync. Gen	Sync. Gen	Sync. Gen	Sync. Gen	Sync. Gen	Sync. Gen
Unit No.	10 69 80	12	25 49 100	31 26 87	54 46 111	59 103	65 61	66	89
Rated Power (MVA)	590	125	330	410	75	100	233	512	835
Rated Voltage (kV)	22	15.5	24	13.8	13.8	20	24	20	
T_r(s)	0.000	0.060	0.000	0.000	0.000	0.060	0.000	0.000	0.000
K_a(p.u.)	200	25	400	400	0.050	25	250	200	400
T_a(s)	0.3575	0.200	0.050	0.020	20.000	0.200	0.060	0.395	0.020
V_{Rmax}(p.u)	5.730	1.000	3.810	5.270	4.380	1.000	4.420	3.840	18.300
V_{Rmin}(p.u)	-5.730	-1.000	-3.810	-5.270	0.000	-1.000	-4.420	-3.840	-18.300
K_e(p.u)	1.000	-0.0601	-0.170	1.000	1.000	-0.0582	1.000	1.000	1.000
T_e(s)	0.011	0.6758	0.950	0.920	1.980	0.6544	0.613	0.008	0.942
K_f(p.u)	0.0529	0.108	0.040	0.030	0.000	0.105	0.053	0.0635	0.030
T_f	1.000	0.350	1.000	1.000	0.100	0.350	0.330	1.000	1.000
E_1	4.2975	2.4975	3.6675	2.4675	2.385	2.5785	2.610	2.880	3.765
$SE(E_1)$	0.000	0.0949	0.0111	0.4351	0.0951	0.0889	0.000	0.000	0.8147
E_2(p.u)	5.730	3.330	4.890	3.290	3.180	3.438	3.480	3.840	5.020
$SE(E_2)$(p.u)	0.000	0.37026	0.0178	0.6001	0.3712	0.3468	0.000	0.000	2.6756

Table 4.4: IEEE 118-bus Modified Test Governor Data

Parameter	Sync. Gen	Sync. Gen	Sync. Gen	Sync. Gen	Sync. Gen	Sync. Gen	Sync. Gen	Sync. Gen	Sync. Gen
Unit No.	10 69 80	12	25 49 100	31 26 87	54 46 111	59 103	65 61	66	89
Rated Power (MVA)	590	125	330	410	75	100	233	512	835
Rated Voltage (kV)	22	15.5	24	13.8	13.8	20	24	20	
P_{max}(p.u.)	0.9372	1.056	1.050	0.8951	1.000	1.050	0.901	0.898	0.9177
R(p.u.)	0.0085	0.040	0.0152	0.0122	0.066	0.050	0.0214	0.0098	0.006
T_1(s)	0.080	0.083	0.100	0.180	0.090	0.090	0.150	0.150	0.180
T_2(s)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.050	0.030
T_3(s)	0.150	0.200	0.400	0.040	0.200	0.200	0.100	0.300	0.200
T_4(s)	0.050	0.050	0.050	0.250	0.300	0.300	0.300	0.260	0.000
T_5(s)	10.000	5.000	8.000	8.000	0.000	0.000	10.000	8.000	8.000
F(s)	0.280	0.280	0.250	0.267	1.000	1.000	0.237	0.270	0.300

at 5% electrical distance away from each end bus. Because each line connects two end buses, in this research, it is defined that faults are placed near the *from* buses. Faults will be first cleared at 5 cycles at the near end bus. After one additional cycle, the fault line will be cut off completely at the far end bus [51]. The Python file to generate the contingency file is attached in Appendix G.

The monitor data are set for the base scenario. All the sub-scenarios, simulating system with fault conditions, use the same monitor data as the base scenario. Because rotor angle and COA are the variables to determine system transient stability, as described in Section 2.6, rotor angles for each generator are monitored, while all the buses are defined as in one region. The monitor data file is attached in Appendix H.

4.2 Simulation Results

The process for time-domain simulation and data analysis takes the DSA system around 5 minutes on average to assess the 118-bus system with 473 contingencies. DSA systems generate security reports with plots of insecure cases. Operators can refer these security report to make operation decisions.

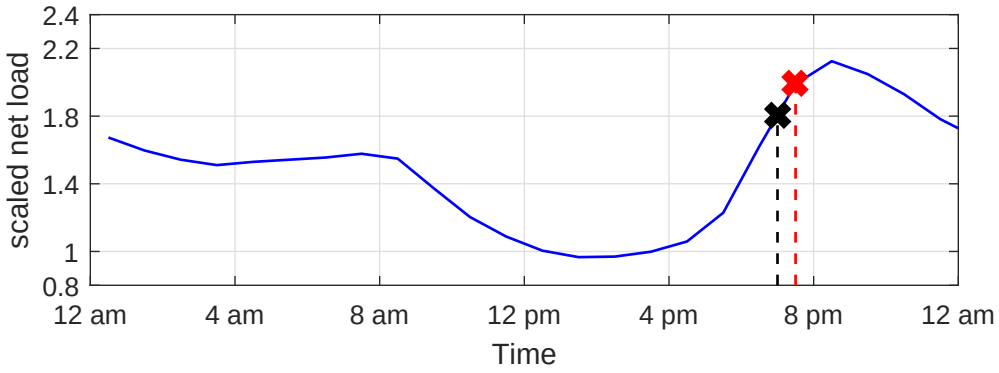

Figure 4.3: Illustration for The Study Time

Figure 4.3 illustrates the working point for a case study. Suppose that present operating point is 7:00 pm, as shown in the black *"X"*. We need to evaluate the system

performance for 7:30 pm, which is represented by the red *"X"*. The proposed DSA system uses forecast operating point at 7:30 pm to perform the transient stability simulation. The existing DSA system uses the present operating point at 7:00 pm to initialize the simulation.

Figures 4.4 and 4.5 show buses with voltage below 0.95 at 7:00 pm and 7:30 pm, respectively. A total of 16 buses experience voltage drop larger than 5% at 7:00 pm, while 32 buses at 7:30 pm. The significant difference of voltage drop between the two operating points indicates the necessity of forecast operating points for transient stability simulation.

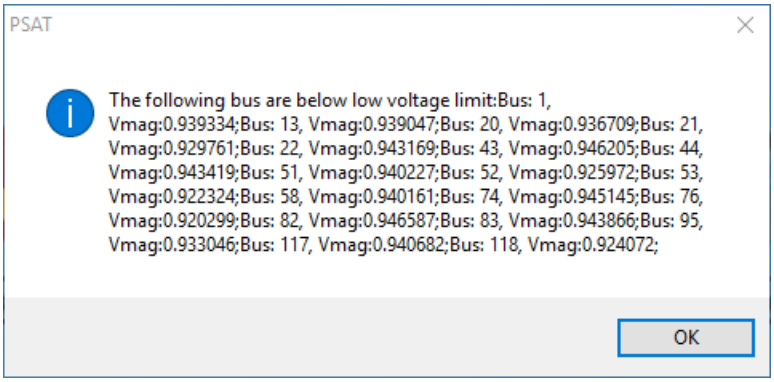

Figure 4.4: Bus Voltage Violation at 7:00 pm

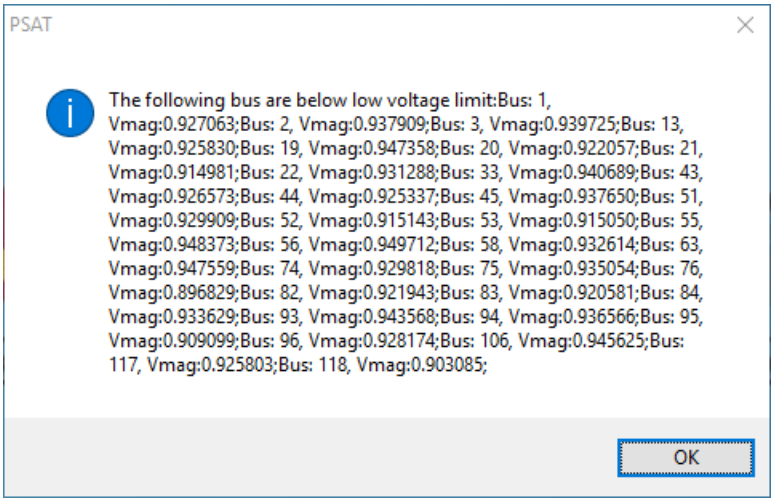

Figure 4.5: Bus Voltage Violation at 7:30 pm

Figures 4.6 and 4.7 show two of the endangerments predicted by the proposed dynamic security assessment system for 7:30 pm. The 118-bus system will face cascading failure if a fault occurs on bus 5 or on 5% of the line from bus 5 to bus 11. However, if the system is evaluated by the existing DSA system, endangerments

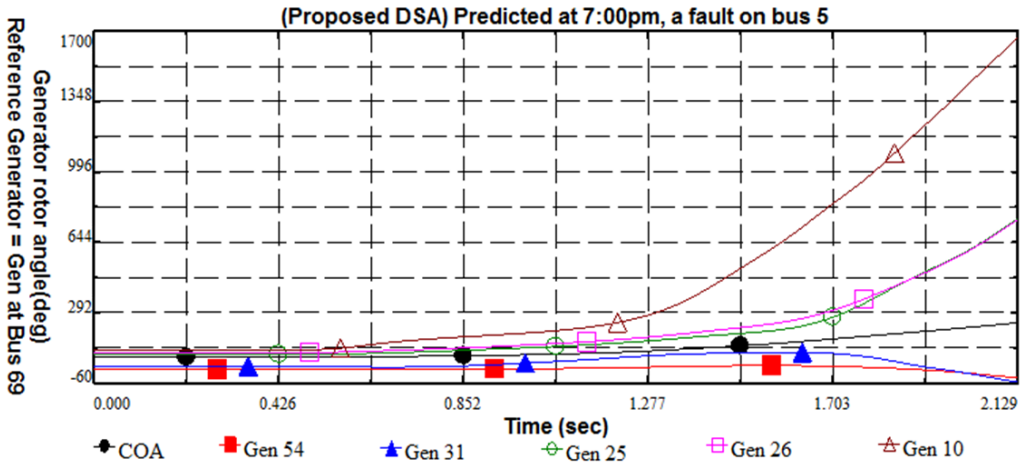

Figure 4.6: Generator Rotor Angle During A Fault on Bus 5 Predicted by Proposed DSA System

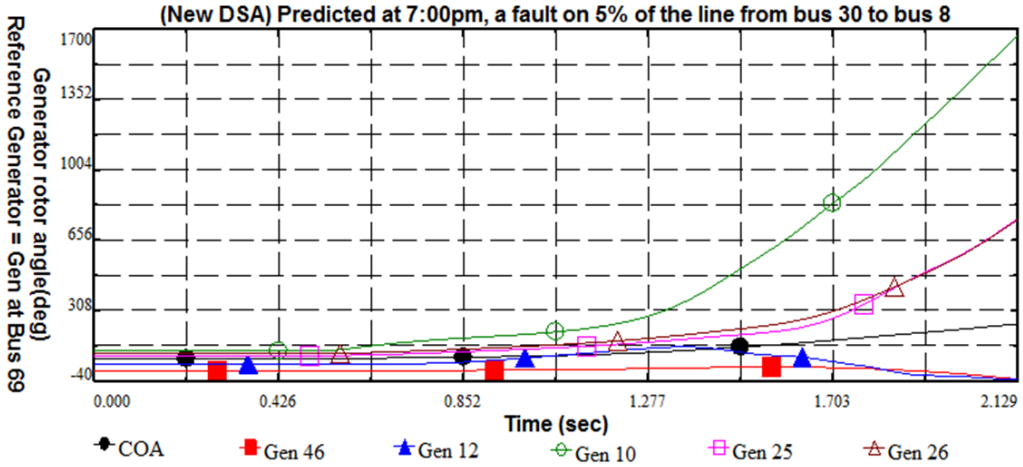

Figure 4.7: Generator Rotor Angle During A Fault on 5% of the Line from Bus 30 to Bus 8 Predicted by Proposed DSA System

shown in Figures 4.6 and 4.7 will not be predicted. Simulation results for the existing DSA system are shown in Figures 4.8 and 4.9 because a present working point with

load factor 1.81 will be used instead of 1.99. It indicates that some endangerments will not be predicted by the existing DSA systems on time.

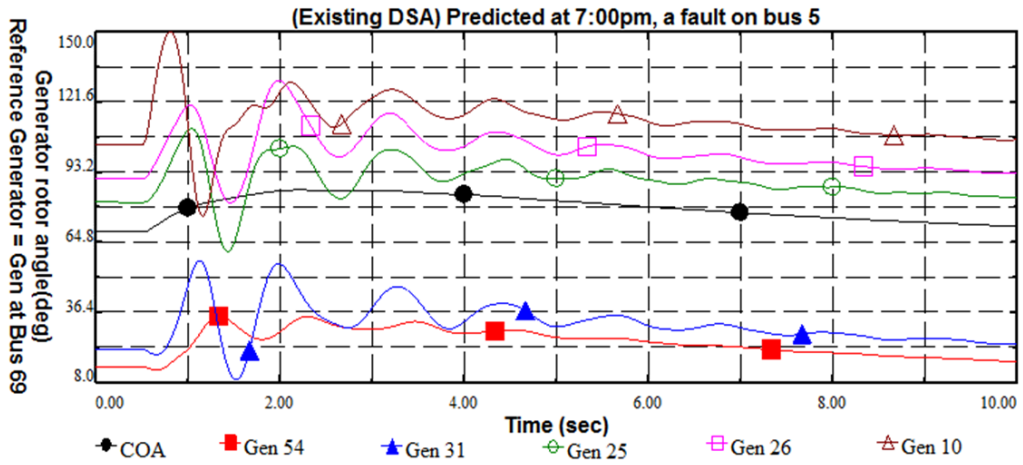

Figure 4.8: Generator Rotor Angle During A Fault on Bus 5 Predicted by Existing DSA System

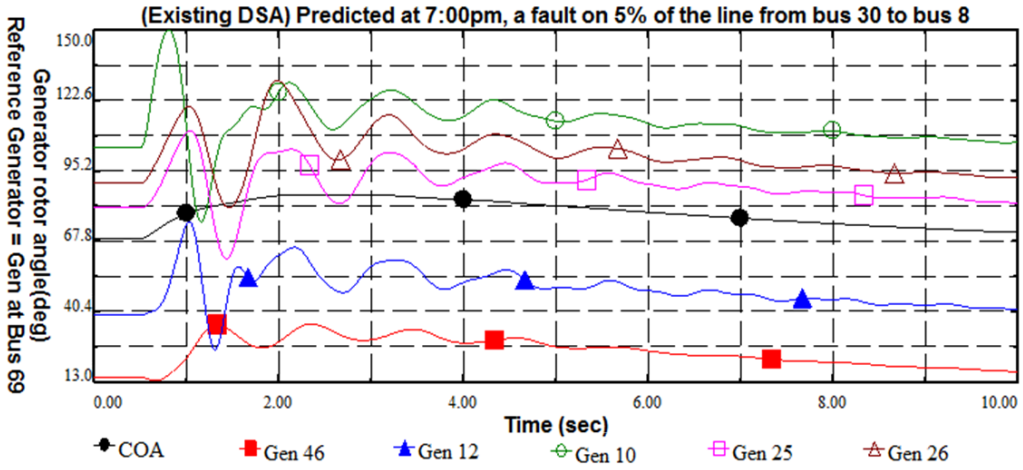

Figure 4.9: Generator Rotor Angle During A Fault on 5% of the Line from Bus 30 to Bus 8 Predicted by Existing DSA System

Table 4.5 compares performance between two DSA systems for three net load patterns, including maintain, decrease, and increase. DSA systems start the evaluation process from present time.

Table 4.5: Comparison of the Number of Insecure Cases Between the Existing and Proposed DSA System

Current time	DSA System	Load Factor	No. of insecure scenario
7:00 am	Existing	1.57	18
	New	1.58	18
	Actual	1.58	18
8:30 am	Existing	1.55	18
	New	1.46	16
	Actual	1.46	16
12:00 pm	Existing	1.05	14
	New	1.00	12
	Actual	1.00	12
5:30 pm	Existing	1.23	14
	New	1.43	16
	Actual	1.43	16
8:00 pm	Existing	2.06	32
	New	2.12	41
	Actual	2.12	41
9:30 pm	Existing	2.05	32
	New	1.99	26
	Actual	1.99	26

The performance of the two DSA systems has no significant difference when net load changes a little, as about 7:00 am. However, the number of predicted insecure scenarios differs significantly when load decreases or increases quickly. For example, if the existing DSA system uses the operating point at 8:00 pm to predict endangerment at 8:30 pm, it will ignore 9 insecure scenarios, which is around 22% of the actual potential risks. In addition, although the load decreases by only 3% from 9:30 pm to 10 pm, the existing DSA system will predict 6 more insecure scenarios than it should for 10:00 pm, which is about 23% of the total actual potential risks.

Because ideal load prediction was assumed for the proposed DSA system, the prediction resulting from it should be the same as the actual system conditions. The differences of predictions between the two DSA systems demonstrate that the existing DSA system is less accurate because it fails to take into consideration load changes in advance.

Table 4.7 summarizes the simulation results of the 118-bus system for 14 different load scale factors. The bus numbers are listed for buses with voltages below 0.95 per unit. The fault scenarios are also listed when one or more generator loss synchronous.

Table 4.7: Summary of Insecure Fault Cases for IEEE 118-bus System.

Scale Factor	Buses Below low Voltage Limit	Faults Causing system Insecurity
1.73	bus 1, 13, 20 21, 22, 44 51, 52, 53, 58, 74, 76, 83, 95, 117, 118	Fault on bus 8, bus 9, bus 10, bus 69, bus 8 → bus 9, bus 8 → bus 30, bus 9 → bus 10, bus 26 → bus 30, bus 69 → bus 70, bus 69 → bus 75, bus 69 → bus 77, bus 85 → bus 86, bus 9 → bus 8, bus 10 → bus 9,bus 25 → bus 23, bus 30 → bus 26 , bus 86 → bus 85, bus 87 → bus 86, bus 86 → bus 87, bus 110 → bus 111, bus 111 → bus 110
1.64	bus 1, 13, 20, 21, 22, 51, 52, 53, 58, 76,95, 118	Fault on bus 8, bus 9, bus 10, bus 69, bus 8 → bus 9, bus 8 → bus 30, bus 9 → bus 10 , bus 26 → bus 30, bus 69 → bus 70, bus 69 → bus 75, bus 69 → bus 77, bus 85 → bus 86, bus 86 → bus 87, bus 9 → bus 8, bus 10 → bus 9, bus 30 → bus 26 , bus 86 → bus 85,

continued on next page

continued from previous page

Scale Factor	Buses Below low Voltage Limit	Faults Causing system Insecurity
		bus 87 → bus 86, bus 110 → bus 111, bus 111 → bus 110
1.60	bus 1, 13, 20, 21, 51, 52, 53, 76, 95, 118	Fault on bus 8, bus 9, bus 10, bus 69, bus 8 → bus 9, bus 8 → bus 30, bus 9 → bus 10 , bus 26 → bus 30, bus 69 → bus 75, bus 69 → bus 77, bus 85 → bus 86, bus 86 → bus 87, bus 110 → bus 111, bus 9 → bus 8, bus 10 → bus 9, bus 30 → bus 26 , bus 86 → bus 85, bus 87 → bus 86, bus 111 → bus 110
1.54	bus 20, 21, 52, 53, 58, 76, 118	Fault on bus 8, bus 9, bus 10, bus 69, bus 8 → bus 9, bus 8 → bus 30, bus 9 → bus 10 , bus 26 → bus 30, bus 69 → bus 77, bus 85 → bus 86, bus 86 → bus 87, bus 110 → bus 111, bus 9 → bus 8, bus 10 → bus 9, bus 30 → bus 26, bus 86 → bus 85, bus 87 → bus 86, bus 111 → bus 110
1.46	bus 20, 21, 52, 53, 76, 118	Fault on bus 9, bus 10, bus 8 → bus 9, bus 8 → bus 30, bus 9 → bus 10 , bus 26 → bus 30, bus 69 → bus 77, bus 85 → bus 86, bus 86 → bus 87, bus 110 → bus 111, bus 9 → bus 8, bus 10 → bus 9, bus 30 → bus 26, bus 86 → bus 85, bus 87 → bus 86, bus 111 → bus 110
1.37	bus 20, 21, 52, 53, 76, 118	Fault on bus 9, bus 10, bus 8 → bus 9, bus 8 → bus 30, bus 9 → bus 10 , bus 26 → bus 30, bus 85 → bus 86, bus 86 → bus 87,bus 111 → bus 110,

continued on next page

continued from previous page

Scale Factor	Buses Below low Voltage Limit	Faults Causing system Insecurity
		bus 110 → bus 111, bus 9 → bus 8, bus 10 → bus 9, bus 30 → bus 26, bus 86 → bus 85, bus 87 → bus 86,
1.29	bus 21, 52, 53, 76, 118	Fault on bus 9, bus 10, bus 8 → bus 9, bus 9 → bus 10 , bus 26 → bus 30, bus 85 → bus 86, bus 86 → bus 87, bus 110 → bus 111, bus 9 → bus 8, bus 10 → bus 9, bus 30 → bus 26, bus 86 → bus 85, bus 87 → bus 86, bus 111 → bus 110,
1.15	bus 53, 76, 118	Fault on bus 9, bus 10, bus 8 → bus 9, bus 9 → bus 10 , bus 26 → bus 30, bus 85 → bus 86, bus 86 → bus 87, bus 110 → bus 111, bus 9 → bus 8, bus 10 → bus 9, bus 30 → bus 26, bus 86 → bus 85, bus 87 → bus 86, bus 111 → bus 110
1.0	bus 53, 76, 118	Fault on bus 10, bus 8 → bus 9, bus 9 → bus 10, bus 26 → bus 30, bus 85 → bus 86, bus 86 → bus 87, bus 110 → bus 111, bus 9 → bus 8, bus 10 → bus 9, bus 86 → bus 85, bus 87 → bus 86, bus 111 → bus 110
1.81	bus 1, 13, 20, 21, 22, 43, 44, 51, 52, 53, 58, 74, 76, 82, 86, 83, 95, 117,	Fault on bus 8, bus 9, bus 10, bus 69, bus 8 → bus 9, bus 8 → bus 30, bus 9 → bus 10 , bus 26 → bus 30, bus 68 → bus 81, bus 69 → bus 70, bus 69 → bus 75, bus 85 → bus 86, bus 86 → bus 87, bus 110 → bus 111, bus 9 → bus 8, bus 10 → bus 9, bus 25 → bus 23, bus 30 → bus 26 ,

continued on next page

continued from previous page

Scale Factor	Buses Below low Voltage Limit	Faults Causing system Insecurity
	118	bus 86 → bus 85, bus 87 → bus 86, bus 111 → bus 110
1.99	bus 1, 2, 3, 13, 19, 20, 21, 22, 33, 43, 44, 45, 51, 52, 53, 55, 56, 58, 63, 74, 75, 76, 82, 83, 84, 93, 94, 95, 96, 106, 117, 118	Fault on bus 5, bus 8, bus 9, bus 10, bus 29, bus 4 → bus 5, bus 5 → bus 6, bus 5 → bus 11, bus 8 → bus 9, bus 8 → bus 30, bus 9 → bus 10 , bus 25 → bus 27, bus 26 → bus 30, bus 85 → bus 86, bus 86 → bus 87, bus 110 → bus 111, bus 5 → bus 4, bus 9 → bus 8,bus 30 → bus 8, bus 10 → bus 9, bus 25 → bus 23, bus 30 → bus 26 , bus 86 → bus 85, bus 87 → bus 86, bus 89 → bus 88, bus 111 → bus 110
2.05	bus 1, 2, 3, 13, 15,16, 19, 20, 21, 22, 33, 38, 43, 44, 45, 51, 52, 53, 55, 56, 58, 63, 74, 75, 76,82, 83, 84, 93, 94, 95, 96, 106, 117,118	Fault on bus 4, bus 5, bus 8, bus 9, bus 10, bus 25, bus 26, bus 30, bus 4 → bus 5, bus 4 → bus 11, bus 5 → bus 6, bus 5 → bus 11, bus 8 → bus 9, bus 8 → bus 30, bus 9 → bus 10, bus 25 → bus 27, bus 30 → bus 38, bus 85 → bus 86, bus 86 → bus 87, bus 110 → bus 111, bus 5 → bus 3, bus 5 → bus 4, bus 9 → bus 8, bus 30 → bus 8, bus 10 → bus 9, bus 25 → bus 23, bus 30 → bus 26 , bus 38 → bus 30, bus 86 → bus 85, bus 87 → bus 86, bus 89 → bus 88, bus 111 → bus 110
	bus 1, 2, 3, 13, 14, 15, 16, 18, 19, 20,21, 22, 33, 38,	Fault on bus 4, bus 5, bus 8, bus 9, bus 10, bus 25, bus 26, bus 30, bus 4 → bus 5, bus 12 → bus 11, bus 4 → bus 11, bus 5 → bus 6, bus 5 → bus 11,

continued on next page

continued from previous page		
Scale Factor	Buses Below low Voltage Limit	Faults Causing system Insecurity
2.12	39, 43, 44, 45, 51, 52, 53, 55, 56, 57, 58, 63, 74, 75, 76, 82, 83, 84, 86, 93, 94, 95, 96, 106, 117, 118	bus 8 → bus 9, bus 8 → bus 30, bus 9 → bus 10, bus 11 → bus 12, bus 23 → bus 25, bus 25 → bus 27, bus 26 → bus 30, bus 30 → bus 38,bus 80 → bus 97, bus 80 → bus 98, bus 80 → bus 99, bus 85 → bus 86, bus 86 → bus 87, bus 5 → bus 3, bus 5 → bus 4, bus 9 → bus 8, bus 30 → bus 8, bus 10 → bus 9, bus 25 → bus 23, bus 30 → bus 26 , bus 38 → bus 30, bus 65 → bus 38, bus 75 → bus 69, bus 77 → bus 69, bus 80 → bus 77, bus 86 → bus 85, bus 87 → bus 86, bus 89 → bus 88,
1.93	bus 1, 2, 3, 13, 20, 21, 22, 33, 43, 44, 45, 51, 52, 53, 55, 58,63, 74, 75, 76, 82, 83, 84, 93, 94, 95, 96, 106, 117, 118	Fault on bus 8, bus 9, bus 10, bus 68, bus 8 → bus 9, bus 8 → bus 30, bus 9 → bus 10 , bus 26 → bus 30, bus 30 → bus 38, bus 68 → bus 116, bus 85 → bus 86, bus 86 → bus 87, bus 110 → bus 111, bus 9 → bus 8, bus 30 → bus 8, bus 10 → bus 9, bus 25 → bus 23, bus 30 → bus 26 , bus 86 → bus 85, bus 87 → bus 86, bus 111 → bus 110

4.3 Error Analysis

In practice, load forecast for power system is not ideal and is always with prediction error. Therefore, it is important to evaluate the performance of the new DSA system using load forecast data with error. The average daily mean absolute percentage error(DMAPE) for the short-term load forecast of PJM Electricity Markets in the United States is less than 1.61% [43]. Assume that the worst case of load forecast error is 2%. The ability for the new DSA system to predict possible insecurity is evaluated for the case with -2% load forecast error.

Simulation results shows that with -2% load forecast error, the new DSA system can still predict the insecure case on 5% of the line from bus 30 to bus 8 as shown in Figure 4.10. This endangerment can not be predicted by the existing DSA system as shown in Figure 4.9.

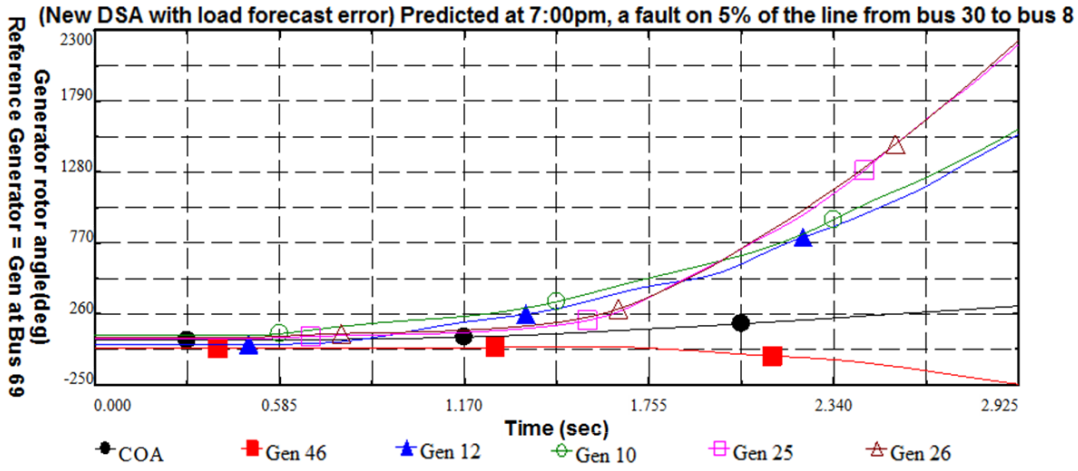

Figure 4.10: Contingency Evaluated by the New DSA System with -2% Error

With -2% load forecast error, the new DSA system can provide lower security prediction error than the existing DSA system for the 118-bus system. The comparison results are summarized in Table 4.8.

Table 4.8: Comparison of the Prediction Error Between the Existing and Proposed DSA System

Error	New DSA System with -2% error for forecast load	Existing DSA System
Root Mean Square Error	12.65	19.98
Maximum Error	-78.95%	-115.79%

The new on-line DSA system gives better prediction than the existing DSA system even with -2% load forecast error for each load pattern as demonstrated in Table 4.9.

Table 4.9: Comparison of the Number of Insecure Cases Between the Existing and Proposed DSA System

Current time	DSA System	Load Factor	No. of insecure scenario
7:00 am	Existing	1.57	18
	New	1.55	18
	Actual	1.58	18
8:00 pm	Existing	2.06	32
	New	2.08	40
	Actual	2.12	41
9:30 pm	Existing	2.05	32
	New	1.95	22
	Actual	1.99	26

Chapter 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

DSA systems are developed to help predict as many risks that might occur. With the information of the potential risk and its corresponding possibilities, operators are able to take proper actions to prevent cascading events.

With implementations of renewable energy, especially wind and solar, the net load curve for power supply will become more variable. The fast-changing demand places challenges on the existing DSA system to provide accurate security predictions, especially when the system operates near limitations.

The contribution of this research is to developed a novel open source DSA system to improve the prediction accuracy of the existing DSA system. The improvements of the proposed DSA system contains the following aspects.

i) The new DSA system uses PMU data to calculate system states. With usage of PMU data, the new DSA system can perform linear state estimation instead of nonlinear state estimation. This helps the DSA system to obtain system states more quickly and more accurately. Therefore, the process for the first stage of DSA systems will be more accurate.

ii) The new DSA system uses forecast operating points to examine system endangerments instead of present operating points. According to the simulation results, the new DSA system can avoid up to 23% error introduced by the fast-changing load for the IEEE 118-bus system. This demonstrates that the proposed DSA system can

provide more accurate prediction for system security for the future grid with the help of load forecast data with acceptable error.

iii) Typically, a system will be more vulnerable when power flow increases. Because the proposed DSA system provides more accurate system security predictions, it can strengthen operation limits appropriately when net load increases. Therefore, the power system will be more reliable during the period of increasing net load.

iv) On the contrary, when power flow decreases, a system can withstand larger disturbances as well as operate in a more economic way with relaxed limitations. The new DSA scheme can help the system to achieve more economic operation by preventing the system from generating alerts for impossible endangerments.

Part of this research is published in [40].

5.2 Future Work

In this research, we performed a time-domain simulation for transient stability analysis. The advantage for using time-domain simulation is that it can provide more accurate and credible results. However, to realize the time domain dynamic simulation for a wide area system is very time consuming. An intelligent DSA system can be developed based on the off-line simulation results in the future. This system can make use of forecast operating points to predict system security.

REFERENCES

[1] "Nerc planning standards (1997, sept.).." `http://www.nerc.com/~filez/pss-psg.html`. Accessed: 2017-34-19. 1.1.1

[2] P. Kundur, J. Paserba, V. Ajjarapu, G. Andersson, A. Bose, C. Canizares, N. Hatziargyriou, D. Hill, A. Stankovic, C. Taylor, T. V. Cutsem, and V. Vittal, "Definition and classification of power system stability ieee/cigre joint task force on stability terms and definitions," *IEEE Transactions on Power Systems*, vol. 19, pp. 1387–1401, Aug 2004. 1.1.1

[3] A. A. Fouad, F. Aboytes, V. F. Carvalho, S. L. Corey, K. J. Dhir, and R. Vierra, "Dynamic security assessment practices in north america," *IEEE Transactions on Power Systems*, vol. 3, pp. 1310–1321, Aug 1988. 1.1.2, 1.2

[4] "Ieee standard for synchrophasor measurements for power systems," *IEEE Std C37.118.1-2011 (Revision of IEEE Std C37.118-2005)*, pp. 1–61, Dec 2011. 1.1.3

[5] P. Overholt, D. Ortiz, and A. Silverstein, "Synchrophasor technology and the doe: Exciting opportunities lie ahead in development and deployment," *IEEE Power and Energy Magazine*, vol. 13, pp. 14–17, Sept 2015. 1.1.3

[6] G. D. Friedlander, "Prevention of power failures 2014;the fpc report of 1967," *IEEE Spectrum*, vol. 5, pp. 53–61, Feb 1968. 1.2

[7] P. Kundur, *Transient stability*. McGraw-Hill, Inc, 1994. 1.2

[8] "Parallel processing in power systems computation," *IEEE Transactions on Power Systems*, vol. 7, pp. 629–638, May 1992. 1.2, 1.2

[9] S. Huang, Y. Chen, C. Shen, W. Xue, and J. Wang, "Feasibility study on online dsa through distributed time domain simulations in wan," *IEEE Transactions on Power Systems*, vol. 27, pp. 1214–1224, Aug 2012. 1.2, 1.2

[10] J. Tong and L. Wang, "Design of a dsa tool for real time system operations," in *2006 International Conference on Power System Technology*, pp. 1–5, Oct 2006. 1.2, 1.2, 1.3, 3.2

[11] M. A. El-Kady, C. K. Tang, V. F. Carvalho, A. A. Fouad, and V. Vittal, "Dynamic security assessment utilizing the transient energy function method," *IEEE Power Engineering Review*, vol. PER-6, pp. 55–56, Aug 1986. 1.2

[12] A. Michel, A. Fouad, and V. Vittal, "Power system transient stability using individual machine energy functions," *IEEE Transactions on Circuits and Systems*, vol. 30, pp. 266–276, May 1983. 1.2

[13] J. H. Chow, A. Chakrabortty, M. Arcak, B. Bhargava, and A. Salazar, "Synchronized phasor data based energy function analysis of dominant power transfer paths in large power systems," *IEEE Transactions on Power Systems*, vol. 22, pp. 727–734, May 2007. 1.2

[14] J. Yan, C. C. Liu, and U. Vaidya, "Pmu-based monitoring of rotor angle dynamics," *IEEE Transactions on Power Systems*, vol. 26, pp. 2125–2133, Nov 2011. 1.2

[15] H. Modir and R. A. Schlueter, "A dynamic state estimator for dynamic security assessment," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-100, pp. 4644–4652, Nov 1981. 1.2

[16] N. Zhou, D. Meng, Z. Huang, and G. Welch, "Dynamic state estimation of a synchronous machine using pmu data: A comparative study," in *2015 IEEE Power Energy Society General Meeting*, pp. 1–1, July 2015. 1.2

[17] E. Ghahremani and I. Kamwa, "Local and wide-area pmu-based decentralized dynamic state estimation in multi-machine power systems," *IEEE Transactions on Power Systems*, vol. 31, pp. 547–562, Jan 2016. 1.2

[18] F. Wu and Y.-K. Tsai, "Probabilistic dynamic security assessment of power systems-i: Basic model," *IEEE Transactions on Circuits and Systems*, vol. 30, pp. 148–159, Mar 1983. 1.2, 1.2

[19] T. Odun-Ayo and M. L. Crow, "Structure-preserved power system transient stability using stochastic energy functions," *IEEE Transactions on Power Systems*, vol. 27, pp. 1450–1458, Aug 2012. 1.2

[20] M. Abapour and M. R. Haghifam, "On-line assessment of the transient instability risk," *IET Generation, Transmission Distribution*, vol. 7, pp. 602–612, June 2013. 1.2

[21] D. J. Sobajic and Y. H. Pao, "Artificial neural-net based dynamic security assessment for electric power systems," *IEEE Transactions on Power Systems*, vol. 4, pp. 220–228, Feb 1989. 1.2, 1.2

[22] A. D. Rajapakse, F. Gomez, K. Nanayakkara, P. A. Crossley, and V. V. Terzija, "Rotor angle instability prediction using post-disturbance voltage trajectories," *IEEE Transactions on Power Systems*, vol. 25, pp. 947–956, May 2010. 1.2

[23] K. Sun, S. Likhate, V. Vittal, V. S. Kolluri, and S. Mandal, "An online dynamic security assessment scheme using phasor measurements and decision trees," *IEEE Transactions on Power Systems*, vol. 22, pp. 1935–1943, Nov 2007. 1.2, 1.2

[24] F. Hashiesh, H. E. Mostafa, A. R. Khatib, I. Helal, and M. M. Mansour, "An intelligent wide area synchrophasor based system for predicting and mitigating transient instabilities," *IEEE Transactions on Smart Grid*, vol. 3, pp. 645–652, June 2012. 1.2, 1.2

[25] Y. Xu, Z. Y. Dong, J. H. Zhao, P. Zhang, and K. P. Wong, "A reliable intelligent system for real-time dynamic security assessment of power systems," *IEEE Transactions on Power Systems*, vol. 27, pp. 1253–1263, Aug 2012. 1.2, 1.2, 1.2

[26] A. S. Debs and R. E. Larson, "A dynamic estimator for tracking the state of a power system," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-89, pp. 1670–1678, Sept 1970. 1.2

[27] A. Abur and antonio Gomez Exposito, *Power System State Estimation Theory and Implementation*. CRC Press. 1.2

[28] C. Mishra, V. A. Centeno, and A. Pal, "Kalman-filter based recursive regression for three-phase line parameter estimation using synchrophasor measurements," in *2015 IEEE Power Energy Society General Meeting*, pp. 1–5, July 2015. 1.2

[29] S. Rovnyak, S. Kretsinger, J. Thorp, and D. Brown, "Decision trees for real-time transient stability prediction," *IEEE Transactions on Power Systems*, vol. 9, pp. 1417–1426, Aug 1994. 1.2

[30] M. He, V. Vittal, and J. Zhang, "Online dynamic security assessment with missing pmu measurements: A data mining approach," *IEEE Transactions on Power Systems*, vol. 28, pp. 1969–1977, May 2013. 1.2

[31] C. A. Jensen, M. A. El-Sharkawi, and R. J. Marks, "Power system security assessment using neural networks: feature selection using fisher discrimination," *IEEE Transactions on Power Systems*, vol. 16, pp. 757–763, Nov 2001. 1.2

[32] L. S. Moulin, A. P. A. da Silva, M. A. El-Sharkawi, and R. J. Marks, "Support vector machines for transient stability analysis of large-scale power systems," *IEEE Transactions on Power Systems*, vol. 19, pp. 818–825, May 2004. 1.2

[33] A. D. Rajapakse, F. Gomez, K. Nanayakkara, P. A. Crossley, and V. V. Terzija, "Rotor angle instability prediction using post-disturbance voltage trajectories," *IEEE Transactions on Power Systems*, vol. 25, pp. 947–956, May 2010. 1.2

[34] J. Shu, W. Xue, and W. Zheng, "A parallel transient stability simulation for power systems," *IEEE Transactions on Power Systems*, vol. 20, pp. 1709–1717, Nov 2005. 1.2

[35] G. Aloisio, M. A. Bochicchio, M. L. Scala, and R. Sbrizzai, "A distributed computing approach for real-time transient stability analysis," *IEEE Transactions on Power Systems*, vol. 12, pp. 981–987, May 1997. 1.2

[36] W. G. C4.601, "Review of on-line dynamic security assessment tools and techniques," tech. rep., CIGRE, 2007. 1.2, 1.2

[37] T. S. Group, "Technical analysis of the august 14, 2003, blackout: What happened, why, and what did we learn?," tech. rep., North American Electric Reliability council, 2004. 1.2

[38] "2016 state of the interconnection," tech. rep., WECC, 2016. 1.3

[39] "CAISO fast facts." `https://www.caiso.com/Documents/FlexibleResourcesHelpRenewables_FastFacts.pdf`. Accessed: 2017-3-25. 1.3, 1.4

[40] Q. Wang and G. Karady, "Real-time online dynamic security assessment using phasor measurement and load forecast," in *2016 Grid of the Future Symposium*, CIGRE US National Committee, Oct. 2016. 2.2, 5.1

[41] "Historic demand data." `http://www2.nationalgrid.com/UK/Industry-information/Electricity-transmission-operational-data/Data-Explorer/`. Accessed: 2017-3-25. 2.3

[42] H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term load forecasting: a review and evaluation," *IEEE Transactions on Power Systems*, vol. 16, pp. 44–55, Feb 2001. 2.2

[43] O. Abedinia, N. Amjady, and H. Zareipour, "A new feature selection technique for load and price forecast of electrical power systems," *IEEE Transactions on Power Systems*, vol. 32, pp. 62–74, Jan 2017. 2.2, 4.3

[44] K. D. Jones, J. S. Thorp, and R. M. Gardner, "Three-phase linear state estimation using phasor measurements," in *2013 IEEE Power Energy Society General Meeting*, pp. 1–5, July 2013. 2.3, 3.1.3

[45] A. Pal, G. A. Sanchez-Ayala, V. A. Centeno, and J. S. Thorp, "A pmu placement scheme ensuring real-time monitoring of critical buses of the network," *IEEE Transactions on Power Delivery*, vol. 29, pp. 510–517, April 2014. 2.4

[46] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, pp. 12–19, Feb 2011. 2.5

[47] D. Hu and V. M. Venkatasubramanian, "New wide-area algorithms for detection and mitigation of angle instability using synchrophasors," in *2007 IEEE Power Engineering Society General Meeting*, pp. 1–8, June 2007. 2.6

[48] "Matpower." `http://www.pserc.cornell.edu/matpower/`. Accessed: 2017-3-25. 3.1.1

[49] "IEEE 118-bus system." `http://icseg.iti.illinois.edu/ieee-118-bus-system/`. Accessed: 2017-3-25. 4.1, 4.1.1, 4.1

[50] M. Burnett, "Menergy storage and the california "duck curve"." `http://large.stanford.edu/courses/2015/ph240/burnett2/`. Accessed: 2017-3-31. 4.1

[51] Q. Wang, Z. Tang, I. Knezevic, J. Yu, and G. Karady, "Power system protection education and digital relay training based on a physical platform," in *2016 North American Power Symposium (NAPS)*, pp. 1–5, Sept 2016. 4.1.2

APPENDIX A

MATLAB FILE FOR OPTIMAL PMU PLACEMENT AND PMU-BSED STATE ESTIMATION

```
% Assume only PMU data is used for state estimation, complex
    numbers are
% used, further decouple are required regarding. M is the
    measurements:
% voltage phasor, current phasor. E is the measurement errors
    . W is the
% relationship between the phasor measurement and the states.
function State_Estimation(Input)

%% extract from MPC
mpc      = psse2mpc(Input);
baseMVA = mpc.baseMVA;
bus      = mpc.bus;
branch   = mpc.branch;

%% constants
nb = size(bus, 1);                  %% number of buses
nl = size(branch, 1);               %% number of lines

%% define named indices into bus, branch matrices
[PQ, PV, REF, NONE, BUS_I, BUS_TYPE, PD, QD, GS, BS, BUS_AREA
    , VM, ...
     VA, BASE_KV, ZONE, VMAX, VMIN, LAM_P, LAM_Q, MU_VMAX,
        MU_VMIN] = idx_bus;
[F_BUS, T_BUS, BR_R, BR_X, BR_B, RATE_A, RATE_B, RATE_C, ...
     TAP, SHIFT, BR_STATUS, PF, QF, PT, QT, MU_SF, MU_ST, ...
     ANGMIN, ANGMAX, MU_ANGMIN, MU_ANGMAX] = idx_brch;

%% build binary connective matrix A
[Ybus, Yf, Yt] = makeYbus(mpc);
Ybus = full(Ybus);
Yf   = full(Yf);                    % From bus Y matrix(1/Zf+Yf/2):
    line is the branch number, column is the bus number
Yt   = full(Yt);                    % T busY matrix (-1/Zf); line is
    the branch number, column is the bus number
A = zeros(nb,nb);                   % If k = m or k and m are
    connected, A(k,m)=1
for k = 1:nb                        % Transfer the enties of bus
    admittance matrix to binary form
     for m = 1:nb
          if Ybus(k,m)~= 0
               A(k,m)=1;
          end
          if k == m
               A(k,m) = 1;
          end
     end
```

```
end

%% Determine PMU number and placement
f=ones(nb,1);
b = ones(nb,1);
lb = zeros(nb,1);
ub = ones(nb,1);
A=sparse(A);
b=sparse(b);
f=sparse(f);
x=intlinprog(f,nb, -A, -b,[],[],lb,ub);
PMU_place = find(x~=0);
PMU_num = length(PMU_place);
%[x, fval, exitflag, output] = cplexbilp(f, -A, -bineq); %
    must have IBM
%cplex package to usg this function.

%% State Estimation begins
%% build connection matrices
f = branch(:, F_BUS);                                   %% list
    of "from" buses
t = branch(:, T_BUS);                                   %% list
    of "to" buses
Cf = sparse(1:nl, f, ones(nl, 1), nl, nb);              %%
    connection matrix for line & from buses
Ct = sparse(1:nl, t, ones(nl, 1), nl, nb);              %%
    connection matrix for line & to buses

Cf = full(Cf);
Ct = full(Ct);

%% system measurement-state matrix
I = eye(nb);
Y_measured_voltage = I(PMU_place,:);                                %
     Voltage Measurement Bus Incidence Matrix
C = Cf + Ct;                                                        %
     connection matrix for line & from and to buses
PMU_current_place = find(C * x~=0);                             %
    measured lines
Y_measured_current = Yf(PMU_current_place,:);                   %
    current measurement-bus admittance matrix
A = [Y_measured_voltage;Y_measured_current];

%% Get measurement
results = runpf(mpc);                       % create a set of
    measurement by solving powewr flow
```

```
Vm_m     = results.bus(:,8);                % get the magnitude
    of measured voltage
Vm_a     = results.bus(:,9)*pi/180;         % get the angle of
    the measured voltage, and convert from degree to radiance
Vm = Vm_m .* exp(j*Vm_a);                   % calculate the
    complex value of measured voltage
Im = Yf*Vm;                                 % calculate branch
    current
M_V = Vm(PMU_place,:);                      % get PMU voltage
    measurement
M_I = Im(PMU_current_place,:);
M  = [M_V; M_I];

%% State Estimation
% Get W matrix corresponding to L
W = A;
W_pse = pinv(W);                    % pseudoinverse of W
Ve    = W_pse * M;                  % estimated voltage
Ve_m   = abs(Ve);
Ve_a   = angle(Ve)*180/pi;          % get the angle of estimated
    voltage and convert it from angle to degree.
```

APPENDIX B

PYTHON FILE FOR AUTOMATION

```
import subprocess
import time
import os
import os.path
import win32com.client
import sys

PSATPATH = "D:\\DSATools_16-SL\\DSATools_16_net\\Psat\\bin\\
    psat.exe"     #PSAT.exe location
# PFFOLDER = "C:\\Users\\qwang112\\Desktop\\PSATCASES\\PFLIST
    \\"   #powerflow folder location
PFFOLDER = "D:\Thesis\\PSATCASES\\PFLIST\\"
OUTFOLDER = "C:\\Users\\qwang112\\Desktop\\PSATCASES\\"    #
    solved powerflow folder location
OUTFOLDER = "D:\Thesis\\PSATCASES\\"    #solved powerflow
    folder location

# PSAT_py_script = "C:\\Users\\qwang112\\Desktop\\PSATCASES\\
    SolveAndSave.txt"
PSAT_py_script = "D:\Thesis\\PSATCASES\\SolveAndSave.txt"
MAXWAIT = 20 #defines the maximin time (sec) to wait for psat
     to complete

TSATPATH = "D:\\DSATools_16-SL\\DSATools_16_net\\Tsat\\bin\\
    tsat_batch"
# TSATFOLDER = "C:\\Users\\qwang112\\Desktop\\TSATCASES\\"
TSATFOLDER = "D:\Thesis\\TSATCASES\\"
StandardRawFile = "C:\\Users\\qwang112\\Desktop\\IEEE_118_Bus
    _for_PSAT.raw" #Standard IEEE 118 bus raw file

POWERFLOWS = []
SolvedPF =[]
TSA_FILE = []
BIN_FILE = []
minute = [0,30]                    #powerflow list
for i in range(0,24):
    for num in range(0,2):
        FileName = "IEEE118_%d_"%i + "%d.raw"%minute[num]
              #Raw Power Flow file name in order of time
        POWERFLOWS.append(FileName)
        FileName = "Solved_IEEE118_%d_"%i + "%d.pfb"%minute[
            num]      #Raw Power Flow file name in order of
            time
        SolvedPF.append(FileName)
        FileName = "IEEE118_%d_"%i + "%d.tsa"%minute[num]
            #Tsat project file name in order of time
        TSA_FILE.append(FileName)
```

```
        FileName = "IEEE118_%d_"%i + "%d.bin"%minute[num]
            #Tsat output file name in order of time
        BIN_FILE.append(FileName)

#used to keep PSAT from "popping-up" by starting it minimized
subprocess.STARTF_USESHOWWINDOW = 1
subprocess.SW_SHOWMINNOACTIVE = 7

#-------------------------------------------------------------#
# Instruct PSAT to run a PSAT-python script directly (PSAT
    v10+ only) #
def Run_Python_File(scriptpath, output, maxwait):
    #We monitor for the output file to let us know when PSAT
        has finished
    #clear any output trigger file before proceeding
    if(os.path.isfile(output)):
        print "Output file found and removed"
        os.remove(output)
    #Command line + arguments
    cmd = PSATPATH+" \""+scriptpath+"\" python"
    print "Running ",cmd

    startupinfo = subprocess.STARTUPINFO()
    startupinfo.dwFlags |= subprocess.STARTF_USESHOWWINDOW
    startupinfo.wShowWindow = subprocess.SW_SHOWMINNOACTIVE

    pPSAT = subprocess.Popen(cmd, startupinfo=startupinfo)

    # check every second to see if PSAT has finished
    timeCount = 0
    while (timeCount < maxwait): # wait maxwait seconds #
      timeCount+=1
      if (pPSAT.poll() == None):
          time.sleep(1)
          if(os.path.isfile(output)):
              print "Output file found... exiting PSAT"
              time.sleep(10)
              pPSAT.terminate()
              break
          #print maxwait - timeCount #print a countdown
      else:
          #PSAT exited on it's own
          print "PSAT Exited (",pPSAT.returncode,")"
          break
    else:
       print "PSAT Killed"
       pPSAT.terminate()
```

```
    if(os.path.isfile(output)): #Did it do what it was
        supposed to?
        return True
    else:
        return False

#-------------------------------------------------------------#
#function to create a PSAT python script to open, solve and
    save a powerflow
def Create_PSATPython_File(filepath,inputpf,outputpf):
    filelines = []
    filelines.append("from psat_python import *\r\n")
    filelines.append("error = psat_error()\r\n")
    filelines.append("psat_command(\"Import:\\\""+inputpf+"
        \\\";PTI Rawd 33\",error)\r\n")
    filelines.append("psat_command(\"Solve\",error)\r\n")
    filelines.append("Vnum_H = []\n")
    filelines.append("Vnum_L = []\n")
    filelines.append("Vmag_H = []\n")
    filelines.append("Vmag_L = []\n")
    filelines.append("for busid in range(0,119):\n")
    filelines.append("  busdat = get_bus_dat(busid,error)\n")
    filelines.append("  if busdat.vmag > 1.05 :\n")
    filelines.append("      Vnum_H.append(busdat.number)\n")
    filelines.append("      Vmag_H.append(busdat.vmag)\n")
    filelines.append("  elif busdat.vmag < 0.95:\n")
    filelines.append("      Vnum_L.append(busdat.number)\n")
    filelines.append("      Vmag_L.append(busdat.vmag)\n")
    filelines.append("psat_command(\"SavePowerflowAs:\\\""+
        outputpf+"\\\"\",error)\r\n")
    filelines.append("if len(Vnum_H) != 0:\n")
    filelines.append("  msgtext='The following bus are above
        high voltage limit:'\n")
    filelines.append("  for num in range(0,len(Vnum_H)):\n")
    filelines.append("      msgtext=msgtext + 'Bus: %d,'%
        Vnum_H[num]+' Vmag:%f;'%Vmag_H[num]\n")
    filelines.append("if len(Vnum_L) != 0:\n")
    filelines.append("  msgtext = 'The following bus are
        below low voltage limit:'\n")
    filelines.append("  for num in range(0,len(Vnum_L)):\n")
    filelines.append("      msgtext = msgtext + 'Bus: %d,'%
        Vnum_L[num]+' Vmag:%f;'%Vmag_L[num]\n")
    filelines.append("psat_msg_box(msgtext)\n")

    filename = filepath
    dir = os.path.dirname(filename)
```

```
    if not os.path.exists(dir):
        os.makedirs(dir)
    FILE = open(filepath, 'w')
    FILE.writelines(filelines)
    FILE.close()

#----Get branch information and create line contingency----#
def Create_linefault_index(inputfile):
    FromBus = []
    ToBus = []
    ID = []
    with open(inputfile,'r') as pf:
        lines = pf.readlines()
        ## Find branch Data Index
        for line in lines:
            if "BEGIN BRANCH DATA" in line:
                Idx_BrStart = lines.index(line) + 1
            elif "END OF BRANCH DATA" in line:
                Idx_BrEnd = lines.index(line)
                break

        for br in range(Idx_BrStart,Idx_BrEnd):
            temp = lines[br].split(",")
            FromBus.append(int(temp[0]))
            ToBus.append(int(temp[1]))
            string = temp[2]
            ID.append(int(string[1:-1]))

    branchnum =Idx_BrEnd - Idx_BrStart
    return branchnum, FromBus, ToBus, ID


#----------------------Change .tsa file----------------------#
def Create_TSA_File(sovled_pf_file,tsafile):
    if(os.path.isfile(tsafile)):                    #clear the
        existing .tsa file
        print "TSA file found and removed"
        os.remove(tsafile)
    print "Creating ",tsafile,"\n"

    FromBus = []
    ToBus = []
    ID = []

    with open("C:\\Users\\qwang112\\Desktop\\IEEE 118 Bus for
        PSAT.raw",'r') as pf:
        lines = pf.readlines()
```

```
        ## Find branch Data Index
        for line in lines:
            if "BEGIN BRANCH DATA" in line:
                Idx_BrStart = lines.index(line) + 1
            elif "END OF BRANCH DATA" in line:
                Idx_BrEnd = lines.index(line)
                break

        for br in range(Idx_BrStart,Idx_BrEnd):
            temp = lines[br].split(",")
            FromBus.append(int(temp[0]))
            ToBus.append(int(temp[1]))
            string = temp[2]
            ID.append(int(string[1:-1]))

    branch =Idx_BrEnd - Idx_BrStart

    with open(tsafile,'a+')as TSAT:
# edit for base scenario
        Date = 'Date = ' + time.strftime("%a,%b,%d,%y,%H:%M:%
           S",time.gmtime())+'\n'
        pf_File = 'File =' + sovled_pf_file + '\n'
        lines = ['[TSAT 8.0]\n','\n','[Base Scenario]\n','\n'
           ,'{Scenario Description}\n',\
                'Title = non fault\n','Author = Qiushi\n',
                   Date,'{End Scenario Description}\n',\
                '\n',\
                '{Scenario Parameters}\n','Transient
                   Stability Margin Algorithm = SWING\n',\
                '{End Scenario Parameters}\n','\n',\
                '{Powerflow Data}\n',pf_File,'Format = PSF
                   NUMBER\n','Parameter File =\n',\
                'Solution File =\n','Control Mode File = \n'
                    ,'Node Breaker File = \n',\
                'Solve Base Powerflow = \n','{End Powerflow
                   Data}\n','\n',\
                '{Dynamic Data}\n','Format = DSATools\n','
                   Name Option = 0\n',\
                'File = D:\\Thesis\\TSATCASES\\
                   modified118_TSAT_dyn.dat\n' ,'{End
                   Dynamic Data}\n','\n',\
                '{Dynamic Representation Data}\n','Name
                   Option = 0\n','File = \n','{End Dynamic
                   Representation Data}\n','\n',\
                '{Monitor Data}\n','Name Option = 0\n','File
                   = D:\Thesis\\118bus_mon.mon \n',\
                'VSAT Monitor File = \n',"Reference
```

```
                    Generator = '   69 '1'' \n", '{End
                        Monitor Data}\n', '\n'\
                    '{PMU Data}\n', 'Name Option = 0\n', 'TCP Port
                        = 7027\n', '{End PMU Data}\n', '\n',\
                    '{Switching Data}\n', 'Name Option = 0\n', '
                        File = D:\Thesis\\Contingency\\118
                        _bus_fault_non.swi \n',\
                    'Must-run File =  \n',"Don't-run File =  \n"
                        , '{End Switching Data}\n', '\n', '{Relay
                        Data}\n', 'File =  \n',\
                    '{End Relay Data}\n', '\n', '{Criteria Data}\n
                        ', 'Name Option = 0\n', 'File =  \n', '{End
                        Criteria Data}\n', '\n',\
                    '{Transaction Data}\n', 'Name Option = 0\n', '
                        Transfer File = \n', 'Parameter File = \n'
                        ,\
                    'Interface And Circuit File = \n', 'Generator
                        Capability File =\n' , 'Generator
                        Coupling File = \n',\
                    '{End Transaction Data}\n', '\n', '{Sequence
                        Network Data}\n', 'File =  \n', 'Format =
                        PSS/E\n',\
                    '{End Sequence Network Data}\n', '\n', '[End
                        Base Scenario]\n']
        for i in range(len(lines)):
            TSAT.write(lines[i])

# edit for sub-scenarios, begin bus faults
        for num in range(1, 119):
            TSAT.write('[Scenario]\n' + '\n')
            TSAT.write('{Scenario Description}\n')
            TSAT.write('Title = fault at bus %d\n' %num)
            TSAT.write('Author = Qiushi\n')
            TSAT.write('Date = ' + time.strftime("%a,%b,%d,%y
                ,%H::%M:%S",time.gmtime())+'\n')
            TSAT.write('{End Scenario Description}\n' + '\n')
            TSAT.write('{Scenario Parameters}\n')
            TSAT.write('{End Scenario Parameters}\n' + '\n')
            TSAT.write('{Switching Data}\n')
            TSAT.write('Name Option = 0\n')
            TSAT.write('File = D:\Thesis\\Contingency\\fault
                %d_118bus.swi\n' %num )
            TSAT.write('Must-run File =\n')
            TSAT.write("Don't-run File =\n")
            TSAT.write('{End Switching Data}\n')
            TSAT.write('[End Scenario]\n')
            TSAT.write('\n'+'\n')
```

```
# begin line faults, from bus to to bus
        for num in range(0, branch):
            TSAT.write('[Scenario]\n' + '\n')
            TSAT.write('{Scenario Description}\n')
            TSAT.write('Title = fault on the 5 percent of the
                line from bus %d '%FromBus[num]+' to bus %d\n'
                %ToBus[num])
            TSAT.write('Author = Qiushi\n')
            TSAT.write('Date = ' + time.strftime("%a,%b,%d,%y
                ,%H::%M:%S",time.gmtime())+'\n')
            TSAT.write('{End Scenario Description}\n' + '\n')
            TSAT.write('{Scenario Parameters}\n')
            TSAT.write('{End Scenario Parameters}\n' + '\n')
            TSAT.write('{Switching Data}\n')
            TSAT.write('Name Option = 0\n')
            TSAT.write('File = D:\Thesis/Contingency/fault_%d
                ' %FromBus[num] \
                        + ' to %d_118.swi\n' %ToBus[num])
            TSAT.write('Must-run File =\n')
            TSAT.write("Don't-run File =\n")
            TSAT.write('{End Switching Data}\n')
            TSAT.write('[End Scenario]\n')
            TSAT.write('\n'+'\n')

# begin line faults, to bus to from bus
        for num in range(0, branch):
            TSAT.write('[Scenario]\n' + '\n')
            TSAT.write('{Scenario Description}\n')
            TSAT.write('Title = fault on the 5 percent of the
                line from bus %d '%ToBus[num]+' to bus %d\n'%
                FromBus[num])
            TSAT.write('Author = Qiushi\n')
            TSAT.write('Date = ' + time.strftime("%a,%b,%d,%y
                ,%H::%M:%S",time.gmtime())+'\n')
            TSAT.write('{End Scenario Description}\n' + '\n')
            TSAT.write('{Scenario Parameters}\n')
            TSAT.write('{End Scenario Parameters}\n' + '\n')
            TSAT.write('{Switching Data}\n')
            TSAT.write('Name Option = 0\n')
            TSAT.write('File = D:\Thesis/Contingency/fault_%d
                ' %ToBus[num] \
                        + ' to %d_118.swi\n' %FromBus[num])
            TSAT.write('Must-run File =\n')
            TSAT.write("Don't-run File =\n")
            TSAT.write('{End Switching Data}\n')
            TSAT.write('[End Scenario]\n')
```

```
                TSAT.write('\n'+'\n')

    TSAT.close()

##-------------------------------TSAT-------------------------------##
#function to call TSAT to determine which scenario is
    unsecured

def Run_TSAT(filepath_tsa,outputTSAT,branchnum):

    pcmd = TSATPATH + " \"" + filepath_tsa
    print "running ", pcmd
    print TSATPATH
    subprocess.call([TSATPATH, filepath_tsa],shell=True)

    print "Generating report for", filepath_tsa
    reader = win32com.client.Dispatch("ResultScript.BinReader
        ")
    #time.sleep(20)
    print outputTSAT
    reader.file = outputTSAT
    reader.scen = 1
    reader.ctg = 1
    Gen_bus =
        [10,12,25,26,31,46,49,54,59,61,65,66,69,80,87,89,100,103,111]


    scen = 119 + 2*branchnum +1
    unsecured_scen = []
    Unsecurity = {}
    count = 0

    #identify unsecured contingency and the corresponding
        generators
    for num in range (1,scen) :
        reader.scen = num
        reader.quan = "rgcn_ang"
        reader.curvename = "118bus"
        COA = reader.curveValues()
        COA_1 = COA[-1]
        gen_num = []
        d_angles = []
        count =0

        for w in Gen_bus[:]:
            reader.quan = "gen_ang"
            reader.bus1 = w
```

```
                reader.id = "1"
                arr = reader.curveValues()
                arr_1 = arr[-1]
                Delta_Angle = abs(arr_1 - COA_1)
                '''
                maxValue = (max(Delta_Angle))
                minValue = abs(min(Delta_Angle))
                maxValue = max([maxValue,minValue])
                '''
                if Delta_Angle > 60:
                    gen_num.append(w)
                    d_angles.append(Delta_Angle)
                    if count == 0:
                        unsecured_scen.append(num)       #to
                            prevent multiple insecurity report for
                             the same fault
                count = count +1
            #Store the generator number with maximum angle
               difference between rotor angle and COA.
            if len(gen_num)>0:
                temp = sorted(zip(d_angles,range(len(d_angles))))
                if len(temp)>= 5:
                    temp = temp[-5:]
                gen_indicator = [x[1] for x in temp]  #get the
                   indicator number for the firt fifth maximum
                   d_angles
                Unsecurity[num] = [gen_num[x] for x in
                   gen_indicator]

    return unsecured_scen, Unsecurity


##-------------------TSAT results Output--------------------##
#print alert for unsecured contingengy, generate documents
   for the plots of unsecured contingency
def OutPut_TSAT(unsecured_scen,Unsecurity,branchnum,frombus,
   tobus,ID,TSAToutput):
    unsecured_scen_name=[]

    print unsecured_scen

    if len(unsecured_scen)==0:
        print("The system is secured\n")
    else:
        print('The system is unsecured when: ')
        for num in range(0,len(unsecured_scen)):
            if unsecured_scen[num]<1:
```

```
            name = "None fault in the system"
            unsecured_scen_name.append(name)
        elif unsecured_scen[num]<119:
            name = "A fault on bus %d"%(unsecured_scen[
                num]-1) #substract 1 for base case without
                  fault
            unsecured_scen_name.append(name)
        elif unsecured_scen[num]<119+branchnum:
            name = "A fault on 5 percent of the line from
                bus %d"%frombus[unsecured_scen[num
                ]-119-1] \
                    + " to bus %d"%tobus[unsecured_scen[
                        num]-119-1]
            # substract another 1 because the fist
                element in a list is tobus[0]
            unsecured_scen_name.append(name)
        else:
            name = "A fault on 5 percent of the line from
                bus %d"%tobus[unsecured_scen[num]-119-
                branchnum-1]+\
                    " to bus %d"%frombus[unsecured_scen[num
                        ]-119-branchnum-1]
            unsecured_scen_name.append(name)
print(unsecured_scen_name)
print('\n')

#Plot the unsecured scenorio (ragione center of angle
    and the unsecured generator)
g=win32com.client.Dispatch("ResultScript.Plotter")
Plot_output = TSAToutput[:-4] + ".doc"
g.setOutput(Plot_output,"doc")
g.setGrid("dash","dash")
g.PlotsPerPage=2
g.DoMark = True
g.DoColor = True

reader = win32com.client.Dispatch("ResultScript.
    BinReader")
reader.file = TSAToutput
reader.ctg = 1
reader.quan = "gen_ang"
Gen_bus =
    [10,12,25,26,31,46,49,54,59,61,65,66,69,80,87,89,100,103,111]



for num in range (0,len(unsecured_scen)) :
```

```
            reader.scen = unsecured_scen[num]
            reader.quan = "rgcn_ang"
            reader.curvename = "118bus"
            COA = reader.curveValues()
            xarr = reader.timeValues()
            curvename = "COA"
            g.AddTYCurve(curvename,xarr,COA)

            for w in Unsecurity[unsecured_scen[num]]:
                reader.bus1 = w
                reader.id = "1"
                reader.quan = "gen_ang"
                arr = reader.curveValues()
                curvename = "Gen %d"%w
                g.AddTYCurve(curvename,xarr,arr)
            g.DoPlot(TSATtoutput[-8:-4] + " " +
                unsecured_scen_name[num],"Time (sec)","
                Generator rotor angle(deg)",\
                    "Reference Generator = Gen at Bus 69")
        g.Finish()


##-------------------------Main-------------------------##
branchnum, FromBus, ToBus, ID = Create_linefault_index(
    StandardRawFile)      #Get branch number and bus index
for i in range(len(POWERFLOWS)):
    print "Processing Powerflow ( ",\
        time.strftime("%a,%b,%d,%y,%H:%M:%S",time.localtime
            ()),\
        "): ",SolvedPF[i]
    output_pf = OUTFOLDER+SolvedPF[i]
    Create_PSATPython_File(PSAT_py_script,PFFOLDER+POWERFLOWS
        [i],output_pf)  #create a PSAT-python script
    success = Run_Python_File(PSAT_py_script,output_pf,
        MAXWAIT)
    print "Success = ",success,"\r\n"
    if(success):
        Create_TSA_File(output_pf,TSATFOLDER + TSA_FILE[i])
        Unsecured_Scen, Unsecurity = Run_TSAT(TSATFOLDER +
            TSA_FILE[i],TSATFOLDER + BIN_FILE[i],branchnum)
        OutPut_TSAT(Unsecured_Scen,Unsecurity,branchnum,
            FromBus,ToBus,ID,TSATFOLDER + BIN_FILE[i])
        pass #Do other processing like vsat or tsat
```

APPENDIX C

IEEE 118 BUS SYSTEM RAW LOAD FILE

```
from psat_python import *

error = psat_error()

psat_command("Import:\"C:\Users\qwang112\Desktop\PSATCASES\PFLIST\IEEE
118_5_30pm.raw\";PTI Rawd 33",error)

psat_command("Solve",error)

Vnum_H = []
Vnum_L = []
Vmag_H = []
Vmag_L = []
for busid in range (0,119):
  busdat = get_bus_dat(busid,error)
  if busdat.vmag > 1.05 :
      Vnum_H.append(busdat.number)
      Vmag_H.append(busdat.vmag)
  elif busdat.vmag < 0.95:
      Vnum_L.append(busdat.number)
      Vmag_L.append(busdat.vmag)
psat_command("SavePowerflowAs:\"C:\Users\qwang112\Desktop\PSATCASES
\Solved_IEEE118_5_30pm.pfb\"",error)

if len(Vnum_H) != 0:
  msgtext='The following bus are above high voltage limit:'
  for num in range (0,len(Vnum_H)):
      msgtext=msgtext + 'Bus: %d,'%Vnum_H[num]+' Vmag:%f;'%Vmag_H[num]
if len(Vnum_L) != 0:
  msgtext = 'The following bus are below low voltage limit:'
  for num in range (0,len(Vnum_L)):
      msgtext = msgtext + 'Bus: %d,'%Vnum_L[num]+' Vmag:%f;'%Vmag_L[num]
psat_msg_box(msgtext)
```

APPENDIX D

MATLAB FILE FOR LOAD CHANGE

```
function Gen_Dist(rawfile,Output)

%% run OPF to determine generation distribution
mpc = psse2mpc(rawfile);      %convert PTI raw data format to
    .m format
mpcm = case118;
mpcm.bus = mpc.bus;
[results,success] = rundcopf(mpcm);
Pgen = results.gen(:,2);          %get optimized P and Q
Qgen = results.gen(:,3);
Gen = [Pgen,Qgen];
csvwrite(Output,Gen);
```

APPENDIX E

PYTHON FILE FOR CHANGING SYSTEM LOAD

```
''' This script change the load of the power flow file  for
   Peak Hours by a time factor of klp and klq for real and
   reactive power respectively
'''

from array import *
import matlab.engine
import csv, os, time

#------------------------------Write New Raw File
   ---------------------------
def Change_load(from_file,to_file,klp,klq):
    ff = open(from_file)
    lines = ff.readlines()
    ## Find Load Data Index
    for line in lines:
        if "BEGIN LOAD DATA" in line:
            Idx_LDSart = lines.index(line)

        elif "END OF LOAD DATA" in line:
            Idx_LDEnd = lines.index(line)

    ## Rewrite Power Flow File
    tf = open(to_file,'w')
    tf.close()
    tf = open(to_file,'r+')

    for line in lines:
        num = lines.index(line)
        if num > Idx_LDSart and num < Idx_LDEnd:
            temp = line.split(",")
            temp[5] = "    " + repr(round(float(temp[5].strip
               ()) * klp,3))
            temp[6] = "    " + repr(round(float(temp[6].strip
               ()) * klq,3))
            line = ','.join(temp)
            tf.write(line)
        else:
            tf.write(line)

    tf.close()
    ff.close()


#------------------------do optimal power flow
   ---------------------------
def OPF(fromfile,tofile):
```

```
    handle = matlab.engine.start_matlab()
    handle.addpath(r'C:\Users\qwang112\Desktop\matpower6.0b1'
        ,nargout=0)

    handle.Gen_Dist(fromfile,tofile,nargout = 0)


#--------------------Retrive optimized Pgen and Qgen
    ----------------------
def Change_Gen(fromfile,gen_file,pf_file):
    P = []
    Q = []
    with open(gen_file,'rb') as f:
        reader = csv.reader(f)
        for row in reader:
            P.append(round(float(row[0]),3))
            Q.append(round(float(row[1]),3))

        # edit power flow file
        ff = open(fromfile)
        lines = ff.readlines()
        ## Find Generator Data Index
        for line in lines:
            if "BEGIN GENERATOR DATA" in line:
                Idx_GenStart = lines.index(line)

            elif "END OF GENERATOR DATA" in line:
                Idx_GenEnd = lines.index(line)

        ## Rewrite Power Flow File
        tf = open(pf_file,'w')
        tf.close()
        tf = open(pf_file,'r+')

        for line in lines:
            num = lines.index(line)
            if num > Idx_GenStart and num < Idx_GenEnd:
                temp = line.split(",")
                gen_num = num - Idx_GenStart - 1
                temp[2] = "    " + repr(P[gen_num])
                temp[3] = "    " + repr(Q[gen_num])
                line = ','.join(temp)
                tf.write(line)
            else:
                tf.write(line)

        tf.close()
```

```
        ff.close()
    os.remove(fromfile)
    os.remove(gen_file)
    printline = fromfile + ' created'
    print(printline)

#-------------------------------Main
    ------------------------------------------
FromFL = 'IEEE 118 Bus for PSAT.raw'          # the standard
   power flow file
# Create the load file name based on peak hours in a day
TempFile = []
FILE = []
Gen_File = []
minute = [00,15,30,45]
for i in range(5,7):
    for num in range(0,4):
        FileName = "IEEE118_%d_"%i + "%dpm_temp.raw"%minute[
           num]          #Temporary Power Flow file name in
           order of time
        TempFile.append(FileName)
        FileName = "C:\Users\qwang112\Desktop\PSATCASES\
           PFLIST\IEEE118_%d_"%i + "%dpm.raw"%minute[num] #
           Power Flow file name in order of time
        FILE.append(FileName)
        filename = "IEEE118_Gen_%d_"%i + "%dpm.csv"%minute[
           num]       #CSV file for P and Q data
        Gen_File.append(filename)

# change load of raw files
klp = [0.9,0.95,1.0,1.1,1.2,1.3,1.4,1.45]
klq = klp

for i in range(0,len(TempFile)):
    Change_load(FromFL,TempFile[i],klp[i],klq[i])
    OPF(TempFile[i],Gen_File[i])
    Change_Gen(TempFile[i],Gen_File[i],FILE[i])
```

APPENDIX F

TSAT DYNAMIC FILE

```
10, 'DG0S4',1,590,0.0046,2.110,2.020,0.155,0.280,0.490,0.215,
0.215,0.5573,0.1371,0.0246,0.0272,2.319,2.000,,0.079,0.349/
12, 'DG0S4',1,125,0.004,1.220,1.160,0.0078,0.174,0.250,0.134,
0.134,8.970,0.500,0.033,0.07,4.768,2.000,,0.1026,0.432/
25, 'DG0S4',1,330,0.000,1.950,1.920,0.199,0.317,1.120,0.200,
0.200,0.9754,0.875,0.0473,0.0134,3.006,2.000,,0.082,0.290/
26, 'DG0S4',1,410,0.0019,1.7668,1.7469,0.1834,0.2738,1.0104,
0.2284,0.2284,0.8418,0.8676,0.035,0.035,3.704,2.000,,0.2632,
0.5351/
31, 'DG0S4',1,75, 0.0031,1.050,0.980,0.070,0.185,0.360,0.130,
0.130,1.0748,0.1102,0.0267,0.0358,6.187,2.000,,0.100,0.3928/
46, 'DG0S4',1,75, 0.0031,1.050,0.980,0.070,0.185,0.360,0.130,
0.130,1.0748,0.1102,0.0267,0.0358,6.187,2.000,,0.100,0.3928/
49, 'DG0S4',1,330,0.000,1.950,1.920,0.199,0.317,1.120,0.200,
0.200,0.9754,0.875,0.0473,0.0134,3.006,2.000,,0.082,0.290/
54, 'DG0S4',1,100,0.0035,1.180,1.050,0.075,0.220,0.380,0.145,
0.145,1.100,0.1086,0.0277,0.0351,4.985,2.000,,0.0933,0.4044/
59, 'DG0S4',1,233,0.0016,1.569,1.548,0.204,0.324,0.918,0.249,
0.249,1.0614,0.8895,0.0336,0.0381,4.122,2.000,,0.0987,0.303/
61, 'DG0S4',1,233,0.0016,1.569,1.548,0.204,0.324,0.918,0.249,
0.249,1.0614,0.8895,0.0336,0.0381,4.122,2.000,,0.0987,0.303/
65, 'DG0S4',1,512,0.004,1.700,1.650,0.160,0.270,0.470,0.200,
0.200,0.6035,0.1367,0.0556,0.0310,2.631,2.000,,0.090,0.400/
66, 'DG0S4',1,512,0.004,1.700,1.650,0.160,0.270,0.470,0.200,
0.200,0.6035,0.1367,0.0556,0.0310,2.631,2.000,,0.090,0.400/
69, 'DG0S4', 1, 590, 0.0046,2.110,2.020,0.155,0.280,0.490,0.215,
0.215,0.5573,0.1371,0.0246,0.0272,2.319,2.000,,0.079,0.349/
80, 'DG0S4', 1, 590, 0.0046,2.110,2.020,0.155,0.280,0.490,0.215,
0.215,0.5573,0.1371,0.0246,0.0272,2.319,2.000,,0.079,0.349/
87, 'DG0S4',1,75, 0.0031,1.050,0.980,0.070,0.185,0.360,0.130,
0.130,1.0748,0.1102,0.0267,0.0358,6.187,2.000,,0.100,0.3928/
89, 'DG0S4',1,835,0.0019,2.183,2.157,0.246,0.413,1.285,0.339,
0.339,5.690,1.500,0.041,0.144,2.4619,2.00,,0.134,0.617/
100, 'DG0S4',1,330,0.000,1.950,1.920,0.199,0.317,1.120,0.200,
0.200,0.9754,0.875,0.0473,0.0134,3.006,2.000,,0.082,0.290/
103, 'DG0S4',1,100,0.0035,1.180,1.050,0.075,0.220,0.380,0.145,
0.145,1.100,0.1086,0.0277,0.0351,4.985,2.000,,0.0933,0.4044/
111, 'DG0S4',1,100,0.0035,1.180,1.050,0.075,0.220,0.380,0.145,
0.145,1.100,0.1086,0.0277,0.0351,4.985,2.000,,0.0933,0.4044/

10, 'EXC1',1,0,0,200,0.3575,1.000,0.011,4.2975,0.000,5.730,0.000,
0.0529,1.000,0,0,5.730,-5.730,1,0,0,0,0/
12, 'EXC1',1,0,0,25,0.200, -0.0601,0.6758,2.4975,0.0949,3.330,
0.37026,0.108,0.350,0,0,1.000,-1.000,1,0,0.060,0,0/
25, 'EXC1',1,0,0,400,0.050,-0.17,1.950,3.6675,0.0111,4.890,0.0178,
0.040,1.000,0,0,3.810,-3.810,1,0,0,0,0/
26, 'EXC1',1,0,0,400,0.020,1.000,0.920,2.4675,0.4351,3.290,0.6001,
```

```
0.030,1.000,0,0,5.270,-5.270,1,0,0,0,0/
31, 'EXC1',1,0,0,0.05,20,1.00,1.980,2.385,0.0951,3.180,0.3712,0,
0.1,0,0,4.380,0,1,0,0,0,0/
46, 'EXC1',1,0,0,0.05,20,1.00,1.980,2.385,0.0951,3.180,0.3712,0,
0.1,0,0,4.380,0,1,0,0,0,0/
49, 'EXC1',1,0,0,400,0.050,-0.17,1.950,3.6675,0.0111,4.890,0.0178,
0.040,1.000,0,0,3.810,-3.810,1,0,0,0,0/
54, 'EXC1',1,0,0,25,0.20,-0.0582,0.6544,2.5785,0.0889,3.438,0.3468,
0.105,0.350,0,0,1.00,-1.00,1,0,0.060,0,0/
59, 'EXC1',1,0,0,250,0.060,1.00,0.613,2.610,0.00,3.480,0.00,0.053,
0.330,0,0,4.420,-4.420,1,0,0,0,0/
61, 'EXC1',1,0,0,250,0.060,1.00,0.613,2.610,0.00,3.480,0.00,0.053,
0.330,0,0,4.420,-4.420,1,0,0,0,0/
65, 'EXC1',1,0,0,200,0.395,1.00,0.008,2.880,0.000,3.840,0.000,
0.0635,1.000,0,0,3.840,-3.840,1,0,0,0,0/
66, 'EXC1',1,0,0,200,0.395,1.00,0.008,2.880,0.000,3.840,0.000,
0.0635,1.000,0,0,3.840,-3.840,1,0,0,0,0/
69, 'EXC1',1,0,0,200,0.3575,1.000,0.011,4.2975,0.000,5.730,0.000,
0.0529,1.000,0,0,5.730,-5.730,1,0,0,0,0/
80, 'EXC1',1,0,0,200,0.3575,1.000,0.011,4.2975,0.000,5.730,0.000,
0.0529,1.000,0,0,5.730,-5.730,1,0,0,0,0/
87, 'EXC1',1,0,0,0.05,20,1.00,1.980,2.385,0.0951,3.180,0.3712,0,0.1,
0,0,4.380,0,1,0,0,0,0/
89, 'EXC1',1,0,0,400,0.02,1.00,0.942,3.765,0.8147,5.020,2.6756,
0.030,1.000,0,0,18.30,-18.30,1,0,0,0,0/
100, 'EXC1',1,0,0,400,0.050,-0.17,1.950,3.6675,0.0111,4.890,0.0178,
0.040,1.000,0,0,3.810,-3.810,1,0,0,0,0/
103, 'EXC1',1,0,0,25,0.20,-0.0582,0.6544,2.5785,0.0889,3.438,0.3468,
0.105,0.350,0,0,1.00,-1.00,1,0,0.060,0,0/
111, 'EXC1',1,0,0,25,0.20,-0.0582,0.6544,2.5785,0.0889,3.438,0.3468,
0.105,0.350,0,0,1.00,-1.00,1,0,0.060,0,0/

10,'GOV8',1,0.9372,1,0.080,0,0.15,0.0058,0.050,10,2.8/
12,'GOV8',1,1.056,1,0.083,0,0.2,0.040,0.05,5,1.4/
25,'GOV8',1,1.050,1,0.1,0,0.4,0.0152,0.050,8,2/
26,'GOV8',1,0.8951,1,0.18,0,0.040,0.0122,0.25,8,2.136/
31,'GOV8',1,1.0,1,0.090,0,0.2,0.066,0.3,0,0/
46,'GOV8',1,1.0,1,0.090,0,0.2,0.066,0.3,0,0/
49,'GOV8',1,1.050,1,0.1,0,0.4,0.0152,0.050,8,2/
54,'GOV8',1,1.050,0.09,0,0.2,0.05,0.3,0,0/
59,'GOB8',1,0.901,0.15,0,0.1,0.0214,0.3,10,2.37/
61,'GOB8',1,0.901,0.15,0,0.1,0.0214,0.3,10,2.37/
65,'GOV8',1,0.898,0.15,0.05,0.3,0.0098,0.26,8,2.16/
66,'GOV8',1,0.898,0.15,0.05,0.3,0.0098,0.26,8,2.16/
69,'GOV8',1,0.9372,1,0.080,0,0.15,0.0058,0.050,10,2.8/
80,'GOV8',1,0.9372,1,0.080,0,0.15,0.0058,0.050,10,2.8/
87,'GOV8',1,1.0,1,0.090,0,0.2,0.066,0.3,0,0/
```

```
89,'GOV8',1,0.9177,0.18,0.03,0.2,0.006,0,8,2.4/
100,'GOV8',1,1.050,1,0.1,0,0.4,0.0152,0.050,8,2/
103,'GOV8',1,1.050,0.09,0,0.2,0.05,0.3,0,0/
111,'GOV8',1,1.050,0.09,0,0.2,0.05,0.3,0,0/
```

APPENDIX G

PYTHON FILE FOR GENERATE CONTINGENCY DATA

```
'''
Generate contingency file with a three phase fault on each
    bus and each line;
For fault on each bus: it is cleared after 5 cycles;
For fault on each line: the fault is at 5% of the line ,
near end of the line operate after 5 cycles , and far end of
    the line operate after 6 cycles
two contingencys are generated for the same line for from-bus
    and to-bus each;
Simulation time is 3 seconds;
Faults occur at 0.5 seconds;
CCT is 5 cycles.
'''

#------------------------Fault on Bus
    ------------------------------------

for num in range(1, 119):
    with open \
                ("C:/Users/qwang112/Desktop/Contingency/
                    fault_%d_118bus.swi" % num
                , 'w')as Contingency:
        Contingency.write('Description fault at bus ')
        bus = str(num)
        Contingency.write(bus + '\n')
        Contingency.write('Simulation 10.00 Seconds\n')
        Contingency.write('Integration RK4\n')
        Contingency.write('Step Size 0.25 Cycles\n')
        Contingency.write('At Time 0.5 Seconds\n')
        Contingency.write('Three Phase Fault At Bus ;')
        Contingency.write(bus + '\n')
        Contingency.write('After 5 Cycles\n')
        Contingency.write('Clear Three Phase Fault\n')
        Contingency.write('Nomore\n')
        Contingency.write('/\n')
        Contingency.write('END\n')

#------------------------Fault on lines
    ------------------------------------

#get from-bus and to-bus pair
FromBus = []
ToBus = []
ID = []
with open("C:\\Users\\qwang112\\Desktop\\IEEE 118 Bus for
    PSAT.raw",'r') as pf:
    lines = pf.readlines()
    ## Find branch Data Index
    for line in lines:
```

```
        if "BEGIN BRANCH DATA" in line:
            Idx_BrStart = lines.index(line) + 1

        elif "END OF BRANCH DATA" in line:
            Idx_BrEnd = lines.index(line)
            break

    for br in range(Idx_BrStart,Idx_BrEnd):
        temp = lines[br].split(",")
        FromBus.append(int(temp[0]))
        ToBus.append(int(temp[1]))
        string = temp[2]
        ID.append(int(string[1:-1]))

branch =Idx_BrEnd - Idx_BrStart
for i in range(0, branch):
    with open \
                ("C:/Users/qwang112/Desktop/Contingency/
                    fault_%d" %FromBus[i] \
                 + " to %d_118.swi" %ToBus[i], 'w') as
                     Contingency:
        Contingency.write('Description delayed fault
            clearance and bus name\n/\n')
        Contingency.write('Simulation 16.00 Seconds\n')
        Contingency.write('Step Size 0.005 Seconds\n')
        Contingency.write('Plot Every 5 Steps\n')
        Contingency.write('Report Every 99 Steps\n')
        Contingency.write('Integration RK4\n/\n')
        Contingency.write('At Time 0.5 Seconds\n')
        Contingency.write('Three Phase Fault On Line ; %d'%
            FromBus[i] +'; %d'%ToBus[i] + '; %d'%ID[i] + '
            5.0')
        Contingency.write('\n')
        Contingency.write('After 5 Cycles\n')
        Contingency.write('Clear Three Phase Line Fault At
            Near End\n')
        Contingency.write('\n')
        Contingency.write('After 1 Cycles\n')
        Contingency.write('Clear Three Phase Line Fault At
            Far End\n/\n')
        Contingency.write('Nomore\n')
        Contingency.write('/\n')
        Contingency.write('END\n')
    with open \
                ("C:/Users/qwang112/Desktop/Contingency/
                    fault_%d" %ToBus[i] \
```

```
        + " to %d_118.swi" %FromBus[i], 'w') as
            Contingency:
Contingency.write('Description delayed fault
    clearance and bus name\n/\n')
Contingency.write('Simulation 16.00 Seconds\n')
Contingency.write('Step Size 0.005 Seconds\n')
Contingency.write('Plot Every 5 Steps\n')
Contingency.write('Report Every 99 Steps\n')
Contingency.write('Integration RK4\n/\n')
Contingency.write('At Time 0.5 Seconds\n')
Contingency.write('Three Phase Fault On Line ; %d'%
    ToBus[i] +'; %d'%FromBus[i] + '; %d'%ID[i] + '
    5.0')
Contingency.write('\n')
Contingency.write('After 5 Cycles\n')
Contingency.write('Clear Three Phase Line Fault At
    Near End\n')
Contingency.write('\n')
Contingency.write('After 1 Cycles\n')
Contingency.write('Clear Three Phase Line Fault At
    Far End\n/\n')
Contingency.write('Nomore\n')
Contingency.write('/\n')
Contingency.write('END\n')
```

APPENDIX H

MONITOR DATA FILE

```
[TSAT 8.X Monitor]

{Generator}
Only, Rotor Angle
10, ' 1'
31, ' 1'
80, ' 1'
103, ' 1'
89, ' 1'
111, ' 1'
100, ' 1'
66, ' 1'
65, ' 1'
49, ' 1'
87, ' 1'
69, ' 1'
25, ' 1'
26, ' 1'
59, ' 1'
54, ' 1'
12, ' 1'
61, ' 1'
46, ' 1'
{End Generator}

{Region}
ID = '118bus'
Bus,20,
Bus,96,
Bus,85,
Bus,74,
Bus,9,
Bus,84,
Bus,108,
Bus,98,
Bus,10,
Bus,80,
Bus,95,
Bus,79,
Bus,78,
Bus,103,
Bus,89,
Bus,106,
Bus,23,
Bus,13,
Bus,117,
Bus,47,
```

```
Bus,111,
Bus,112,
Bus,76,
Bus,113,
Bus,31,
Bus,32,
Bus,37,
Bus,38,
Bus,110,
Bus,109,
Bus,88,
Bus,15,
Bus,100,
Bus,14,
Bus,29,
Bus,104,
Bus,33,
Bus,86,
Bus,3,
Bus,72,
Bus,99,
Bus,90,
Bus,91,
Bus,42,
Bus,7,
Bus,21,
Bus,64,
Bus,81,
Bus,6,
Bus,116,
Bus,19,
Bus,82,
Bus,27,
Bus,18,
Bus,115,
Bus,28,
Bus,66,
Bus,65,
Bus,16,
Bus,45,
Bus,71,
Bus,62,
Bus,51,
Bus,4,
Bus,39,
Bus,5,
Bus,8,
```

```
Bus,49,
Bus,87,
Bus,2,
Bus,70,
Bus,22,
Bus,107,
Bus,1,
Bus,105,
Bus,34,
Bus,43,
Bus,41,
Bus,52,
Bus,60,
Bus,92,
Bus,73,
Bus,102,
Bus,17,
Bus,30,
Bus,11,
Bus,69,
Bus,68,
Bus,83,
Bus,36,
Bus,75,
Bus,67,
Bus,97,
Bus,56,
Bus,94,
Bus,25,
Bus,26,
Bus,93,
Bus,63,
Bus,59,
Bus,54,
Bus,24,
Bus,77,
Bus,12,
Bus,61,
Bus,46,
Bus,50,
Bus,118,
Bus,44,
Bus,114,
Bus,57,
Bus,58,
Bus,55,
Bus,40,
```

```
Bus,35,
Bus,53,
Bus,101,
Bus,48,
{End Region}
```