

**PURDUE UNIVERSITY**  
**GRADUATE SCHOOL**  
**Thesis/Dissertation Acceptance**

This is to certify that the thesis/dissertation prepared

By Maryam Naghsh Nilchi

Entitled

ELECTRIC UTILITY PLANNING METHODS FOR THE DESIGN OF ONE SHOT STABILITY CONTROLS

For the degree of Master of Science in Electrical and Computer Engineering

Is approved by the final examining committee:

Steven Rovnyak

Chair

Yaobin Chen

Eliza Du

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Steven Rovnyak

Approved by: Yaobin Chen

Head of the Graduate Program

11/15/2012

Date

**PURDUE UNIVERSITY  
GRADUATE SCHOOL**

**Research Integrity and Copyright Disclaimer**

Title of Thesis/Dissertation:

ELECTRIC UTILITY PLANNING METHODS FOR THE DESIGN OF ONE  
SHOT STABILITY CONTROLS

For the degree of Master of Science in Electrical and Computer Engineering

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Executive Memorandum No. C-22*, September 6, 1991, *Policy on Integrity in Research*.\*

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Maryam Naghsh Nilchi

\_\_\_\_\_  
Printed Name and Signature of Candidate

11/15/2012

\_\_\_\_\_  
Date (month/day/year)

\*Located at [http://www.purdue.edu/policies/pages/teach\\_res\\_outreach/c\\_22.html](http://www.purdue.edu/policies/pages/teach_res_outreach/c_22.html)

ELECTRIC UTILITY PLANNING METHODS FOR THE DESIGN OF ONE  
SHOT STABILITY CONTROLS

A Thesis

Submitted to the Faculty

of

Purdue University

by

Maryam Naghsh Nilchi

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Electrical and Computer Engineering

December 2012

Purdue University

Indianapolis, Indiana

## ACKNOWLEDGMENTS

This thesis would not have been possible without the guidance and assistance of several individuals who in one way or another contributed their valuable help in the preparation and completion of this study.

First and foremost, my greatest appreciation to my research advisor, Dr. Steven M. Rovnyak, whose genuineness and encouragement I will never forget. I will always be thankful for his excellent guidance, patience, and support in the completion of this research work. One simply could not wish for a better or friendlier supervisor.

I would also like to express gratitude to my advisory committee members, Dr. Yaobin Chen, and Dr. Eliza Y. Du for their time and support during the completion of this thesis.

I wish to express my sincere appreciation to Dr. Brian King who provided me with brilliant guidance during the completion of this degree. He encouraged me to take my further steps and played a great role in the financial aspect of my graduate study.

I would like to thank Sherrie Tucker for her kindness in formatting this thesis and for keeping me in mind for every upcoming opportunity in the ECE Department.

I would also like to extend my gratefulness to Valerie Lim Diemer and other ECE faculty members who assisted me in different aspects to move my master's study forward.

I would like to acknowledge the numerous individuals who have affected my educational experience at IUPUI, including but not limited to my dear lab partner Diana C. Vasquez and other graduate students. In my daily work I have been blessed with a friendly and cheerful group of fellow students and without them, this research could not be achieved at such a level.

Finally, I would like to thank my husband Shahriar whose personal support and unlimited patience at all times helped me make this research through calmly to the end.

Last but not least, I would like to thank my parents who have given me their indisputable support throughout, as always, for which my meager expression of appreciations likewise does not suffice.

## TABLE OF CONTENTS

	Page
LIST OF TABLES . . . . .	vi
LIST OF FIGURES . . . . .	vii
LIST OF ABBREVIATIONS . . . . .	ix
ABSTRACT . . . . .	x
1 INTRODUCTION . . . . .	1
1.1 Problem Statement . . . . .	1
1.2 Previous Work . . . . .	4
1.3 Tools of This Thesis . . . . .	4
1.4 About This Thesis . . . . .	6
2 PROPOSED METHOD FOR INSTABILITY THRESHOLD IDENTIFICATION . . . . .	9
2.1 Background . . . . .	9
2.2 The Histogram Method . . . . .	11
2.3 The Plotting Method . . . . .	12
3 INDICES USED IN DATA CLASSIFICATION . . . . .	16
3.1 ISGA, an Index of Previous Researches . . . . .	16
3.2 Calculation of ISBA Index . . . . .	17
3.3 Gradient of the ISBA . . . . .	20
3.4 Rate of Change of Bus Angles . . . . .	20
4 DECISION TREES . . . . .	21
4.1 Overview . . . . .	21
4.2 Rattle Parameters . . . . .	22
4.3 DT Data . . . . .	24
5 ONE SHOT CONTROL WITH RESPONSE BASED DECISION TREE . . . . .	25

	Page
5.1 Overview . . . . .	25
5.2 Application of One shot Control . . . . .	25
5.2.1 Location and Time of Control . . . . .	26
5.3 Results . . . . .	27
6 PROPOSED METHOD USING THREE DECISION TREES . . . . .	30
6.1 Overview . . . . .	30
6.2 DT1: Event Detection . . . . .	30
6.3 DT2: Instability Prediction . . . . .	32
6.4 DT3: Control Selection . . . . .	33
6.5 Comparison Between 1DT, 2DT, and 3DT Methods . . . . .	36
7 ROLE OF CONTROL APPLICATION IN STEADY-STATE ANALYSIS	40
8 CONCLUSIONS AND FUTURE WORK . . . . .	43
8.1 Conclusions . . . . .	43
8.2 Future Work . . . . .	43
LIST OF REFERENCES . . . . .	45
APPENDIX: MATLAB CODE FOR POWER SYSTEM SIMULATIONS . .	47

## LIST OF TABLES

Table	Page
5.1 Results of response based DT method with application of control case 1 to 385 cases . . . . .	28
5.2 Results of response based DT method with application of control case 2 to 385 cases . . . . .	29
6.1 Results of the 2DTs method tested on 385 cases . . . . .	34
6.2 Different combinations of complexity cost and loss matrix used in development of the control selection rules . . . . .	35
6.3 Results of the 3DTs method tested on 385 cases . . . . .	37



## LIST OF FIGURES

Figure	Page
1.1 Rotor angle stability . . . . .	3
1.2 Transmission lines retained in the 176-bus model of the WECC [1]. . .	7
1.3 Three decision tree system . . . . .	8
2.1 Maximum GA difference for 385 simulations. . . . .	10
2.2 Maximum GA difference for 385 simulations. . . . .	11
2.3 Histogram of maximum GA differences for 6 seconds of mostly stable sim- ulations. . . . .	12
2.4 Close view of histogram of maximum GA differences for mostly stable simulations. . . . .	13
2.5 Close view of distribution of maximum GA differences for the first 5 sec- onds of mostly stable simulations. . . . .	14
2.6 Maximum GA difference for all 6 seconds of mostly stable simulations vs. the threshold for determining the mostly stable simulations. . . . .	14
2.7 Maximum GA difference for 5 seconds of mostly stable simulations vs. the threshold for determining the mostly stable simulations. . . . .	15
3.1 Discontinuity of Bus Angles . . . . .	18
3.2 Reconstructed Bus Angles . . . . .	18
3.3 SBA and ISBA for the Trajectories in Figure 3.2 . . . . .	19
3.4 ISBA and Gradient of ISBA . . . . .	20
4.1 Training data from 480 simulations in ISBA-GradISBA space . . . . .	22
4.2 DT Example . . . . .	23
5.1 Timing of control application for 30 unstable cases . . . . .	27
5.2 Number of cases stabilized with control applied at different times . . .	28
6.1 Rules for event detection DT . . . . .	31
6.2 Comparison of the results of the 1DT, 2DTs, and 3DTs method tested with 385 cases . . . . .	36

Figure	Page
6.3 Comparison between the results of this thesis, tested on 385 cases, with the approximated 2DT results in [13], tested on 480 cases . . . . .	39
7.1 Post event equilibrium in stable cases of the 385 cases set . . . . .	41
7.2 Post event equilibrium in stable cases of the 385 cases set . . . . .	41
7.3 Post event equilibrium in stable cases of the 385 cases set . . . . .	42

## LIST OF ABBREVIATIONS

BA	Bus Angle
COA	Center Of Angle
DSA	Dynamic Security Assessment
DT	Decision Tree
EMS	Energy Management System
FP	False Positive
FN	False Negative
GA	Generator Angle
HVDC	High Voltage Direct Current
ISBA	Integral Square Bus Angle
ISGA	Integral Square Generator Angle
MISO	Midwest Independent System Operators
NERC	North American Electric Reliability Corporation
PMU	Phasor Measurement Units
PSAT	PowerFlow and Short circuit Assessment Tool
Rattle	R Analytical Tool To Learn Easily
SBA	Squared Bus Angles
SCADA	Supervisory Control and Data Acquisition
TN	True Negative
TP	True Positive
TPL	Transmission Planning
TSAT	Transient Security Assessment Tool
WAMS	Wide Area Measurement Systems
WECC	Western Electric Coordinating Council

## ABSTRACT

Naghsh Nilchi, Maryam. M.S.E.C.E., Purdue University, December 2012. Electric Utility Planning Methods for the Design of One Shot Stability Controls. Major Professor: Steven M. Rovnyak.

Reliability of the wide-area power system is becoming a greater concern as the power grid is growing. Delivering electric power from the most economical source through fewest and shortest transmission lines to customers frequently increases the stress on the system and prevents it from maintaining its stability. Events like loss of transmission equipment and phase to ground faults can force the system to cross its stability limits by causing the generators to lose their synchronism. Therefore, a helpful solution is detection of these dynamic events and prediction of instability.

Decision Trees (DTs) were used as a pattern recognition tool in this thesis. Based on training data, DT generated rules for detecting event, predicting loss of synchronism, and selecting stabilizing control. To evaluate the accuracy of these rules, they were applied to testing data sets.

To train DTs of this thesis, direct system measurements like generator rotor angles and bus voltage angles as well as calculated indices such as the rate of change of bus angles, the Integral Square Bus Angle (ISBA) and the gradient of ISBA were used.

The initial method of this thesis included a response based DT only for instability prediction. In this method, time and location of the events were unknown and the one shot control was applied when the instability was predicted. The control applied was in the form of fast power changes on four different buses. Further, an event detection DT was combined with the instability prediction such that the data samples of each case was checked with event detection DT rules. In cases that an event was detected, control was applied upon prediction of instability.

Later in the research, it was investigated that different control cases could behave differently in terms of the number of cases they stabilize. Therefore, a third DT was trained to select between two different control cases to improve the effectiveness of the methodology.

It was learned through internship at Midwest Independent Transmission Operators (MISO) that post-event steady-state analysis is necessary for better understanding the effect of the faults on the power system. Hence, this study was included in this research.

# 1. INTRODUCTION

## 1.1 Problem Statement

The increasing dependence of our daily lives on consistent supply of electricity makes the reliability of the power system extremely vital. Reliability is mainly composed of two parts: Adequacy and Security. Adequacy is the capability of the power grid to serve the electrical load and all their customers requirements at all-time along with considering the planned and reasonable outages. Security is the capability of the system to tolerate the sudden contingencies and loss of transmission equipment [16]. Steady-state and transient stability analysis are major components of system security. Steady state stability of the system is referred to a situation in which, after a small contingency, the system can reach a steady state condition close to its pre-contingency operating situation. Whereas, the transient stability of the system is referred to a condition in which the system can reach any acceptable steady state situation and maintain its synchronism after a large disturbance [16].

The power stability problem became more important when remote hydraulic stations started serving loads through long distance transmission lines. The blackouts of Northern parts of the US which happened in 1965 and 1967 highlighted the importance of effective modeling and planning in the power industry [16]. As the wide area power system is growing smarter and the use of system measurement units is increasing, monitoring and supervision of the grid behavior is becoming easier. Wide Area Measurement Systems (WAMS) on the system allow to retrieve (nearly) real-time data from the system and analyze its dynamic behavior. This is done by detecting events and constraints, predicting loss of synchronism and taking on time accurate actions to bring the system back into its stable situation. The purpose and conditions of fault detection and location for protective relaying are different from dynamic

event detection and location. Fault detection for protective relaying is typically done with local measurements to isolate a faulted element. System-wide event detection is more likely implemented using measurements with more focus on investigation, restoration and control [3]. During my internship at MISO, I learned that although the dynamic events and the transient instabilities in the power system are rare on an everyday basis, their analysis engage more complex non-linear models and techniques which involve the steady-state condition that the system reached following the transient [16]. Dynamic events are of significant importance and if they happen, their following consequences can be severe.

In general, there are two categories of dynamic stability issues in the power system. One category is stated as the rotor angle stability and is affected by the generator rotor angles and power angles relationship [16]. This type of instability is the major focus of this thesis. The other type of dynamic stability issue, stated as the Voltage Stability, occurs without loss of synchronism and could happen if a bus in the system experiences voltage collapse. The concern of the system at such situation is to maintain the voltage of all the buses within an acceptable range after experiencing disturbances [16].

In this research, the bus voltage angles, which are retrieved from the existing Phasor Measurement Units (PMUs), along with the generator rotor angles are used to analyze the overall system behavior. Monitoring six seconds of simulation with more than 720 sample points, it is observed that some of the generator angles start to diverge and lose their synchronism as they move toward the second half of simulation where this forces the system to go unstable. Figure 1.1 represents examples of two cases. The first case shows a generator which remains synchronous after experiencing a fault where the second case represents a situation where the system loses its rotor angle stability after undergoing a fault.

To prevent the system from going unstable, response based control method is used which continuously monitors the retrieved real-time data and applies one shot controls when a loss of synchronism is predicted in the system.

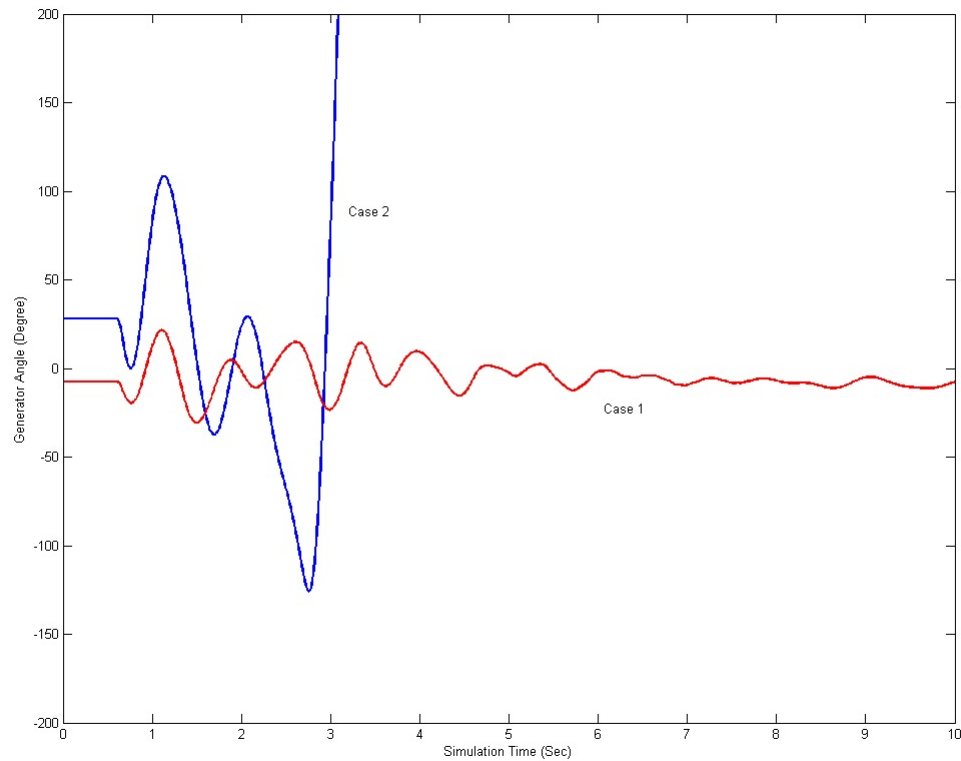


Fig. 1.1. Rotor angle stability

A one shot control includes fast power changes on four buses which are similar to 500 MW fast power increases on two HVDC interties and reduces angle differences in the AC network [1].

According to [2], one shot control is an arrangement that applies control utmost once during an event using the measurements prior to the control. When the decision is made and the control is operated, one shot does not use any measurements until it is reset.



## 1.2 Previous Work

Application of response based one shot controls for maintaining the stability and synchronism of WAMS have been the main focus of previous researches in this area.

Pattern recognition tools such as Decision Trees (DTs) were employed to develop rules from training data sets using different system variables/indices. In [1], initial classification of stable and unstable cases was done with a 300-degree threshold for the maximum generator angle difference. This provided the target value of 0 for Unstable cases and target value of 1 for Stable cases. In [1] it is shown that a larger number of cases could be stabilized when a target value was assigned to every input output pair.

Various indices implemented by different researchers for data classification include: voltage angle of one bus and rate of voltage angle change of another bus [1], Integral Square Generator Angle (ISGA) [4], wavelet detail coefficients [5], apparent resistance [2], [7], a performance index of weighed sum squared error between the simulated behavior and the ideal trajectory of generator angles [6], and indices proposed for Dynamic Security Assessment (DSA) [8], [9].

These previous approaches were all predicting instability without knowing the time and location of the faults. Finding a way to detect dynamic event first and then check for instability after the event occurrence may increase the accuracy of the outcome. This was done in [8] employing the variance of generator rotor frequencies in a sliding window frame as the index.

## 1.3 Tools of This Thesis

Following tools were used interactively to achieve the objectives of this thesis:

a) Transient Security Assessment Tool (TSAT) is a software tool established by Powertech Labs Inc., for transient analysis of power systems. Its main technology is a nonlinear time-domain simulation engine which has the ability to produce precise responses to various types of contingencies happening in large interconnected power

systems. It has a set of security assessment modules hard coded in it that implement complete analysis for the dynamic behavior of the system according to NERCs TPL standards [12]. This tool was used for transient stability analysis of our 176-buses model of Western Electric Coordinating Council (WECC) power grid. Details of this model are presented in section 1.4.

b) PowerFlow and Short circuit Assessment Tool (PSAT) is another software tool by Powertech Labs Inc., used in power system studies. It is a graphical program for creating powerflow models [20]; An important capability of PSAT is to perform steady state system analysis needed in system planning studies [20]. After event Steady state powerflow, also known as thermal loading, of the transmission lines is the main analysis that PSAT was used for in this thesis.

c) MATLAB which is an advanced technical computing language and collaborative environment for data analysis and numerical computation [14] was used to call TSAT to do specified simulations, retrieve particular parts of the simulation results from TSAT, calculate the desired indices for preparing and documenting training data for our DT tool and ; and analyze the results of the study. MATLAB allowed us to increase the speed of our calculations and computations by automating the process of running hundreds of simulations.

d) The R Analytical Tool To Learn Easily (Rattle) is a pattern recognition and data mining toolbox that brings multiple R packages together and has significantly improved the process of complicated data analysis by using statistical methods; Rattle is a user friendly interface where no initial knowledge of R is needed to get started [11]. In this thesis, Rattle was mainly used to find thresholds for event detection, instability prediction, and control selection. Chapter 4 presents more details about background of DT method and the characteristics involved.

## 1.4 About This Thesis

Main objectives of this thesis are: 1. Understand the previous work of Rovnyak et al. in improving the reliability of the power system through instability prediction and application of response based one shot control; 2. Expand their work by finding new indices that can be observed in nearly real-time and employ them in the DT methodology to find triggering threshold; 3. Develop dynamic event detection and control selection methods and use them in combination with the instability prediction method; 4. Find the best time to apply control; all with the purpose of improving the accuracy of one shot control approach.

The power system used in this thesis was a 176-bus model of the Western Electric Coordinating Council power system where 17 buses had PMUs, 71 were load buses, and 29 were generator buses. A simplified equivalent of the system is shown in Figure 1.2. Present technology feeds wide-area synchronized phasor measurements into SCADA and EMS systems at a level of 60 cycles per second [3]. Simulations of six seconds were done with the rate of 60 cycles per second and 2 steps per cycle; providing us with about 120 sample points in each second. Rotor angles of 29 generator buses and bus voltage angles of 17 load buses monitored by the PMUs were the measurements that were retrieved from TSAT and used in this thesis.

Two sets of contingency files were used in this thesis. The first set included a total of 385 cases where faults with different lengths happened on 40 lines of the model. 160 cases out of 385 were solo event cases of either a single phase to ground or three phase short circuit faults; the other 225 cases were multiple event cases where a single phase fault on one line was followed by a three phase fault on another line. 263 cases remained stable after fault and instability happened in 122 cases. The second set included 480 cases of three phase faults starting somewhere between 29th and 40th cycle of the simulation; all ending at the 40th cycle, giving 12 different scenarios for each of the 40 lines. This set included 250 stable and 230 unstable cases.

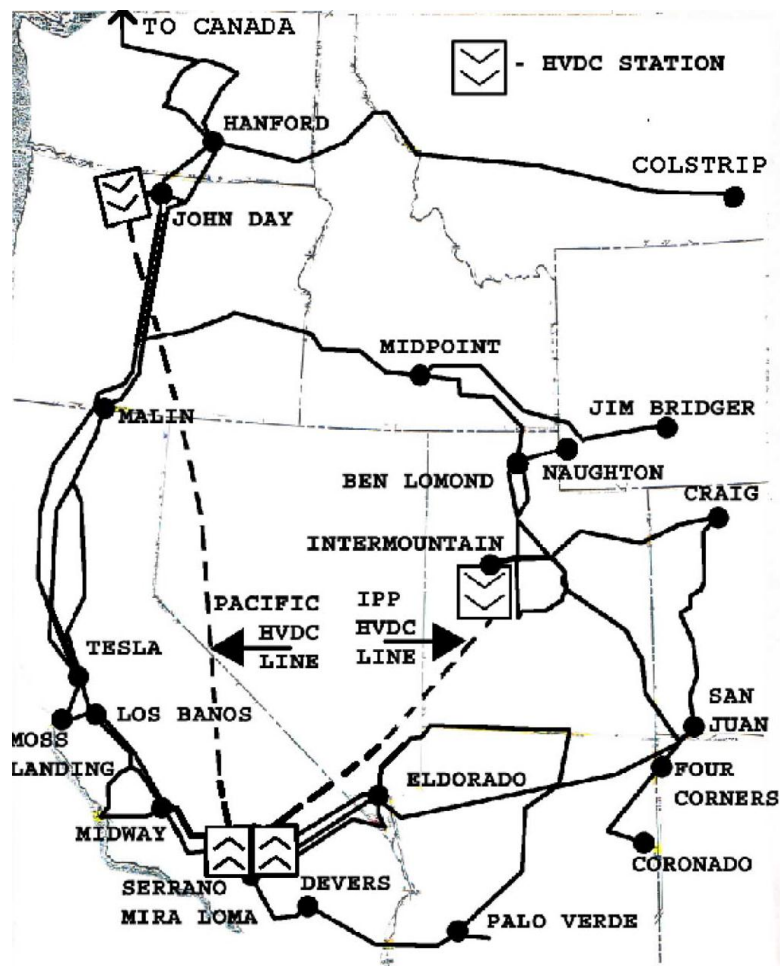


Fig. 1.2. Transmission lines retained in the 176-bus model of the WECC [1].

Simulating the event cases, an initial classification between stable and unstable cases was done where a case is counted unstable when loss of synchronism happened in one generator. A measure for this was when the maximum difference between angles of every two generators exceeds 300 degrees. Details of how this threshold was selected are provided in Chapter 2. After this initial classification, the data was fed into three DTs for dynamic event detection, instability prediction and control selection. An overall view of this approach is shown as a flowchart in Figure 1.3.

During my internship at Midwest Independent Transmission Operators (MISO), I learned not only about how the power grid is operated in high voltage levels, but

also I got involved with the transmission planning concerns. I learned that steady-state analysis is a great concern, because if the steady-state situation after a dynamic event recovery is not monitored, voltage violations and power over loads could lead into cascading contingencies that would put the reliability of the grid in danger. To have this considered in this thesis, the power flow of three main transmission lines during their after fault steady state were monitored and the results are later presented in this report.

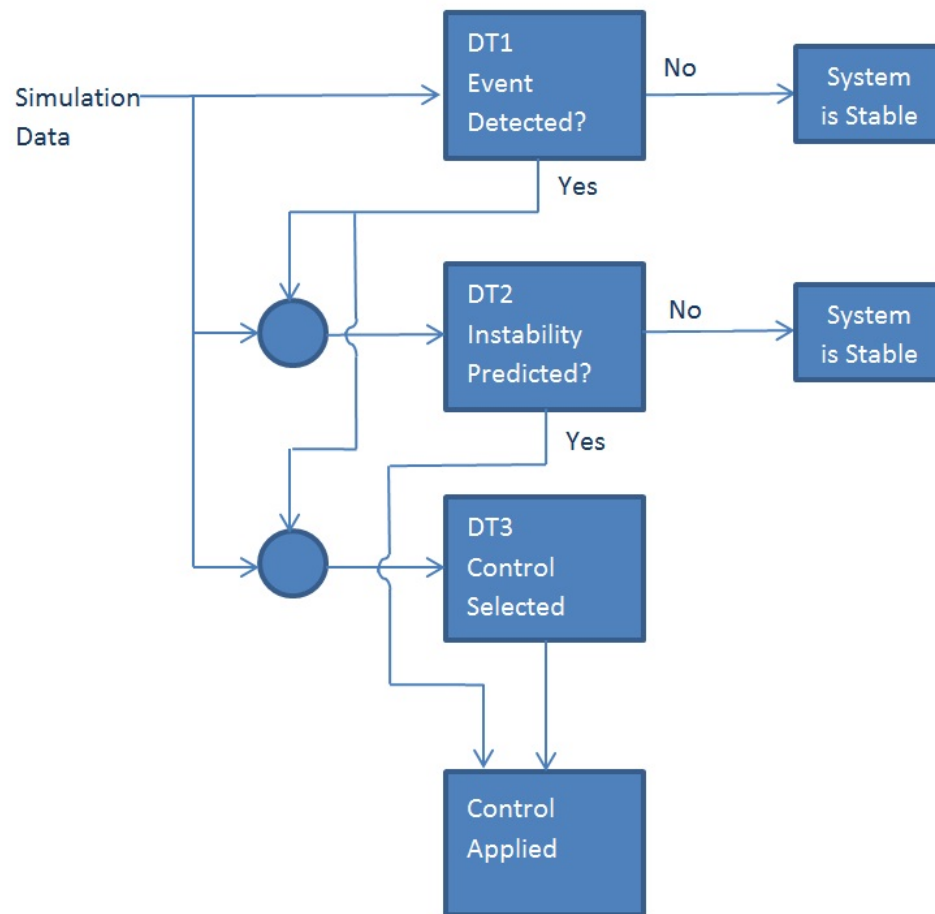


Fig. 1.3. Three decision tree system

## 2. PROPOSED METHOD FOR INSTABILITY THRESHOLD IDENTIFICATION

### 2.1 Background

This section describes some approaches that can be used to define thresholds for initial classification of truly stable and unstable cases using the maximum difference between generator angles (GAs).

According to [15], finding a threshold from simulation data can be difficult knowing that there exist some events that experience loss of synchronism right around the end of our six seconds simulation time frame. This occurrence along with a method to work this out are illustrated for a maximum GA index. Figure 2.1 shows the maximum GA difference in a six seconds window for 385 simulations on the 176-bus model. It is possible to classify every simulation that exceeds the threshold as unstable by selecting a large enough threshold value around 360 degrees whereas the purpose of this chapter is to explain the method for defining a lower threshold value that can help the instability to be detected earlier in the simulation, [15]. Figure 2.1 shows that the maximum angle difference for simulations that keep their synchronism and do not go beyond the 360 degree limit is about 340 degrees. There exists only one simulation that the maximum GA difference touches 340 degrees at the very end of the six-second simulation period. Figure 2.1 expresses that this simulation is about to lose its synchronism, but the 360 degree threshold categorizes it as stable which is a mistake, [15].

A lower threshold value could be defined if simulations like this were properly classified as unstable. Similar cases with loss of synchronism toward the end of the simulation period are very possible and their misclassification would lead into further constraints.

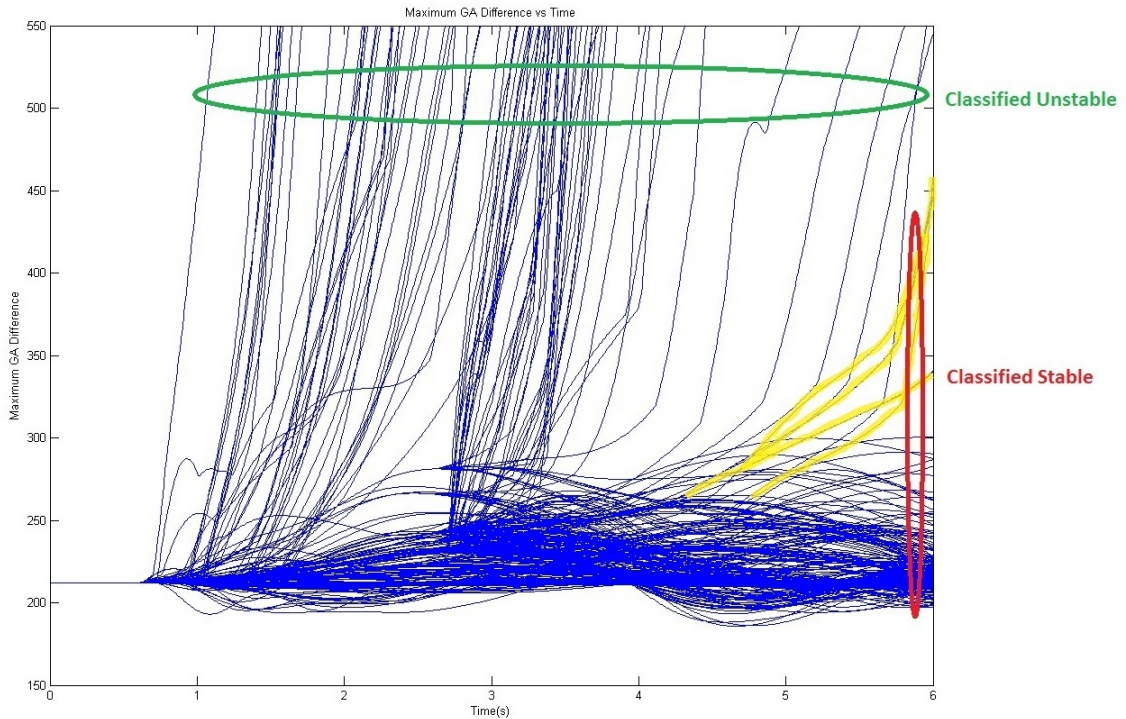


Fig. 2.1. Maximum GA difference for 385 simulations.

The purpose is to find a way to define a more precise threshold for classification of stable and unstable cases while the exact threshold is not known beforehand. The method presented in this chapter includes distinguishing the problem happening in simulations which begin to lose synchronism toward the end of the simulation. One solution could be to extend the simulation for a longer time period that loss of synchronism would never happen at the end. An example of this is shown in Figure 2.2 where a simulation is done for 10 seconds. Possible issues with this solution are that it is time consuming and not very feasible. Although it rarely happens that a simulation loses synchronism after 10 or 20 seconds, there exists a possibility for it to take place in large data sets. Small intentional adjustments to a borderline instability can generate such cases.

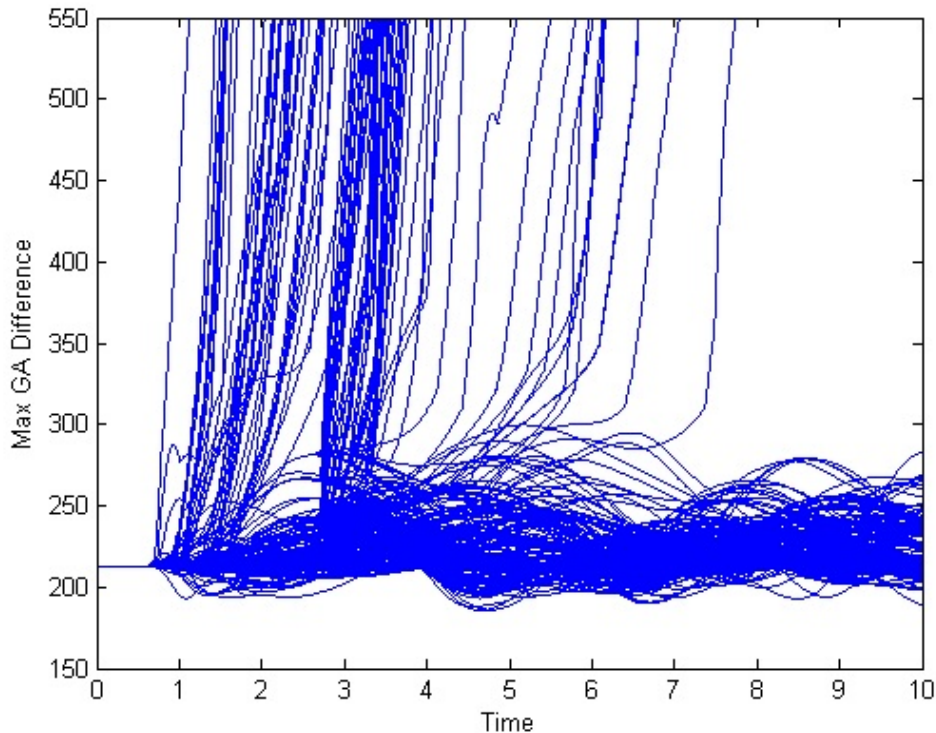


Fig. 2.2. Maximum GA difference for 385 simulations.

## 2.2 The Histogram Method

The technique suggested here includes selecting a conservative evaluation of instability like 360 degrees as a first step to produce a data set of only truly stable simulations [15]. The next step includes ignoring parts of data that stayed within the 360 degree limit during the last few seconds of the simulations, [15]. To have this better visualized here, the subset of simulations with a maximum GA difference of smaller than 500 degrees during the first six seconds were selected as the set of mostly stable simulations. This measure might be suitable for a power system with transmission lines extended over a large geographical area with large power flows from one end to the other. Figure 2.3 shows the histogram of the maximum GA difference from every time step of six second simulations of the mostly stable cases, [15]. An enlarged view of this histogram is shown in Figure 2.4. Figure 2.4 does not clearly



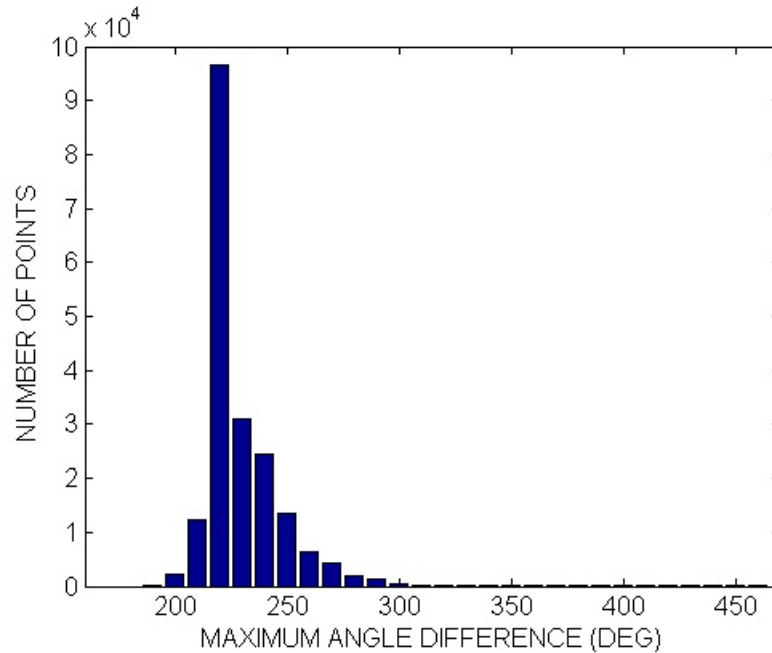


Fig. 2.3. Histogram of maximum GA differences for 6 seconds of mostly stable simulations.

express where in the array of 300 to 360 degrees the instability threshold should be identified. Observing the maximum GA differences of the mostly stable simulations only in the first 5 seconds of each simulation (removing the last second of data from each simulation) can reduce the uncertainty as shown in Figure 2.5. Explicitly, it is still checked that whether the maximum GA difference in the set of mostly stable simulations surpasses 500 degrees in the first 6 seconds. Yet, only the maximum GA differences during the first five seconds of these simulations are shown in Figure 2.5. Figure 2.5 justifies that the instability threshold is closer to 300 degrees than to 360 degrees, [15].

### 2.3 The Plotting Method

Another way to justify the results found above is by plotting the maximum GA difference of the mostly stable simulations using varying thresholds [15]. As shown

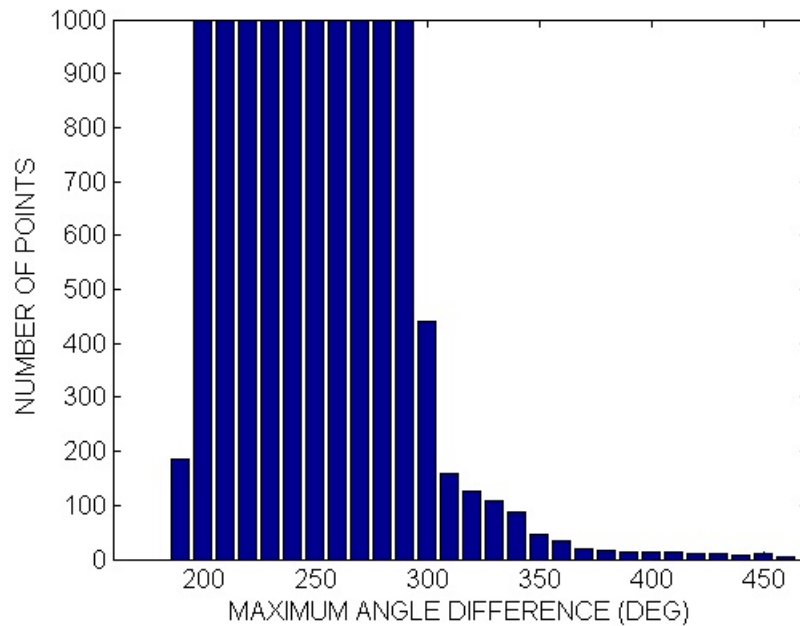


Fig. 2.4. Close view of histogram of maximum GA differences for mostly stable simulations.

in Figure 2.6, the threshold for maximum GA difference used to outline the set of mostly stable simulations differed from 100 to 500 degrees in 25 degree intervals. Those simulations with maximum GA difference smaller than the threshold during the first 6 seconds were counted toward the mostly stable simulations. The vertical axis in Figure 2.6 displays the maximum GA difference among these sets of mostly stable simulations through the first 6 seconds. Similar to Figure 2.4, Figure 2.6 is not clear on where exactly the instability threshold would lie between 300 and 360 degrees. Figure 2.7 presents the same approach as Figure 2.6 does but the maximum GA difference was only shown during the first 5 seconds of mostly stable simulations. The method in both Figures, 2.6 and 2.7, was to check whether the maximum GA differences during the first 6 seconds surpassed the threshold on the horizontal axis. The difference is that after the set of mostly stable simulations was determined, in order to find the value on the vertical axis in Figure 2.7, the last second of simulation data was ignored, [15]. Figure 2.7 justifies that a threshold of about 310 degrees

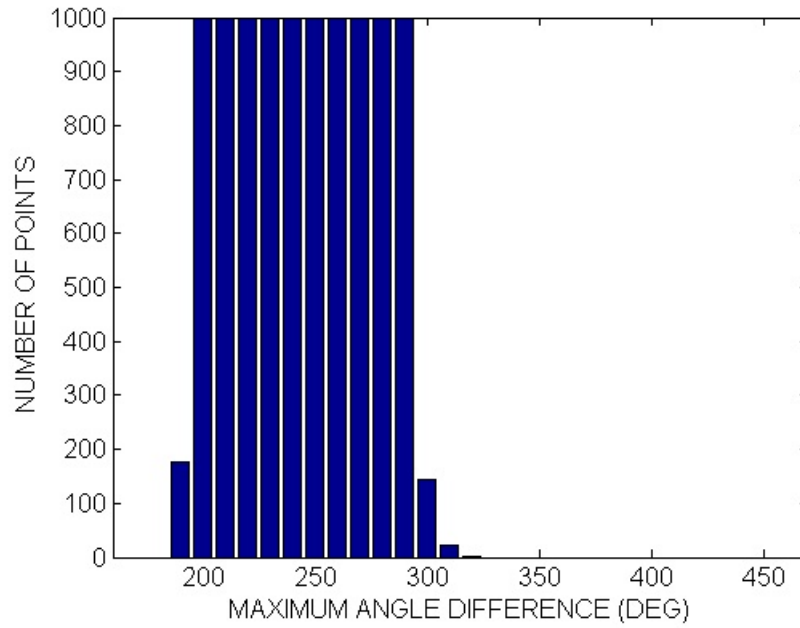


Fig. 2.5. Close view of distribution of maximum GA differences for the first 5 seconds of mostly stable simulations.

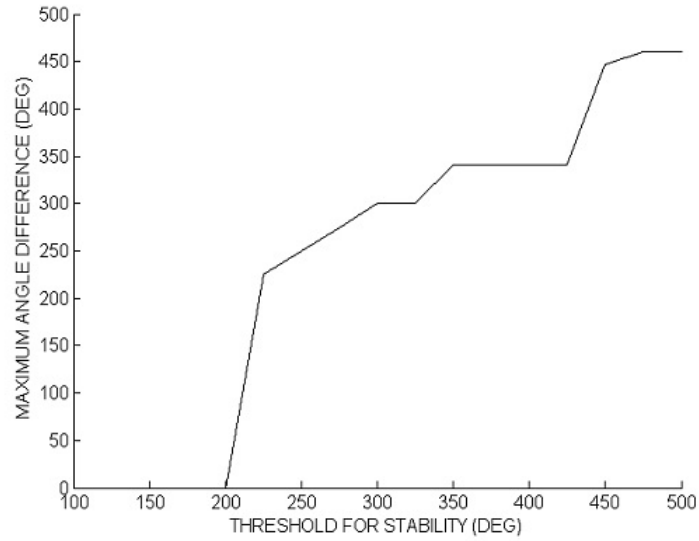


Fig. 2.6. Maximum GA difference for all 6 seconds of mostly stable simulations vs. the threshold for determining the mostly stable simulations.

is reasonable to classify the truly stable simulations from the unstable simulations. The same procedure could be practical to other further calculated angle stability indices such as the ISBA and ISGA. Details on how these two indices are calculated are presented in chapter 3. Implementing the new indices, the threshold of 360 or even 500 degrees for maximum GA difference will be used to define the set of mostly stable simulations during the first 6 seconds. Then, the maximum ISBA during the first 5 seconds of those simulations will be the threshold for the ISBA used for angle instability prediction [15].

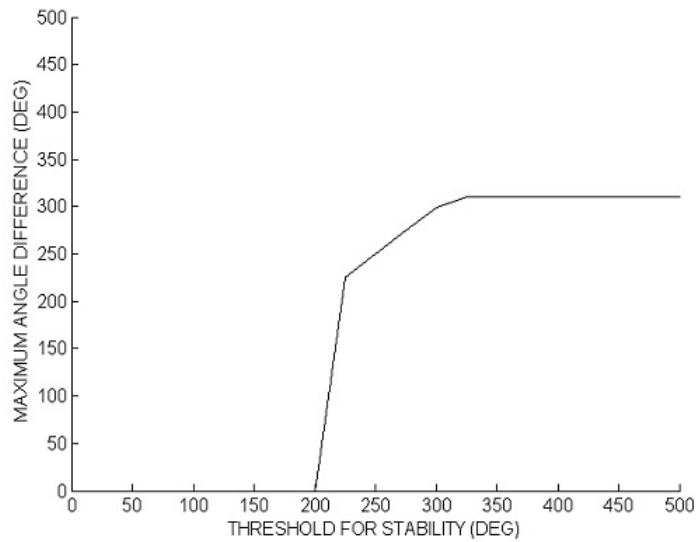


Fig. 2.7. Maximum GA difference for 5 seconds of mostly stable simulations vs. the threshold for determining the mostly stable simulations.

### 3. INDICES USED IN DATA CLASSIFICATION

Aside from the individual generator angles and bus voltage angles obtained from the system measurements, the following indices are calculated and implemented in event detection, instability prediction, and control selection decision making procedures of this thesis.

#### 3.1 ISGA, an Index of Previous Researches

Using system measurements, many different indices have been calculated by researchers to help with the transient stability analysis of power system. Time domain solutions are only capable of providing yes or no output to the transient stability analysis while they could be combined with indices like energy functions [4]. The Integral Square Generator Angle (ISGA), calculated as Equation (3.1), is a coherency based index used for evaluating the severity of the stable/unstable events [4]. Coherency based means that the index indicates the level of coherency [15]. ISGA is a measure for total generator angle differences in transient and steady-state situations [4].

$$ISGA = \int_0^T \sum_i M_i (\delta_i(t) - \delta_{coa}(t))^2 dt. \quad (3.1)$$

Where  $M_i$  stands for the machine inertia,  $\delta_i(t)$  shows generator angles as a function of time, and  $\delta_{coa}(t)$  is the Center of Angle (COA) calculated as Equation (3.2). Generators farther from the COA are penalized more by the ISGA [4].

$$\delta_{coa}(t) = \frac{\sum_i M_i \delta_i(t)}{\sum_i M_i}. \quad (3.2)$$

Further in previous researches, ISGA is normalized by simulation length. Generators disconnected from the system were not included in calculation of COA and ISGA.

Similar to the ISGA index, the Integral Square Bus Angle (ISBA) index is calculated in this research.

### 3.2 Calculation of ISBA Index

Bus voltage angles from synchronized PMUs are used in calculation of ISBA. Using the phase angle differences, ISBA is used to express the overall stress on the system [15] and is a good measure for instability prediction and loss of synchronism control design. ISBA was selected over ISGA because in real-time, direct measurement of the generator rotor angles is not possible where as the bus voltage angles are easily monitored by existing phasor measurement units.

The bus voltage angles are discontinuous by nature and this issue should be resolved before calculating the ISBA. Unlike the GAs, the BAs do not go beyond the -180 to 180 degrees range [15]. As shown in Figure 3.1, if a BA exceeds either +180 or -180 degrees, it wraps around to the opposite side causing a 360 degrees difference. To resolve this problem and help the BAs remain continuous, the bus angles should be unwrapped. The first step to reconstruct the bus angles is to add or subtract 360 degrees from the bus angle after each discontinuity. To find the points of discontinuity, each two consecutive bus angle sample points are compared; if the difference exceeds a preselected threshold, 360 degrees is subtracted from each and every angle measurement after the point of discontinuity. Similarly, if the difference is smaller than the negative amount of the selected threshold, 360 degrees is added to later measurements [13].

To select an appropriate threshold for reconstructing the bus angles, the method of trial and error was used. As the bus angles wrap around  $\pm 180$ , the difference of two consecutive bus measurements will never exceed 360 degrees. This makes the thresholds of 360 degrees or larger to be ineffective. Therefore, threshold of 330 is used in this thesis to find the points of discontinuity. Then, the discontinuity was resolved by adding or subtracting 360 degrees to the measurements of that time step

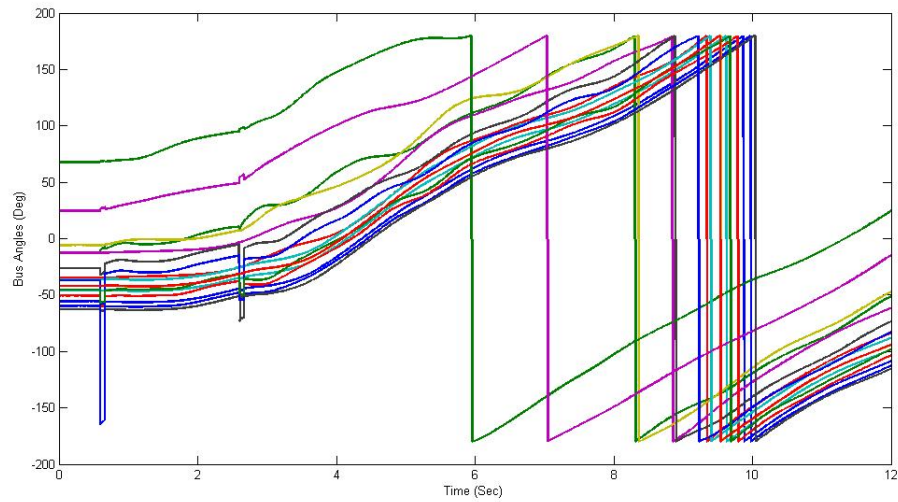


Fig. 3.1. Discontinuity of Bus Angles

and the measurements after that point. No change was made to the data points where the bus angles were changing with consecutive difference less than 330 degrees. Figure 3.2 shows the restructured form of the trajectories shown in Figure 3.1.

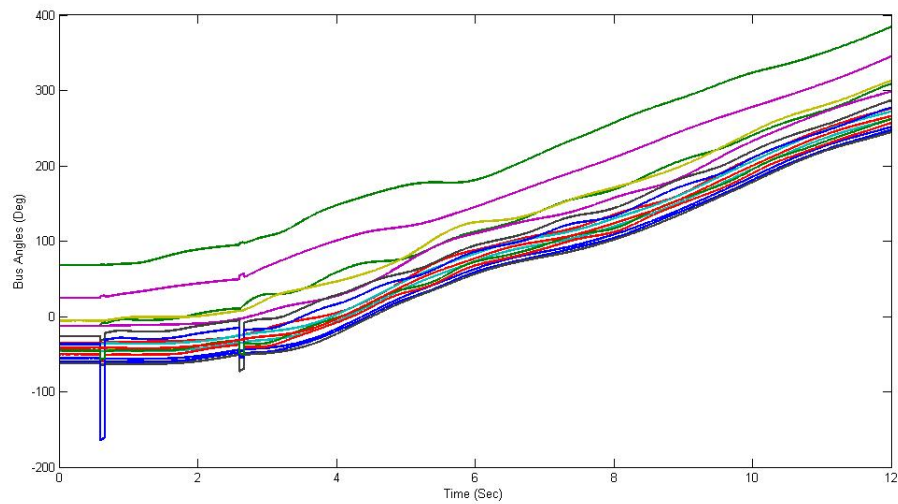


Fig. 3.2. Reconstructed Bus Angles

Having the discontinuity issue resolved, the calculation of ISBA can be performed. First, the Squared Bus Angles Index (SBA) is calculated as Equation (3.3) [15].

$$SBA = \sum_i M_i (\theta_i(t) - \theta_{coa}(t))^2, \quad (3.3)$$

where

$$\theta_{coa}(t) = \frac{\sum_i M_i \theta_i(t)}{\sum_i M_i}. \quad (3.4)$$

$\theta_i(t)$  represents the reconstructed BAs as a function of time,  $\theta_{coa}(t)$  is the COA, and  $M_i$  is chosen as the weight assigned to each bus. In this thesis, each bus is assigned an equal weight of one. Instead of calculating the integral of SBA over a sliding window, ISBA is calculated using a low pass filter with a transfer function  $G(s) = 6/(s + 6)$ . The corner frequency of this filter is 1Hz or 6 Radians/Sec [15]. Figure 3.3 shows the SBA and ISBA of the trajectories of Figure 3.2.

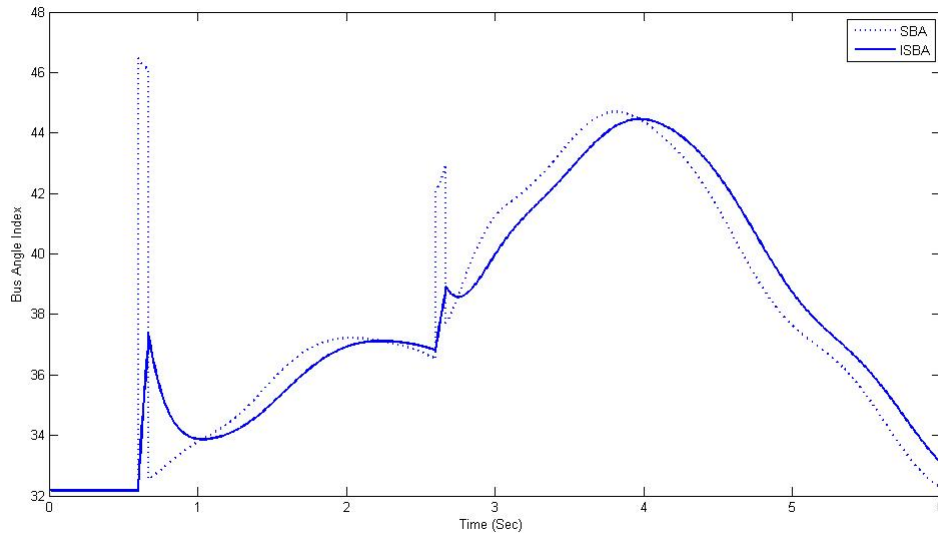


Fig. 3.3. SBA and ISBA for the Trajectories in Figure 3.2



### 3.3 Gradient of the ISBA

Gradient of the ISBA was calculated as another index in this research. This was done by obtaining the point to point difference between the available data points. Because a lowpass filter was used in calculation of ISBA from SBA, gradient of ISBA did not equal SBA [13]. Figure 3.4 compares ISBA and its gradient.

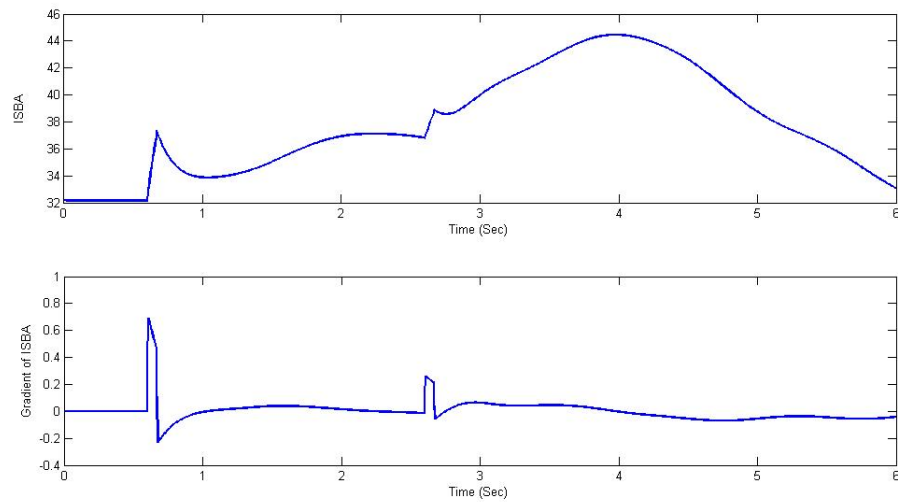


Fig. 3.4. ISBA and Gradient of ISBA

### 3.4 Rate of Change of Bus Angles

Frequency of bus voltages, also known as rate of change of BAs, represents another index in this research. This was calculated by obtaining the point to point difference of bus angles divided by the difference between two consecutive time steps. The bus voltage frequencies along with other measurements and calculated indices were used toward achieving the objectives of this thesis.

## 4. DECISION TREES

### 4.1 Overview

Decision Tree (DT) is a learn by sample pattern recognition tool that provides solutions to classification problems with small number of output classes, such as stable/unstable in this thesis [6], [18]. DTs are built off-line from training data sets where each data point consists of an input vector along with a target value which shows the true classification of that sample [18]. The target value is either 0 or 1 representing the class each data belongs to. Figure 4.1 shows a training data example from 480 trajectories in ISBA-GradISBA plane where a target value is assigned for every time sample. The unstable cases, with maximum generator angle difference larger than 300 degrees, are represented by blue “x” and the stable cases, with maximum generator angle difference smaller than 300 degrees, are represented by red “0”.

Using the training data, the DT follows a statistical method to find classification thresholds. These thresholds, presented in terms of if-then rules, are later tested on on-line unseen data points and the prediction is compared to the true classes [6], [18]. The main advantage of DTs over other pattern recognition tools is its processing time. Another advantage is that with a large number of predictor variables available, a small subset is usually enough for the DT to make the classification and this means that there is much lower chance of facing missing data issue [6].

Figure 4.2 shows a sample DT as a flowchart with rules. Each testing sample point starts at the top node and proceeds down on the tree until it reaches a terminal node. The title of the tree shows the name of the file that contains the training data. The first top node shows that if the GradISBA is greater than or equal 0.12, the left branch is chosen where the data point is predicted to be unstable, 0 in this thesis. The

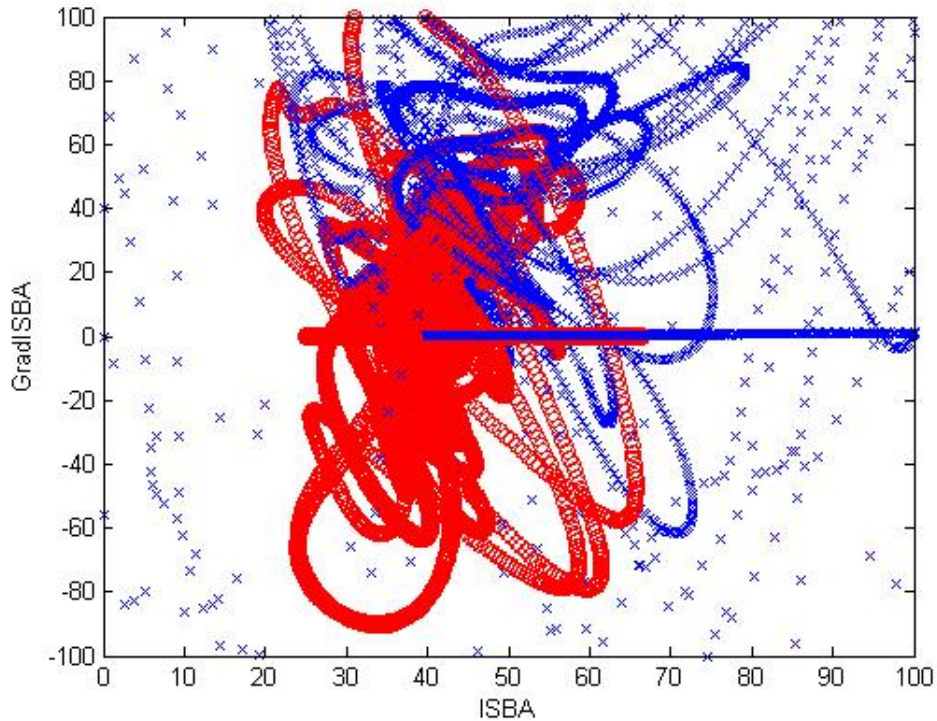


Fig. 4.1. Training data from 480 simulations in ISBA-GradISBA space

top left branch also tells that 69693 training data points fell into the category of that terminal node where 98.7 percent of those points had the correct target value of 0. This percentage represents the precision of the data classification. If the GradISBA of that data point is smaller than 0.12, it chooses the right branch and follows the tree rules until it gets classified at a terminal node.

## 4.2 Rattle Parameters

There are two parameters in the DT tool, Rattle, that need to be adjusted to modify the output: the complexity cost and the loss matrix.

The complexity cost is a number between 0 and 1 with increments of 0.0001 and is used to adjust the size of the tree. Large complexity cost leads to a simple DT with few nodes whereas a small complexity cost leads to a more complex DT with too

**Decision Tree WtrainbeforeControl\_1DTmethod.csv \$ X.istabb.**

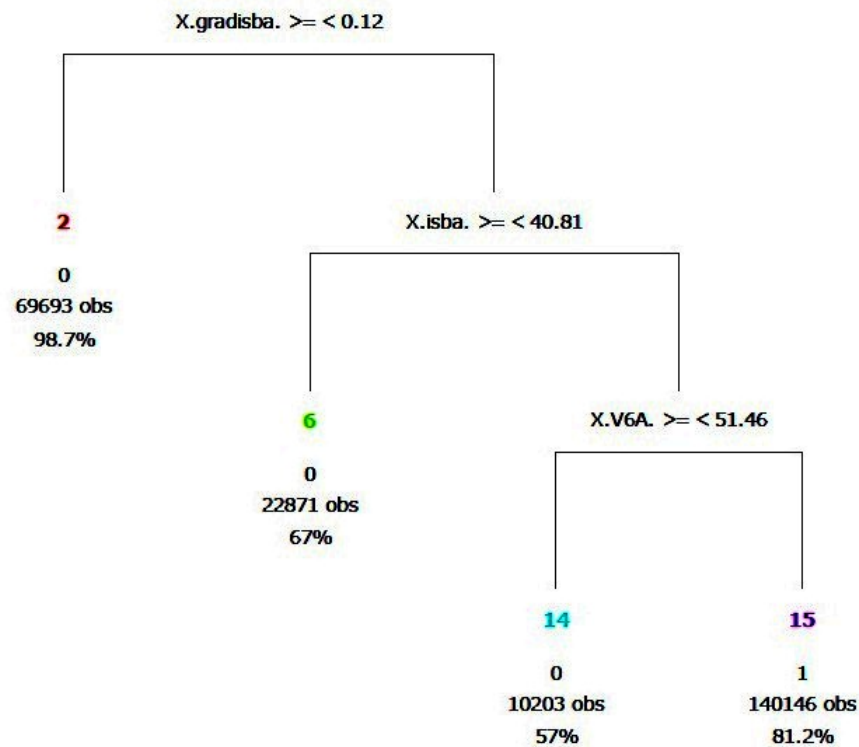


Fig. 4.2. DT Example

many levels and branches. Although selecting the complexity cost too high provides simpler output rules which are less explicit to the training data, it might ignore some important nodes and branches.

The loss matrix is a comparative misclassification cost which is used to make either of the stable or unstable nodes almost pure [1]. Four outcome classes involved in the loss matrix are True Negative (TN), False Negative (FN), False Positive (FP), and True Positive (TP). In this thesis, 1 represents stable and is considered as the positive class. Knowing that, TN is when a 0 is correctly classified as 0, the FN is when a 1 is misclassified as 0. FP is when a 0 is misclassified as 1, and the TP is when a 1 is correctly classified as 1. The loss matrix is entered into Rattle in the

order of TN, FN, FP, TP like 0,20,1,0. This does not assign any weight to the TN and TP, but penalizes the FN 20 times more than the FP.

### 4.3 DT Data

DTs of this thesis were mostly trained with the set of 480 cases. For each of these cases, the TSAT simulation provided GAs and BAs for each time step, 120 per second; then bus frequencies, ISBAs, and the gradient of the ISBAs were calculated. A target value was also assigned to each time sample such that if the case was stable, every sample was assigned a target value of 1; the reverse was done for unstable cases. All of this data was stored in an output spreadsheet and imported into Rattle as the training data. Then, the DT parameters were adjusted. The DT rules were then tested on the set of 385 cases to detect event, predict instability, and select the appropriate control case.

## 5. ONE SHOT CONTROL WITH RESPONSE BASED DECISION TREE

### 5.1 Overview

The response based DT method is used as a part of this thesis to predict loss of synchronism. In this method time and location of the events are not detected and one shot stabilizing control of fast power changes are applied only based on the prediction of instability.

Training for this DT started with 6 seconds of simulation on the 480 cases set where the initial classification was done using the 300 degrees threshold for maximum generator angle difference explained in chapter 2. Then, ISBA and its gradient were calculated and used along with V6A, the BA of the bus INTERMT 345, and V9ADot, the bus frequency of the bus MONTANA 500, as the training predictors for the DT [1]. The target value associated with each time sample was 0 for unstable data points and 1 for stable ones. Implementing the training data into Rattle, five different combinations of complexity cost and loss matrix were selected to create 5 sets of DT rules for instability prediction. These rules were each tested on the 385 cases set where each time sample was compared with the rules and a prediction target value was assigned to it. The first data point, for which a 0 was predicted as the output value, was the point where the instability was predicted on the system.

### 5.2 Application of One shot Control

In TSAT simulations, Add Admittance command ( $\pm 500$  MW) was used as an effective alternative for fast power changes. Add Admittance with positive power change increases the power demand whereas, Add Admittance with negative power

change acts similar to the Shed Load command [6]. [6] Also explains that one shot fast power changes are stronger than ramped power changes and cause more cases to be stabilized.

### 5.2.1 Location and Time of Control

In this thesis, two different control sets are used where each involves fast power changes on four different buses. To create these control cases, five load buses were selected in the northern part of WECC for positive power changes and five load buses were selected in the southern part of WECC for negative power changes. With 10 selected load buses, 100 different control sets were created by selecting two buses out of the five selected northern buses and two buses out of the five selected southern buses. Using a trial and error method, these control sets were tested on 230 unstable cases of the 480 cases set; two controls with the largest number of stabilized cases, out of 230, were selected for the purposes of this thesis.

Other than the selection of buses for one shot controls, the timing of control application has significant importance in preventing the system from losing stability. The control should be applied as soon as the instability is predicted. As a delay exists for information to be transferred over the grid, 6 cycles after the prediction of instability is the earliest time that control can be applied.

Figure 5.1 shows the timing of controls applied by DT to a set of 30 unstable cases, each included a single phase to ground fault followed by a three phase fault. The figure illustrates that for 63 percent of the cases, controls were applied early in the simulation. The same 30 cases were tested with controls applied with different delays. Figure 5.2 states that when the control was applied after the fault clearing time, all 30 cases were stabilized. The number of stabilized cases decreased as the application of control was postponed up to a point where no case was stabilized with control applied at second 3 of the simulation.

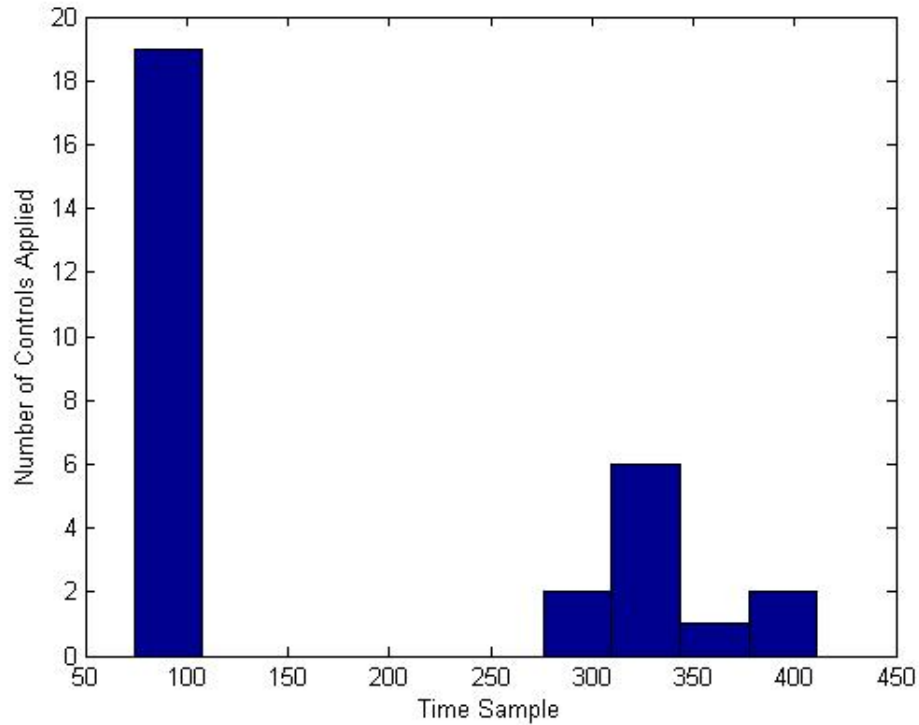


Fig. 5.1. Timing of control application for 30 unstable cases

### 5.3 Results

Tables below represent the results of one shot controls using response based DTs tested on 385 cases set. As explained in section 5.1, the DT tree rules were generated with the training data and 5 different combinations of loss matrix and complexity costs. Table 5.1 shows the result when control case 1 was applied and Table 5.2 represents the results of application of control case 2.

Utilizing indices like ISBA and its gradient as input to the single DT method, show to be effective and demonstrate improvement when comparing results with previous research. But, because the control was applied into too many cases, this method was not found economical as there is a significant cost allocated with each control application.



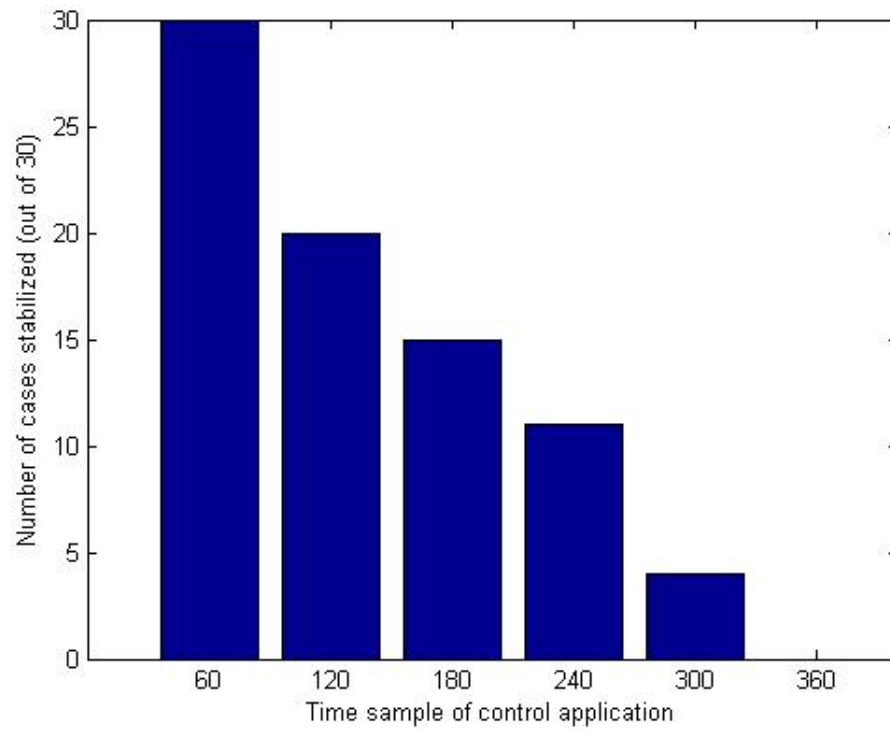


Fig. 5.2. Number of cases stabilized with control applied at different times

Table 5.1

Results of response based DT method with application of control case 1 to 385 cases

Loss Matrix	Complexity Cost	Number of Controls Applied	Number of Cases Stabilized
No Loss Matrix	0.01	271	39
No Loss Matrix	0.04	256	39
0,10,1,0	0.01	195	34
0,30,1,0	0.02	191	31
0,40,1,0	0.01	130	16

Table 5.2  
 Results of response based DT method with application of control case  
 2 to 385 cases

Loss Matrix	Complexity Cost	Number of Controls Applied	Number of Cases Stabilized
No Loss Matrix	0.01	271	32
No Loss Matrix	0.04	256	28
0,10,1,0	0.01	195	24
0,30,1,0	0.02	191	20
0,40,1,0	0.01	130	2

## 6. PROPOSED METHOD USING THREE DECISION TREES

### 6.1 Overview

Predicting instability without knowing the time and location of event is not the best approach for preventing the power system from loss of synchronism. Therefore, an event detection method using decision trees was developed and combined with an instability prediction method similar to the one discussed in the previous chapter. In this combination, event detection was done for all the cases in a set, but instability prediction was done only for the cases which an event was detected. A third set of decision tree rules was then developed to select one of the control cases of chapter 5. When an event was detected and instability was predicted for a case, the selected one shot control was then applied.

### 6.2 DT1: Event Detection

In this section a set of decision tree rules were developed to detect the time of the events in the system prior to any check for instability. To train this decision tree, a set of 284 cases were created. Each case included a single event of fast power change of either 200 MW, 500 MW, 200 MVAR, or 500 MVAR at the 40th cycle at each of the 71 load buses in the power system model. The purpose of creating these cases was to find an event detection threshold that would detect most of the 500 MW/MVAR power changes and not too many of the 200 MW/MVAR ones. If a threshold was found that would detect all the 500 MW/MVAR events, it would be capable of detecting all the larger events that might lead into instability. Therefore, simulations with length of 6 seconds were done on each of these cases and the variables

V6A, V9ADot, ISBA, and gradient of ISBA were recorded for training. As the 200 MW/MVAR power changes would not affect the stability of the system, it was treated as a no event. On the other hand, the 500 MW/MVAR power changes would cause some changes in the system, but might not make the system unstable, so these cases were treated as events. To develop the event detection rules, the training data was imported into Rattle with a loss matrix of 0,20,1,0 and a complexity cost of 0.02; the resulted rules are shown in Figure 6.1. Because the training of the instability

Decision Tree Wtrain\_admittance\_284-eventdetection.csv \$ X.istabb.

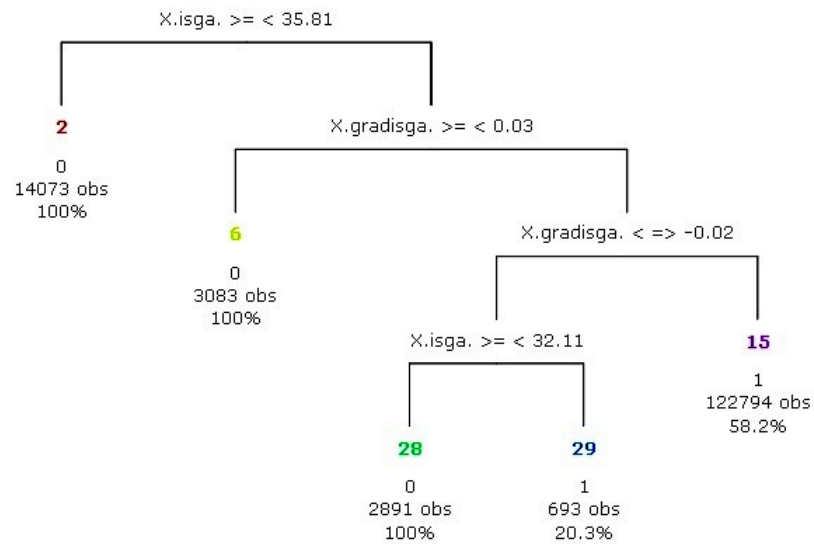


Fig. 6.1. Rules for event detection DT

prediction and control selection decision trees were planned to be done with the set of 480 cases, the event detection rules were tested on the 480 cases set. The variables of each time sample were compared with the event detection rules and were classified as event or no event. The first time that a data sample was classified as event was the time that the start of event was detected for that case. The resulted thresholds detected 441 events which included events of all 230 unstable cases of the 480 cases set. This was an on target result for training of the other two decision trees.

### 6.3 DT2: Instability Prediction

As mentioned in the previous section, for training the instability prediction decision tree, simulations were done on the set of 480 cases. The procedure consisted of an initial classification for every case using the method of chapter 2 where cases with maximum generator angle difference of larger than 300 degrees were classified as unstable and vice-versa. Then, for each case, variables and calculated indices of every time sample were compared with the event detection rules. When an event was detected for the case, a delay of 4 time cycles (8 time sample points) were allowed for the event to be over and then training data was collected for 5 sample points. The 4 cycles delay was selected on a trial and error basis with respect to the time events were detected by the first DT and the length of the events in the cases. With this delay, for most of the cases, the collected 5 sample points fell after the 40th cycle of the simulation which was the end of the first event in all simulated cases. The target value used with this collected post event data was according the initial classification done in the simulations, target value of 1 for stable cases and target value of 0 for unstable cases.

Instability prediction rules were then developed by importing the training information into Rattle. Details about the complexity cost and loss matrix associated with DT rules are presented in Table 6.1.

Testing was performed on the 385 cases set such that, for cases with a detected event, 4 time cycles of wait time were allowed for the event to be over similar to the training process. After the delay, 5 sample points of test data were each compared with the instability prediction rules and were classified into two groups of predicted stable and predicted unstable. Similar to what explained previously, the first time that a data sample was classified as predicted unstable was the time that the instability was predicted for that case. A one shot control was applied 6 cycles after the prediction of instability; this was the earliest time a control could be applied because there was a delay of 6 cycles for the information to be transferred over the grid.

Testing on 385 cases set was done once with application of control case1 and once with control case2. The instability prediction DT was the one that decided on the number of controls applied. The approach of this research up to this point, where event detection method was combined with instability prediction for application of one shot control, is referred to as the 2DTs method in the rest of this report. Table 6.1 represents the results of the 2DTs method.

#### **6.4 DT3: Control Selection**

After the 2DTs method was established, the results showed that the cases that were stabilized by each of the two control cases were not all in common between the two of them. For instance, line 1) of table 6-1 shows that control case 1 stabilized 27 cases and control case 2 stabilized 19 cases, but details of the results showed that not all of the 19 cases were included in the 27 cases, meaning that the overall number of cases stabilized with these two controls was more than 27 cases. This highlighted a potential chance that if, instead of applying only one of the control cases to every case, a selection of these two controls was used, the number of stabilized cases could improve. Therefore, a third set of decision tree rules were developed so that every case with a detected event could choose between control 1 and control 2 to apply it when instability was predicted.

Training of this DT, with the 480 cases set, was very similar to training of the instability prediction DT except for the target values. In terms of training data points, after detection of event, 4 cycles were allowed for the event to be over and then, 5 sample points were collected. In terms of the target value, the purpose was to specifically project what cases were stabilized with each of the controls. Therefore, training of the target value was done for each control case independently. For control case 1, the target value was assigned 1 only when a case was unstable without application of any control and was stabilized with application of control 1; otherwise the target value was 0. Similarly, for control case 2, the target value was assigned 1 only when

Table 6.1  
Results of the 2DTs method tested on 385 cases

Combination	Loss Matrix and Complexity Cost	Number of Controls Applied	Number of Cases Stabilized when Case 1 was Applied	Number of Cases Stabilized when Control Case 2 was Applied
1)	No Loss Matrix, Complexity 0.1	135	27	19
2)	No Loss Matrix, Complexity 0.04	125	27	18
3)	No Loss Matrix, Complexity 0.02	119	27	19
4)	Loss Matrix 0,5,1,0, Complexity 0.04	84	22	15
5)	Loss Matrix 0,10,1,0, Complexity 0.08	67	20	15

a case was unstable without application of any control and was stabilized with application of control 2. Two training sets of data were created, one for each control case, and their rules were developed independently with Rattle. The rules for control case 1 along with the rules for control case 2 were all used together as the third DT for control selection. Table 6.2 shows different combinations of complexity cost and loss matrix used in development of the rules. The control selection DT was tested

Table 6.2  
Different combinations of complexity cost and loss matrix used in development of the control selection rules

Combination	Loss Matrix	Complexity
a)	0,5,1,0	0.04
b)	0,10,1,0	0.08
c)	0,20,1,0	0.1

on the 385 cases set in a chronological order with the event detection and stability prediction DTs. During the test, after the detection of an event and a wait time of 8 sample points, 5 sample points of data were compared with the DT rules of control case 1. For each sample point, according to the rules, an output value of zero or one was assigned, and then the number of ones was counted out of five. The same process was done for control case 2 by comparing the five sample points of data with its rules. The control case that had 3 or more ones out of five was selected to be applied to the case, if both had at least 3 ones out of five, the control case with maximum number of ones was selected. Simultaneous with checking the 5 sample points with control selection rules, the same points were compared with the instability prediction rules and in case instability was predicted for that simulation, the selected control was applied 6 cycles after the prediction of instability.



With the 3DTs method, the overall approach for every case can be summarized as shown in Figure 1.3. Table 6.3 represents detailed results of the 3DTs method.

### 6.5 Comparison Between 1DT, 2DT, and 3DT Methods

As explained previously in this report, the goal of this thesis was to find a method that would reduce the number of controls applied and increase the number of cases stabilized. Figure 6.2 compares the test results of 1DT, 2DTs, and 3DTs methods on 385 cases set. Comparing the 1DT method, only instability prediction, with the 2DTs

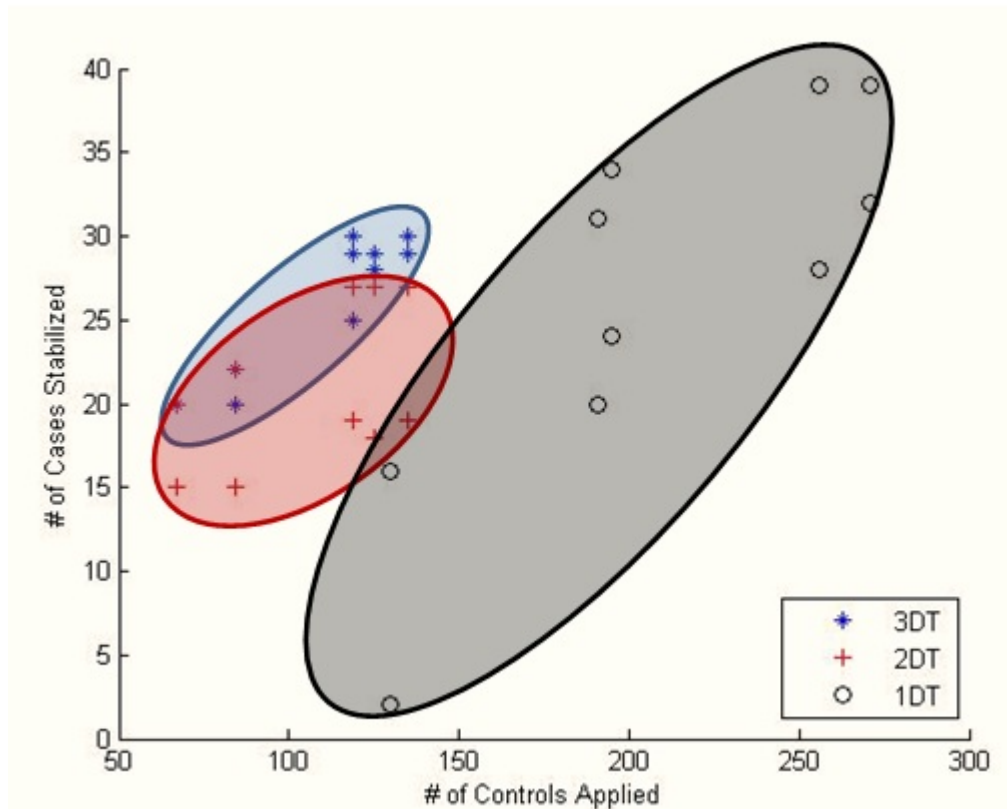


Fig. 6.2. Comparison of the results of the 1DT, 2DTs, and 3DTs method tested with 385 cases

method, event detection plus instability prediction, the improvements are significant. In the 1DT method, the control was very trigger happy while only, in average, 13% of

Table 6.3  
Results of the 3DTs method tested on 385 cases

Combination	Number of Controls Applied	Number of Times Control Case 1 was Applied	Number of Times Control Case 2 was Applied	Number of Cases Stabilized
Rules 1) and a)	135	77	58	29
Rules 1) and b)	135	102	33	30
Rules 1) and c)	135	102	33	30
Rules 2) and a)	125	77	48	28
Rules 2) and b)	125	101	24	29
Rules 2) and c)	125	101	24	29
Rules 3) and a)	119	77	42	29
Rules 3) and b)	119	87	32	25
Rules 3) and c)	119	102	17	30
Rules 4) and a)	84	41	43	20
Rules 4) and b)	84	81	3	22
Rules 4) and c)	84	81	3	22
Rules 5) and a)	67	67	0	20
Rules 5) and b)	67	67	0	20
Rules 5) and c)	67	67	0	20

the controls applied led to stabilized cases. On the other hand, the 2DTs method increased the average ratio of number of stabilized cases to number of controls applied by 7%. This meant that not only lower number of controls was applied, but even more cases were stabilized in relation with the number of controls applied.

Comparison of the 2DTs method with the 3DTs method, in which one of the two control cases was selected, showed even more improvement. Figure 6.2 shows that the number of controls applied in these two methods was the same; this is because the instability prediction DT is the one that decides on control application and this DT stayed the same between these two methods. But, the figure also shows that number of stabilized cases was increased with the 3DT method. The overall comparison shows that in the 1 DT method, 13% of the controls applied ended up stabilizing the cases, where this number was increased to 20% with the 2DTs method while the 3DTs method improved this number to 24%.

In this thesis, as mentioned before, calculated indices of ISBA and GradISBA were involved in event detection, instability prediction, and control selection processes where as in [13], a Curve Fitting Error (CFE) index was calculated and involved in its 2DT method. Figure 6.3 compares the results of this thesis, tested on 385 cases set, with an approximation of the results of the 2DT method developed in [13] and tested on 480 cases set. The figure shows that in [13], in average, only 9% of the number of controls applied resulted in stabilized cases while the 2DT method of this thesis showed that 20% of its applied controls helped stabilizing cases. Another difference between the results of this thesis and results of [13] is that the control case used in [13] included fast power changes on four DC buses of the model while in this thesis, new control cases were developed which resulted in improved results.

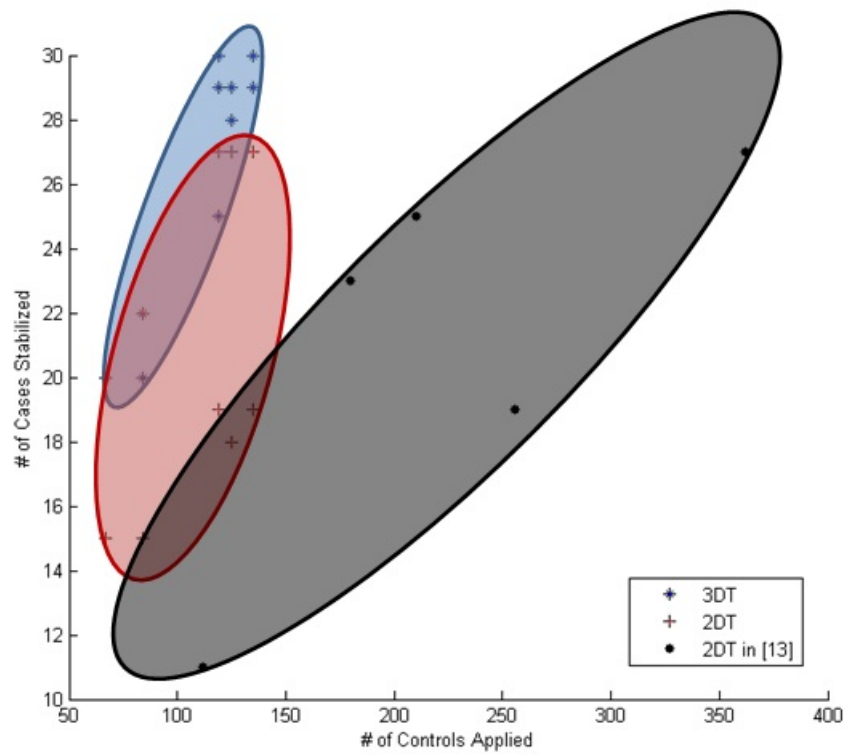


Fig. 6.3. Comparison between the results of this thesis, tested on 385 cases, with the approximated 2DT results in [13], tested on 480 cases

## 7. ROLE OF CONTROL APPLICATION IN STEADY-STATE ANALYSIS

During the internship at MISO the importance of the after-the fact steady-state analysis was studied [21]. In steady-state condition, it is assumed that all the events and disturbances are over and the system has settled [22]. The purpose of the steady-state analysis is to study the impact of the events and the control application on transmission system facilities under steady-state conditions. This analysis is primarily done with powerflow analysis.

Therefore, for the power system model of this research, steady-state powerflows were compared with and without control. For the comparison to be realistic, simulations with length of 12 seconds were done to allow the system to settle. These simulations were mainly done on the cases that experienced an event but did not cause the system to lose its stability. In these cases, the system did not lose its synchronism either without or with control.

With comparison of the steady-state conditions of these cases, the effect of the control application was distinguished. When an event happened on the system, even when it did not cause the system to lose stability, the system settled at a higher steady-state powerflow without application of control. Whereas with control application, the same case experienced lower steady-state powerflow. This effect is shown in Figures 7.1 to 7.3 where the average “with control steady-state powerflow” and average “without control steady-state powerflow” on three different transmission lines were recorded for stable cases of the 385 cases set. The figures show the average powerflow between seconds 10 and 12 of the simulation. The figures represent the fact that for each case, the post event steady-state powerflow is lowered with application of control.

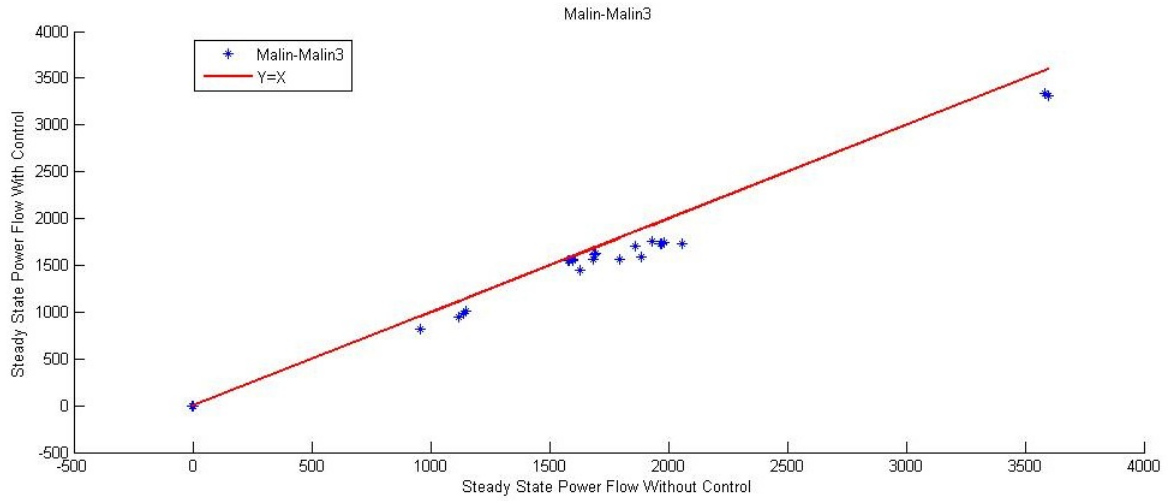


Fig. 7.1. Post event equilibrium in stable cases of the 385 cases set

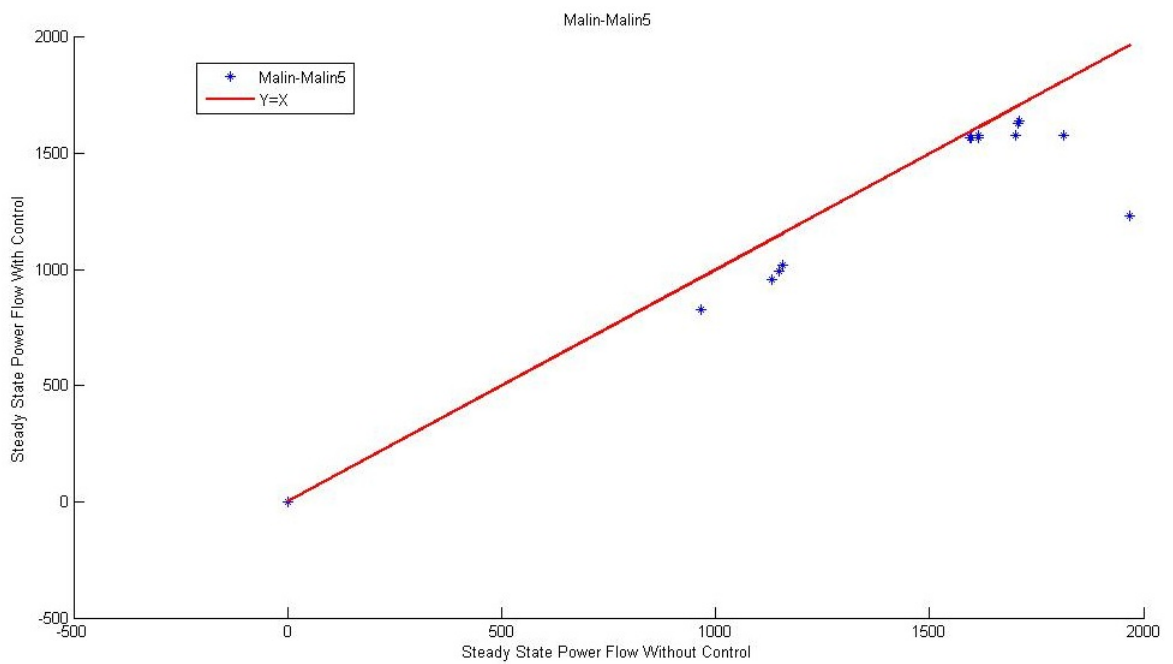


Fig. 7.2. Post event equilibrium in stable cases of the 385 cases set

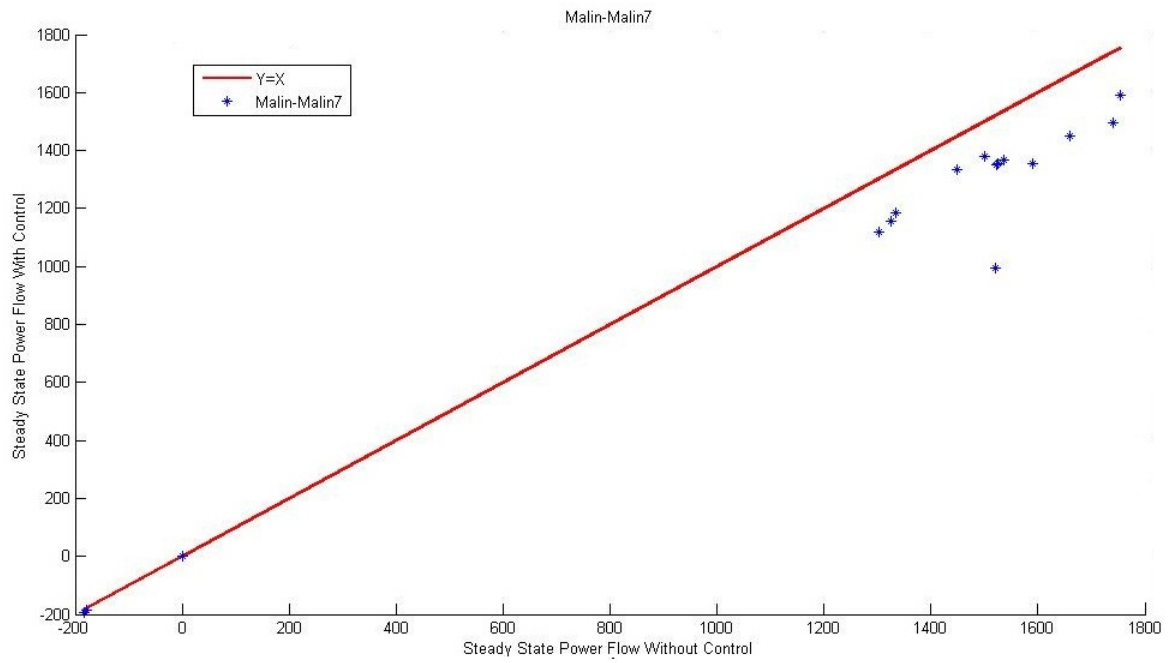


Fig. 7.3. Post event equilibrium in stable cases of the 385 cases set

## 8. CONCLUSIONS AND FUTURE WORK

### 8.1 Conclusions

Maintaining the power system reliability is dependent on the accurate event detection and instability prediction along with application of best possible control. In this research, the ISBA index was presented as a measurement for overall stress of the system. This index and indices similar to this were found to be more reliable than raw system measurements. Also, an event detection method was developed that detects most of the significant events on the system and does not detect small changes. This method was combined with the stability prediction method in a way that when an event was detected, the stability check DT started checking data points to predict any upcoming instability. A third DT was also developed for selecting a control case between two available control cases. It was shown that the combination of event detection with instability prediction and control selection, the 3DT method, performed better in maintaining the reliability of the power system compared with the 1DT method which only involved instability prediction with no event detection or control selection.

### 8.2 Future Work

Finding a method for detecting the time of an event was a real improvement to this research. One step further is to find a way to detect not only the time of an event, but also its location of occurrence. Knowing the location of the event can be helpful in selection of the control case. Multiple control cases can be developed for different geographical locations so that in case of event, the control case closest to the event location could be selected as opposed to selecting general control cases regardless of



where the event has happened. This idea also includes the fact that creating more control cases of fast power changes, line power flow changes, and tripping generators may also help improving the results and accuracy of the method. Another recommendation is to try variables and indices other than the ones used in this thesis and see how they might help improving the accuracy of event detection, instability prediction and control selection. Detecting the end of an event can be helpful in optimizing the timing of the control application, whereas in this research an average of wait time is allowed for the event to be over.

## LIST OF REFERENCES

## LIST OF REFERENCES

- [1] Q. Gao and S. M. Rovnyak, "Decision trees using synchronized phasor measurements for wide-area response-based control," *IEEE Transactions on Power Systems*, vol. 26, pp. 855-861, May 2011.
- [2] S. M. Rovnyak, G. Li, and K. Mei, "One-shot controls for preventing loss of synchronism," in *Proceedings of the IEEE PES General Meeting*, vol. 4, July 2003.
- [3] S. M. Rovnyak and K. Mei, "Dynamic event detection and location using wide area phasor measurements," *European Transactions on Electrical Power*, vol. 21, pp. 1589-1599, May 2011.
- [4] G. Li and S. M. Rovnyak, "Integral square generator angle index for stability ranking and control," *IEEE Transactions on Power Systems*, vol. 20, pp. 926-934, May 2005.
- [5] Y. Sheng and S. M. Rovnyak, "Decision trees and wavelet analysis for power transformer protection," *IEEE Transactions on Power Delivery*, vol.17, pp. 429-433, April 2002.
- [6] S. M. Rovnyak, C. W. Taylor, J. R. Mechenbier, and J. S. Thorp, "Plans to demonstrate decision control tree using phasor measurements for HVDC fast power chages," *Faults and Disturbance Analysis and Precise Measurements in Power Systems Conference*, November 1995.
- [7] K. Mei, S. M. Rovnyak, "Response-Based Decision Trees to Trigger One-Shot Stabilizing Control," *IEEE Transactions on Power Systems*, vol.19, pp. 531-537, February 2004.
- [8] V. Brandwajn, A. B. R. Kumar, A. Ipakchi, and A. Bose, and S. D. Kuo, "Severity indices for contingency screening in dynamic security assessment," *IEEE Transactions on Power Systems*, vol. 12, pp. 1136-1142, August 1997.
- [9] C. Fu and A. Bose "Contingency ranking based on severity indices in dynamic security analysis," *IEEE Transactions on Power Systems*, vol. 14, pp.980-986, August 1999.
- [10] Togaware, "Rattle a graphical user interface for data mining using R," [Online]. Available: <http://rattle.togaware.com/> Last modified August 2012.
- [11] G. J. Williams, "Rattle a data mining GUI for R," *The R Journal*, vol. 1, no. 2, pp. 45-55, 2009.
- [12] PowerTech Labs Inc., "TSAT: Transient security assessment tool," June 2010. [Online]. Available: <http://www.dsatools.com/downloads/TSAT.pdf>. Last accessed September 2012.

- [13] D. W. Longbottom, "Polynomial curve fitting indices for dynamic event detection in wide-area measurement systems," *Master's Thesis, IUPUI*, May 2012.
- [14] Mathworks Inc. , "Matlab the language of technical computing," [Online]. Available: <http://www.mathworks.com>. Last modified December 2011.
- [15] S. M. Rovnyak, M. N. Nilchi, D. W. Longbottom, and D. C. Vasquez, "Angle stability predictive indices," *Proceedings of the IEEE PES General Meeting*, July 2012.
- [16] V. Vittal, "Consequence and impact of electric utility industry on transient stability and small signal stability analysis," *Proceedings of the IEEE*, Vol. 88, No. 2, pp. 196-207, February 2000.
- [17] S. Kretsinger, S. M. Rovnyak, D. Brown, and J. Thorp, "Parallel decision trees for prediction groups of unstable generators from synchronized phasor measurements," *emphPrecise Measurements in Power System Conference*, October 1993.
- [18] S. M. Rovnyak, S. Kretsinger, J. Thorp , and D. Brown, "Decision trees for real-time transient stability prediction," *emphIEEE Transactions on Power Systems*, vol. 9, pp. 1417-1426, August 1994.
- [19] Q. Gao , "Decision trees based one-shot-control for power system wide area control," *Master's Thesis, IUPUI*, 2009.
- [20] PowerTech Labs Inc., "PSAT: Powerflow and shortcircuit analysis tool," June 2010. [Online]. Available:<http://www.dsatools.com/downloads/PSAT.pdf>. Last modified June 2010.
- [21] MISO Energy, " MISO Using Smart Grid Technology to Improve Reliability," September 2012.[Online]. Available: <https://www.misoenergy.org/Pages/Home.aspx>. Last modified September 2012.
- [22] K. Tomsovic, V. Venkatasubramanian, "Power System Analysis,". Available: <http://web.eecs.utk.edu/~tomsovic/Vitae/Publications.html>. Last Accessed November 2012.

## APPENDIX

## APPENDIX: MATLAB CODE FOR POWER SYSTEM SIMULATIONS

```

clear all
clc
delete('C:\Document\Course06\TSAT\WSCC29\Test\swifilebeforecontrol.txt.save');
delete('C:\Document\Course06\TSAT\WSCC29\Test\swifileaftercontrol.txt.save');
delete('C:\Document\Course06\TSAT\WSCC29\Test\destabilized.txt')
delete('C:\Document\Course06\TSAT\WSCC29\Test\Wtrainbeforecontrol.csv');
delete('C:\Document\Course06\TSAT\WSCC29\Test\Wtrainaftercontrol.csv');
Fnamea='DESCR';
Fnameb='NOMOR';
% Control Case 1
confile1= '/DISCONNECT GENERATOR
;CANAD G120.0;1 / ';
confile2= 'ADD ADMITTANCE
;CA230 230 500.0 200.0 MVA /';
confile3= 'ADD ADMITTANCE
;MONTANA 500 500.0 200.0 MVA /';
confile4= 'ADD ADMITTANCE
;RINALDI 230 -500.0 -200.0 MVA /' ;
confile5= 'ADD ADMITTANCE
;SYLMARLA 230 -500.0 -200.0 MVA /' ;
% Control Case 2
confile6= 'ADD ADMITTANCE
;CELILO 230 500.0 200.0 MVA /';
confile7= 'ADD ADMITTANCE
;CRAIG 345 500.0 200.0 MVA /';
confile8= 'ADD ADMITTANCE
;ADELANTO 500 -500.0 -200.0 MVA /' ;
confile9= 'ADD ADMITTANCE
;SYLMARLA 230 -500.0 -200.0 MVA /' ;
% Counter for cases after control
itotUnst=0; % counts number of cases which stayed unstable
itotLose=0; % counts number of cases which lost stability
itotStab=0; % counts number of cases which remained stable
itotSave=0; % counts number of cases which were stabilized
addangles=0;

```

```

counter=0;
% initial matrix for required simulation variables/calculated indices for
% training of event detection(284cases) and
% instability prediction rules(480cases)
X1=[0 0 0 0 0];
X2=[0];
% initial matrix for required simulation variables/calculated indices for
%training of control selection rules(480cases)
X3=[0 0 0 0 0];
SS_Nocontrol=[0 0 0];
% initial matrix for steady state powerflow without control
SS_Withcontrol=[0 0 0];
% initial matrix for steady state powerflow with control
% read the file containing cases and finding the number of lines
swifile = textread('swifilelarge385.txt','%s','delimiter','\n','whitespace','');
nline=length(swifile);
nevent=0; % counter for number of cases in the file
%find start and end of each case in swifile
    for iline=1:nline;
        record = swifile(iline);
        record=char(record);
        h=findstr(record,Fnamea);
        if h==1
            flag1=iline;
            nevent=nevent+1;
            swistart(nevent)=iline;
        end
        k=findstr(record,Fnameb);
        if k==1
            flag2=iline;
            swistop(nevent)=iline;
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% RUN THE EVENT WITH NO CONTROL
for ievent=1:nevent
    ievent
    %Write the fault2.swi file
    fid=fopen('fault2.swi','w+');
    for p=swistart(ievent):swistop(ievent)
        swi=cell2mat(swifile(p));
        fprintf(fid,'%s\r\n',swi);
    end
    fclose(fid);

```

```

clear freq
clear deltat
% time delay because of TSAT liscense limitations
t = timer('TimerFcn',@(x,y)disp('Hello World!'),'StartDelay',10);
    start(t);
wait(t);
% calling TSAT to do the simulation
!C:\dsa_powertools_5_net\tsat\bin\tsat_batch.exe
C:\Document\Course06\TSAT\WSCC29\WECC29.tsa
!C:\dsa_powertools_5_net\tsat\bin\sim2txt WECC29.bin -quan=gen_relang -all >
C:\Document\Course06\TSAT\WSCC29\WECC29GA.txt
!C:\dsa_powertools_5_net\tsat\bin\sim2txt WECC29.bin -quan=bus_va -all >
C:\Document\Course06\TSAT\WSCC29\WECC29BA.txt
!C:\dsa_powertools_5_net\tsat\bin\sim2txt WECC29.bin -quan=line_p -all >
C:\Document\Course06\TSAT\WSCC29\WECC29PF.txt
%%%%calculate maximum generator angle difference and write the results file
BaseFile = 'WECC29GA';
% Developed function to read the generator angle data
R = get_data(BaseFile);
% Transform your cell array into an ordinary matrix
A = R{1};
for i = 2 : 30
    A = [A R{i}];
end
% Find the maximum generator angle difference in the simulated case
B=size(A);
C=zeros(B(1,1),1);
    for M=(1:B(1,1))
        mx=max(A(M,:));
        mn=min(A(M,:));
        diff=abs(mx-mn);
        C(M,1)=diff;
        M=M+1;
    end
maxdiff=max(C);
% the case is stable if the maximum generator angle difference is
% smaller than 300 deg.
if maxdiff<300
    flag=1;
    istab(ievent)=1;
end
% the case is unstable if the maximum generator angle difference is
% larger than 300 deg.
if maxdiff>300

```



```

    flag=0 ;
    istab(ievent)=0;
end
%%calculate successive bus angle differences and freq
%%(set velocity for first sample to be zero)
BaseFile = 'WECC29BA';
%Use the developed function to read bus voltage angle data
R = get_data2(BaseFile);
%Transform the cell array into an ordinary matrix
D = R{1};
for i = 2 : 18
    D = [D R{i}];
end
E=size(D);
nrec=E(1,1);
bus_angles=D;
clear BaseFile R
BaseFile = 'WECC29PF';
%Use the developed function to read line powerflow data (% this was
%used for 12 secs long simulations to find the steady state powerflow)
R = get_data3(BaseFile);
%Transform the cell array into an ordinary matrix
PF = R{1};
for i = 2 : 4
    PF = [PF R{i}];
end
% To unwrap the BAs
% If current - previous ~ 360 then addangles = addangles - 360
% newbus_angles(irec,ind1) equals sum
% bus_angles(irec,ind1) + addangles(irec,ind1)
for ind1=(2:18)%col
    newbus_angles(1,ind1)=bus_angles(1,ind1);
    addangles(1,ind1)=0;
    for irec=(2:nrec)%row
        diff=bus_angles(irec,ind1)-bus_angles(irec-1,ind1);
        addangles(irec,ind1)=addangles(irec-1,ind1);
        if diff<=-330
            addangles(irec,ind1)=addangles(irec-1,ind1)+360;
        end
        if diff>=330
            addangles(irec,ind1)=addangles(irec-1,ind1)-360;
        end
        diff=bus_angles(irec,ind1)+addangles(irec,ind1);
        newbus_angles(irec,ind1)=diff;
    end
end

```

```

end

end

% Make one bus be the reference angle (17th bus was selected)
% (the first column of data was time and the BA data started from column2)
for irec=(1:nrec)%rows
    coa(irec)=0;
    for ind1=(2:18)%columns
        diff=newbus_angles(irec,ind1)-newbus_angles(irec,18);
        newbus_angles(irec,ind1)=diff;
        coa(irec)=coa(irec)+newbus_angles(irec,ind1)/17;
        % calculating the center of angle
    end
end

end

% calculate the sga which is really sba
% calculate the isga which is really isba
sga(1)=32;
isga(1)=sga(1);
for irec=(2:nrec)%rows
    sga(irec)=0;
    for ind1=(2:18)%columns
        diff=newbus_angles(irec,ind1)-coa(irec);
        sga(irec)=sga(irec)+diff*diff/17;
    end
    sga(irec)=sqrt(sga(irec));
    sga(1)=sga(2);
    isga(1)=sga(1);
    isga(irec)=isga(irec-1)+(1/120)*6*(sga(irec-1)-isga(irec-1));
end

gradisga=gradient(isga);

% to calculate frequency (set velocity for first sample to be zero)
t=bus_angles(:,1);
newbus_angles(:,1)=t;
freq=zeros(E(1,1),E(1,2));
for irec=(2:nrec)
    for ind1=(2:18)
        % finding the BA difference b/w two consecutive points
        diff=newbus_angles(irec,ind1)-newbus_angles(irec-1,ind1);
        % to avoid division by zero
        deltat1=(t(irec)-t(irec-1));
        deltat2=(0.1*(t(9)-t(1)));
        deltat=max(deltat1,deltat2);
        f=diff/deltat;
        freq(irec,ind1)=f;
    end
end

```

```

        end
    end
    freq(:,1)=t;
    %(set velocity for first sample to be zero)
    freq(1,:)=zeros(1,18);
    Fname5='.csv';
    [r c]=find(freq(:,2:end)>=1);
    evdet=min(r);
    %V6A and V9Adot, bus voltage angle of the 6th bus and the bus frequency
    %of the 9th bus (the first column of data is time)
    V6A=newbus_angles(:,7);
    V9Adot=freq(:,10);
    y=size(V6A);
    if flag==0;
        istabb=zeros(y(1,1),1); %assigning target value to each sample point.
    end
    if flag==1;
        istabb=ones(y(1,1),1); %assigning target value to each sample point.
    end
    iEVE(ievent)=0;
% matrix that stores the start of the event for each case
    icontrol(ievent)=0;
% matrix that stores the sample point of instability prediction for each case
    iCNTRL(ievent)=0;
% matrix that stores the sample point that one of the control cases were selected
%for each case
    cntrl(ievent)=0;
% matrix that stores which control case was selected for each case
    maxxx(ievent)=0;
% this matrix is later used in the selection of control case
% (stores the number of ones over five for the control case with larger number,
% for each case)
% RULES FOR EVENT DETECTION
    ipredEVE=zeros(y(1,1),1);
    % comparing the variables and indices with the rules and classifying as
    % "event" or "no event"
    for irec=(1:nrec)
        if gradisga(irec)>=-0.01899 && isga(irec)<35.81
            if gradisga(irec)<0.02631
                ipredEVE(irec)=1;
            end
        end
    end
    if gradisga(irec)<0.02631 && gradisga(irec)<-0.01899
        if isga(irec)<32.11 && isga(irec)<35.81

```

```

        ipredEVE(irec)=1;
    end
end

    if (ipredEVE(irec)==0 && iEVE(ievent)==0)
        iEVE(ievent)=irec;
    end
end

% X1 is the matrix that stores the without control variables and indices
% This matrix is later saved to a file as the training data for
% instability prediction
% Data for each new simulation is appended to this matrix
X1=[X1;istabb(iEVE(ievent)+16:iEVE(ievent)+20) V6A(iEVE(ievent)+16:iEVE(ievent)+20)
V9Adot(iEVE(ievent)+16:iEVE(ievent)+20) isga(iEVE(ievent)+16:iEVE(ievent)+20)'
gradisga(iEVE(ievent)+16:iEVE(ievent)+20)'];
% for training of the event detection DT the X1 used is as the next line
% X1=[X1;istabb V6A V9Adot isga' gradisga' ];
% creating a file to save X1 at the end of the code
Fname9='C:\Document\Course06\TSAT\WSCC29\Test\Wtrainbeforecontrol';
Fname10=[Fname9 Fname5];
%   %% RULES FOR INSTABILITY PREDICTION
ipred=zeros(y(1,1),1);
% Different instability prediction rule sets (1 to 5)
%created with various combinations of loss matrix and complexity cost
% One rule set was used at a time
% The check for instability was done only if an event was detected
if iEVE(ievent)>0
    % there is a wait time of 4 cylces, 8 sample points, after the event
    % detection. Then 5 sample points of data are compared with the
    % rules
    for irec=(iEVE(ievent)+16:iEVE(ievent)+20)
        %1)
        if V9Adot(irec)<3.694 && gradisga(irec)<0.08214
            if V6A(irec)>=58.44
                ipred(irec)=1;
            end
        end
    end
    if gradisga(irec)<0.08214 && V6A(irec)<38.49
        if V9Adot(irec)<25.48
            ipred(irec)=1;
        end
    end
end
%   %2)
%   if gradisga(irec)< 0.08333 && V6A(irec)>=38.49

```

```

%         if V9Adot(irec)<3.694 && V9Adot(irec)>=-14.05
%             ipred(irec)=1;
%         end
%     end
%     if gradisga(irec)<0.08333 && V9Adot(irec)<25.48
%         if V6A(irec)<38.49
%             ipred(irec)=1;
%         end
%     end
%     if gradisga(irec)<0.08333 && V9Adot(irec)<25.48
%         if V6A(irec)<38.49 && isga(irec)>=33.66
%             if V9Adot(irec)>=3.694
%                 ipred(irec)=1;
%             end
%         end
%     end
%     if V9Adot(irec)>=25.48 && gradisga(irec)<0.0261
%         ipred(irec)=1;
%     end
%3
%     if gradisga(irec)<0.08333
%         ipred(irec)=1;
%     end
%     if gradisga(irec)>=0.08333 && V9Adot(irec)<54.97
%         if V6A(irec)<38.59
%             ipred(irec)=1;
%         end
%     end
%4
%     if gradisga(irec)<0.08214 && V9Adot(irec)<25.48
%         if V6A(irec)<38.49
%             ipred(irec)=1;
%         end
%     end
%     if V9Adot(irec)<25.48 && V6A(irec)>=58.44
%         if gradisga(irec)<0.08214
%             ipred(irec)=1;
%         end
%     end
%5
%     if V9Adot(irec)<25.48 && gradisga(irec)<0.08214
%         ipred(irec)=1;
%     end
if (ipred(irec)==0 && icontrol(ievent)==0)

```

```

        icontrol(ievent)=irec;
    end
end
end
% if an instability is predicted for a stable case its steady state
% powerflow is recorded. (b/w 10th and 12th secs)
% This data will then be compared with the steady state powerflow with control
    if icontrol(ievent)>0 && istab(ievent)==1
        r1=1200;
        r2=find(PF(:,1)==max(PF(:,1)));
        SS_Nocontrol=[SS_Nocontrol;mean(PF((r1:r2),2)),mean(PF((r1:r2),3)),
            mean(PF((r1:r2),4))];
    end
% RULES FOR CONTROL SELECTION
% 3 different rule sets were created for various loss matrix and
% complexity cost. Rule sets a, b, and c. One rule set at a time
    if icontrol(ievent)>0
        % This is done only for cases with event detected and instability
        % predicted here to save the time of simulation, but in real time,
        % for every case with detected event a control should be selected so
        % that incase the instability is predicted, we know which control to
        % apply without any further delay
        ipredCNTRL6=zeros(y(1,1),1);
        ipredCNTRL98=zeros(y(1,1),1);
        i6=zeros(1,nrec);
        i98=zeros(1,nrec);
        % after the event detection and 8 sample points of wait time, 5
        % sample points were compared with the control selection rules
        for irec=(iEVE(ievent)+16:iEVE(ievent)+20)
            i6_a=0;
            i98_a=0;
%            a)
            if V9Adot(irec)>=16.15 && V6A(irec)>=32.31
                if V9Adot(irec)<85.66
                    ipredCNTRL6(irec)=1;
                    % each sample point is assigned a 0 or one
                    i6_a(1)=(sum(ipredCNTRL6((irec-5):irec)))/5;
                    % ratio of ones over 5 is calculated
                end
            end
            i6(irec)=max(i6_a); % maximum ratio for control case 1 is found
            if isga(irec)<32.45 && V9Adot(irec)<49.08
                if gradisga(irec)>=-0.0006506 && gradisga(irec)<0.04761
                    ipredCNTRL98(irec)=1;
                end
            end
        end
    end
end

```

```

        i98_a(1)=(sum(ipredCNTRL98((irec-5):irec)))/5;
    end
end
if gradisga(irec)>=-0.0006506 && gradisga(irec)<0.04761
    if V9Adot(irec)<-10.28 && isga(irec)>=32.45
        ipredCNTRL98(irec)=1;
        i98_a(2)=(sum(ipredCNTRL98((irec-5):irec)))/5;
    end
end
if gradisga(irec)<0.04761 && V9Adot(irec)<49.08
    ipredCNTRL98(irec)=1;
    i98_a(3)=(sum(ipredCNTRL98((irec-5):irec)))/5;
end
i98(irec)=max(i98_a);% maximum ratio for control case 2 is found
%b
%    if V9Adot(irec)>=16.15
%        ipredCNTRL6(irec)=1;
%        i6_a(1)=(sum(ipredCNTRL6((irec-5):irec)))/5;
%    end
%    i6(irec)=max(i6_a);
%    if V9Adot(irec)<49.08 && gradisga(irec)>=-0.0006506
%        ipredCNTRL98(irec)=1;
%        i98_a(1)=(sum(ipredCNTRL98((irec-5):irec)))/5;
%    end
%    i98(irec)=max(i98_a);
%c
%    if V9Adot(irec)>=16.15
%        ipredCNTRL6(irec)=1;
%        i6_a(1)=(sum(ipredCNTRL6((irec-5):irec)))/5;
%    end
%    if V9Adot(irec)<16.15 && gradisga(irec)>=0.04508
%        ipredCNTRL6(irec)=1;
%        i6_a(2)=(sum(ipredCNTRL6((irec-5):irec)))/5;
%    end
%    i6(irec)=max(i6_a);
%    if V9Adot(irec)<49.08 && gradisga(irec)>=-0.0006506
%        ipredCNTRL98(irec)=1;
%        i98_a(1)=(sum(ipredCNTRL98((irec-5):irec)))/5;
%    end
%    i98(irec)=max(i98_a);
[maxmag cntrlcase]=max([i6(irec) i98(irec)]);
% maximum ratio b/w two control cases is found
% the point that the maximum calculated ratio is larger than 0.6
% meaning that at least 3 out of five points were assigned

```

```

% one, is the point that one control case is selected based on
if (maxmag>=0.6 && iCNTRL(ievent)==0)
    iCNTRL(ievent)=irec;
    cntrl(ievent)=cntrlcase;
    maxxx(ievent)=maxmag;
end
clear i6_a i98_a
end
% if for a case with detected instability, all five sample points
% are compared and no control was selected, then control case 1 is
% automatically selected
if iCNTRL(ievent)==0
    counter=counter+1;
    cntrl(ievent)=1;
    iCNTRL(ievent)=icontrol(ievent);
end

end
delete('WECC29BA.txt')
delete('WECC29GA.txt')
delete('WECC29PF.txt')
clear istabb V6A V9Adot isga gradisga varV6A varV9Adot
clear bus_angles newbus_angles freq t D PF i6 i98
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% RUN THE EVENT WITH CONTROL
%write the fault2.swi with control
fid=fopen('fault2.swi','w+');
for p=swistart(ievent):swistop(ievent)-1
    swi1=cell2mat(swifile(p));
    fprintf(fid,'%s\r\n',swi1);
end
% if instability predicted, control lines added to swifile
if icontrol(ievent)>0
    ncy=6+icontrol(ievent)/2;
    % 6 cycles is the time needed for the information to transfer
    fprintf(fid,'AT TIME %4.0f CYCLES / \r\n',ncy);
    fprintf(fid,'AT TIME 120 CYCLES / \r\n');
    if cntrl(ievent)==1
        fprintf(fid,'%s\r\n',confile1);
        fprintf(fid,'%s\r\n',confile2);
        fprintf(fid,'%s\r\n',confile3);
        fprintf(fid,'%s\r\n',confile4);
        fprintf(fid,'%s\r\n',confile5);
    end
end

```



```

        if cntrl(ievent)==2
            fprintf(fid,'%s\r\n',confile6);
            fprintf(fid,'%s\r\n',confile7);
            fprintf(fid,'%s\r\n',confile8);
            fprintf(fid,'%s\r\n',confile9);
        end
    end

    q=swistop(ievent);
    swi2=cell2mat(swifile(q));
    fprintf(fid,'%s\r\n',swi2);
    fclose(fid);

    clear freq
    clear deltat

    %time delay because of TSAT liscense limitations
    t = timer('TimerFcn',@(x,y)disp('Hello World!'),'StartDelay',15);
    start(t);
    wait(t);

    % Calling TSAT to do the simulation
!C:\dsa_powertools_5_net\tsat\bin\tsat_batch.exe
C:\Document\Course06\TSAT\WSCC29\WECC29.tsa
!C:\dsa_powertools_5_net\tsat\bin\sim2txt WECC29.bin -quan=gen_relang -all >
C:\Document\Course06\TSAT\WSCC29\WECC29GA.txt
!C:\dsa_powertools_5_net\tsat\bin\sim2txt WECC29.bin -quan=bus_va -all >
C:\Document\Course06\TSAT\WSCC29\WECC29BA.txt
!C:\dsa_powertools_5_net\tsat\bin\sim2txt WECC29.bin -quan=line_p -all >
C:\Document\Course06\TSAT\WSCC29\WECC29PF.txt
%calculate maximum generator angle difference and write the results file
BaseFile = 'WECC29GA';

%Use the developed function to read generator angle data
R = get_data(BaseFile);

%Transform your cell array into an ordinary matrix
A = R{1};
for i = 2 : 30
    A = [A R{i}];
end

% Find the maximum generator angle difference in the simulated case
B=size(A);
C=zeros(B(1,1),1);
    for M=(1:B(1,1))
        mx=max(A(M,:));
        mn=min(A(M,:));
        diff=abs(mx-mn);
        C(M,1)=diff;
        M=M+1;
    end

```

```

        end
    maxdiff=max(C);
    % the case is stable with control if the maximum generator angle
    % difference is smaller than 300 deg.
    if maxdiff<300
        flag=1;
        istab(ievent)=1;
    end
    % the case is unstable with control if the maximum generator angle
    % difference is larger than 300 deg.
    if maxdiff>300
        flag=0 ;
        istab(ievent)=0;
    end
    %%%calculate successive bus angle differences and freq
    %%%(set velocity for first sample to be zero)
    BaseFile = 'WECC29BA';
%   Use the developed function to read bus voltage angle data
    R = get_data2(BaseFile);
%   Transform the cell array into an ordinary matrix
    D = R{1};
    for i = 2 : 18
        D = [D R{i}];
    end
    E=size(D);
    nrec=E(1,1);
    bus_angles=D;
    clear BaseFile R
    BaseFile = 'WECC29PF';
%Use the developed function to read line powerflow data (% this was
%used for 12 secs long simulations to find the steady state powerflow)

    R = get_data3(BaseFile);
%Transform the cell array into an ordinary matrix
    PF = R{1};
    for i = 2 : 4
        PF = [PF R{i}];
    end
    if icontrol(ievent)>0
        if (istab(ievent)==1 && istabnew(ievent)==1)
            rr1=1200;
            rr2=find(PF(:,1)==max(PF(:,1)));
            SS_Withcontrol=[SS_Withcontrol;mean(PF((rr1:rr2),2)),mean(PF((rr1:rr2),3)),
            mean(PF((rr1:rr2),4))];
        end
    end

```

```

        end
    end
%   to unwrap the BAs
%   If current - previous ~ 360 then addangles = addangles - 360
%   newbus_angles(irec,ind1) equals sum
%   bus_angles(irec,ind1) + addangles(irec,ind1)
for ind1=(2:18)%col
    newbus_angles(1,ind1)=bus_angles(1,ind1);
    addangles(1,ind1)=0;
    for irec=(2:nrec)%row
        diff=bus_angles(irec,ind1)-bus_angles(irec-1,ind1);
        addangles(irec,ind1)=addangles(irec-1,ind1);
        if diff<=-330
            addangles(irec,ind1)=addangles(irec-1,ind1)+360;
        end
        if diff>=330
            addangles(irec,ind1)=addangles(irec-1,ind1)-360;
        end
        diff=bus_angles(irec,ind1)+addangles(irec,ind1);
        newbus_angles(irec,ind1)=diff;
    end
end

end

% Make one bus be the reference angle (17th bus was selected)
% (the first column of data was time and the BA data started from column2)
for irec=(1:nrec)%rows
    coa(irec)=0;
    for ind1=(2:18)%columns
        diff=newbus_angles(irec,ind1)-newbus_angles(irec,18);
        newbus_angles(irec,ind1)=diff;
        coa(irec)=coa(irec)+newbus_angles(irec,ind1)/17;
    end
end

%   calculate the sga which is really sba
%   calculate the isga which is really isba
sga(1)=32;
isga(1)=sga(1);
for irec=(2:nrec)%rows
    sga(irec)=0;
    for ind1=(2:18)%columns
        diff=newbus_angles(irec,ind1)-coa(irec);
        sga(irec)=sga(irec)+diff*diff/17;
    end
end
sga(irec)=sqrt(sga(irec));

```

```

    sga(1)=sga(2);
    isga(1)=sga(1);
    isga(irec)=isga(irec-1)+(1/120)*6*(sga(irec-1)-isga(irec-1));
end
gradisga=gradient(isga);
% to calculate frequency (set velocity for first sample to be zero)
t=bus_angles(:,1);
newbus_angles(:,1)=t;
freq=zeros(E(1,1),E(1,2));
    for irec=(2:nrec)
        for ind1=(2:18)
            % finding the BA difference b/w two consecutive points
            diff=newbus_angles(irec,ind1)-newbus_angles(irec-1,ind1);
            % to avoid division by zero
            deltat1=(t(irec)-t(irec-1));
            deltat2=(0.1*(t(9)-t(1)));
            deltat=max(deltat1,deltat2);
            f=diff/deltat;
            freq(irec,ind1)=f;
        end
    end
freq(:,1)=t;
freq(1,:)=zeros(1,18);
Fname5=' .csv';
%V6A and V9Adot, bus voltage angle of the 6th bus and the bus frequency
%of the 9th bus (the first column of data is time)
V6A=newbus_angles(:,7);
V9Adot=freq(:,10);
y=size(V6A);
if flag==0;
    istabb=zeros(y(1,1),1);%assigning target value to each sample point.
end
if flag==1;
    istabb=ones(y(1,1),1);%assigning target value to each sample point.
end
newtarget=zeros(y(1,1),1);
% if a case was unstable without control and was stabilized with
% application of one of the control cases,
%we want the target value to be one for that and zero at all other time
if (istab(ievent)==0 && istabnew(ievent)==1)
    newtarget=ones(y(1,1),1);
end
if iEVE(ievent)>0
X2=[X2;newtarget(iEVE(ievent)+16:iEVE(ievent)+20)];

```

```

% X2 is the matrix that stores the newtarget value
% This matrix is later saved to a file along with a part of X1 as the
% training data for control selection DT
% Data for each new simulation is appended to this matrix
Fname11='C:\Document\Course06\TSAT\WSCC29\Test\WtrainControl198';
Fname12=[Fname11 Fname5];
end
%counts cases stabilized with control
if (istab(ievent)==0 && istabnew(ievent)==1)
    itotSave=itotSave+1;
end
%counts cases lost stability with control
if (istab(ievent)==1 && istabnew(ievent)==0)
    itotLose=itotLose+1;
end
%counts cases remained stable with control
if (istab(ievent)==1 && istabnew(ievent)==1)
    itotStab=itotStab+1;
end
%counts cases remained unstable with control
if (istab(ievent)==0 && istabnew(ievent)==0)
    itotUnst=itotUnst+1;
end
clear bus_angles newbus_angles freq
delete('WECC29BA.txt')
delete('WECC29GA.txt')
delete('WECC29PF.txt')
clear istabb V6A V9Adot gradisga A isga PF
end
%Writes the results to a file
fid4=fopen('C:\Document\Course06\TSAT\WSCC29\Test\destabilized.txt','a+');
fprintf(fid4,'EVENTS STABILIZED %4.0f\r\n',itotSave)
fprintf(fid4,'EVENTS DESTABILIZED %4.0f\r\n',itotLose)
fprintf(fid4,'EVENTS KEEP STABLE %4.0f\r\n',itotStab)
fprintf(fid4,'EVENTS KEEP UNSTABLE %4.0f\r\n',itotUnst)
fclose(fid4);
% writes X1 to a file
fid2=fopen(Fname10,'w+');
fprintf(fid2,'{istabb},{V6A},{V9Adot},{isga},{gradisga},{varV6A},{varV9Adot}\n');
dlmwrite(Fname10,X1,'-append')
fclose(fid2);
% X3 is the same as X1 for instability prediction except for its target
% value
X3=[X2 X1(:,2:5)];

```

```

fid6=fopen(Fname12,'w+');
fprintf(fid6,'{newtarget},{V6A},{V9Adot},{isga},{gradisga}\n');
dlmwrite(Fname12,X3,'-append')
fclose(fid6);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%END OF THE CODE

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This section of the code is for the method of chapter 2
% clear all
% clc
% close all
% Fnamea='DESCR';
% Fnameb='NOMOR';
% X1=[0 0 0];
% %maxGA=zeros(1000,390);
% swifile = textread('swifilelarge385.txt','%s','delimiter','\n','whitespace','');
% nline=length(swifile);
% nevent=0;
% %find event locations in swifile
%     for iline=1:nline;
%         record = swifile(iline);
%         record=char(record);
%         h=findstr(record,Fnamea);
%         if h==1
%             flag1=iline;
%             nevent=nevent+1;
%             swistart(nevent)=iline;
%         end
%         k=findstr(record,Fnameb);
%         if k==1
%             flag2=iline;
%             swistop(nevent)=iline;
%         end
%     end
%     YY=zeros(nevent,1);
%     s=zeros(605,385);
% %%% RUN THE EVENT WITH NO CONTROL
% for ievent=1:nevent
%     ievent
%     fid=fopen('fault2.swi','w+');
%     p=swistart(ievent);

```

```

% swi=cell2mat(swifile(p));
% fprintf(fid,'%s\r\n',swi);
% fprintf(fid,'SIMULATION 6.0 SECONDS/ \r\n')
% for p=swistart(ievent)+2:swistop(ievent)
%     swi=cell2mat(swifile(p));
%     fprintf(fid,'%s\r\n',swi);
% end
% fclose(fid);
% clear freq
% clear deltat
% t = timer('TimerFcn',@(x,y)disp('Hello World!'),'StartDelay',10);
% start(t);
% wait(t);
% !C:\dsa_powertools_5_net\tsat\bin\tsat_batch.exe
% C:\Document\Course06\TSAT\WSCC29\WECC29.tsa
% !C:\dsa_powertools_5_net\tsat\bin\sim2txt WECC29.bin -quan=gen_relang -all >
% C:\Document\Course06\TSAT\WSCC29\WECC29GA.txt
% !C:\dsa_powertools_5_net\tsat\bin\sim2txt WECC29.bin -quan=bus_va -all >
% C:\Document\Course06\TSAT\WSCC29\WECC29BA.txt
%
% %%%calculate maximum generator angle difference
% BaseFile = 'WECC29GA';
% % Use the developed function to read data
% R = get_data(BaseFile);
% % Transform your cell array into an ordinary matrix
% A = R{1};
% for i = 2 : 30
%     A = [A R{i}];
% end
% B=size(A);
% C=zeros(B(1,1),1);
% for M=(1:B(1,1))
%     mx=max(A(M,:));
%     mn=min(A(M,:));
%     diff(M)=abs(mx-mn);
%     C(M,1)=diff(M);
%     %M=M+1;
% end
% t=A(:,1);
% % plot(t,diff)
% % plot(t,A);
% % ylim([150 550])
% % axis manual
% % hold on

```

```

%% figure
%% maxGA(1:length(C),ievent)=C;
% maxdiff=max(C);
% if maxdiff<=500
%     flag=1;
%     istab(ievent)=1;
%     S1=find(A(:,1)==5);
%     s(1:S1,ievent)=C(1:S1);
%     maxdiff1=max(C(1:S1));
%     YY(ievent)=maxdiff1;
% end
% if maxdiff>500
%     flag=0 ;
%     istab(ievent)=0;
% end
% if maxdiff<=500
% %%%calculate successive bus angle differences and freq
% %%%(set velocity for first sample to be zero)
% BaseFile = 'WECC29BA';
% % Use the developed function to read data
% R = get_data2(BaseFile);
% % Transform the cell array into an ordinary matrix
% D = R{1};
% for i = 2 : 18
%     D = [D R{i}];
% end
% E=size(D);
% nrec=E(1,1);
% bus_angles=D;
% %If current - previous ~ 360 then addangles = addangles - 360
% % newbus_angles(irec,ind1) equals sum
% % bus_angles(irec,ind1) + addangles(irec,ind1)
% for ind1=(2:18)%col
%     newbus_angles(1,ind1)=bus_angles(1,ind1);
%     addangles(1,ind1)=0;
%     for irec=(2:nrec)%row
%         diff=bus_angles(irec,ind1)-bus_angles(irec-1,ind1);
%         addangles(irec,ind1)=addangles(irec-1,ind1);
%         if diff<=-330
%             addangles(irec,ind1)=addangles(irec-1,ind1)+360;
%         end
%         if diff>=330
%             addangles(irec,ind1)=addangles(irec-1,ind1)-360;
%         end
%     end
% end

```



```

%           diff=bus_angles(irec,ind1)+addangles(irec,ind1);
%           newbus_angles(irec,ind1)=diff;
%       end
%   end
%   % Make one bus be the reference angle
%   for irec=(1:nrec)%rows
%       coa(irec)=0;
%       for ind1=(2:18)%columns
%           diff=newbus_angles(irec,ind1)-newbus_angles(irec,18);
%           newbus_angles(irec,ind1)=diff;
%           coa(irec)=coa(irec)+newbus_angles(irec,ind1)/17;
%       end
%   end
%   %calculate the isga which is really isba
%   sga(1)=32;
%   isga(1)=sga(1);
%   for irec=(2:nrec)%rows
%       sga(irec)=0;
%       for ind1=(2:18)%columns
%           diff=newbus_angles(irec,ind1)-coa(irec);
%           sga(irec)=sga(irec)+diff*diff/17;
%       end
%       sga(irec)=sqrt(sga(irec));
%       sga(1)=sga(2);
%       isga(1)=sga(1);
%       isga(irec)=isga(irec-1)+(1/120)*6*(sga(irec-1)-isga(irec-1));
%   end
%   %%   t=D(:,1);
%   %%   plot(t,isga)
%   %%   ylim([150 550])
%   %%   axis manual
%   %%   hold on
%   %%   S2=find(D(:,1)==10);
%   %%   maxisga1=max(isga(1:S2));
%   %%   ZZ(ievent)=maxisga1;
%   clear t isga
%   end
%   delete('WECC29BA.txt')
%   delete('WECC29GA.txt')
%   clear istab istabb V6A V9Adot diff
%   clear bus_angles newbus_angles freq t A D B R E
% end
% title('isba')
% figure

```

```

% %%% To find the GA threshold of all %%%
% Yth=100:25:500;
% for mm=1:length(Yth)
%     YY(mm)=max(YY(YY<Yth(mm)));
% end
% line(Yth,YYY);
% title('GATHreshold')
% figure
% for nn=(2:length(YYY))
%     if YYY(nn)==YYY(nn-1)
%         GATHreshold1=YYY(nn);
%     end
% end
% %GATHreshold1
% %or using the tabulate
% Table=tabulate(YYY);
% TT=find(Table(:,2)==max(Table(:,2)));
% GATHreshold2=Table(TT,1);
% Fname19='C:\Document\Course06\TSAT\WSCC29\Test\maxdiff_6s_6s';
% Fname20=[Fname19 Fname5];
% fid20=fopen(Fname20,'w+');
% fprintf(fid20,'{maxdiff in 6s}\n');
% dlmwrite(Fname20,YY,'-append')
% fclose(fid20);
% %%% to find the isba threshold
% Zth=35:5:100;
% for mm=1:length(Zth)
%     ZZZ(mm)=max(ZZ(ZZ<Zth(mm)));
% end
% line(Zth,ZZZ);
% title('isbathreshold')
% for nn=(2:length(ZZZ))
%     if ZZZ(nn)==ZZZ(nn-1)
%         isbaThreshold1=ZZZ(nn);
%     end
% end
% %isbaThreshold1
% %or using the tabulate
% TableZ=tabulate(ZZZ);
% TTZ=find(TableZ(:,2)==max(TableZ(:,2)));
% isbaThreshold2=TableZ(TTZ,1);
% Fname21='C:\Document\Course06\TSAT\WSCC29\Test\isba_8s_6s';
% Fname22=[Fname21 Fname5];
% fid21=fopen(Fname22,'w+');

```

```

% fprintf(fid21, '{maxisba in 6s}\n');
% dlmwrite(Fname22, ZZ, '-append')
% fclose(fid21);
% %%%
% clear YYY
% %Histogram
% Yhist=100:10:500;
% z=0;
% for ievent=1:nevent
%     for mm=1:length(Yhist)
%         YYY(ievent,mm)=length(find(s(:,ievent)<=Yhist(mm)))-z;
%         z=sum(YYY(ievent,:));
%     end
%     z=0;
% end
% j=[Yhist' (sum(YYY))'];
% bar(j(2:end,1),j(2:end,2))
% title('Histogram')
% set(gca,'FontSize',12);
% ylabel('NUMBER OF POINTS')
% xlabel('GA')

```