



# Seamless Energy Management Systems

## Part II: Development of Prototype Core Elements

*Final Project Report*

**Power Systems Engineering Research Center**

*Empowering Minds to Engineer  
the Future Electric Energy System*



# **Seamless Energy Management Systems**

## **Part II: Development of Prototype Core Elements**

### **Final Project Report**

#### **Project Faculty Team**

Santiago Grijalva, Georgia Institute of Technology

Anjan Bose, Washington State University

Polo Chau, Georgia Institute of Technology

#### **Graduate Research Students**

Leilei Xiong, Brian Minsuk Kahng, Robert Pienta

Georgia Institute of Technology

Yannan Wang, Pradeep Yemula

Washington State University

**PSERC Publication 14-5**

**September 2014**

**For information about this project contact:**

Santiago Grijalva  
Georgia Power Distinguished Professor  
Director, Advanced Computational Electricity Systems (ACES) Laboratory  
School of Electrical and Computer Engineering  
Georgia Institute of Technology  
Atlanta, Georgia 30332-0250  
Phone: 404-894-2974  
E-mail: [sgrijalva@ece.gatech.edu](mailto:sgrijalva@ece.gatech.edu)

**Power Systems Engineering Research Center**

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

**For additional information, contact:**

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

**Notice Concerning Copyright Material**

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

**© 2014 Georgia Institute of Technology, All rights reserved.**

## **Acknowledgements**

This is Part II of the final report for the Power Systems Engineering Research Center (PSERC) research project entitled “Seamless Energy Management Systems” (S-53G for 2013-2014). The project has been sponsored by EPRI. We express our appreciation to EPRI and in particular Paul Myrda for his support and guidance for the project. We also express our appreciation for the support provided by PSERC’s industrial members, as well as EPRI members, and by the National Science Foundation’s Industry/University Cooperative Research Center program.

## Executive Summary

This is the second in a series of reports about Seamless Energy Management Systems. The first report “Seamless Energy Management Systems, Part I: Assessment of Energy Management Systems and Key Technological Requirements” focused on an assessment of the current state-of-the-art in EMS architectures, and developed technological requirements for seamless systems. This second report focuses on a subset of core technologies for a seamless EMS prototype, including communication systems, unified model, and visualization.

Energy management systems (EMS), control centers that manage the transmission-generation grid, have existed since the 1960s. They have gradually evolved over the years mostly in an incremental manner. Now, major transformation of EMS systems is essential to support emerging behavior of the power system, affected by variable and less predictable renewable energy penetration, availability of newer types of sensors, and communication systems, and powerful computation platforms. As part of this project, various “seams” have been identified that are of interest and that are apparent in current power system operations and control practices. These seams are:

1. Communication architecture of current systems is not suitable for enhanced sensing capability enabled by synchronized phasor measurement units (PMU)
2. The difficulty in comparing application results due to the utilization of different models in operations and planning, and to tie these models to PMU data
3. Repetitiveness of simulations in contingency analysis applications
4. Continued utilization of legacy code in today’s software applications, which rely primarily on sequential computing rather than taking advantage of modern high performance computing technology
5. A lack of look-ahead visualization capabilities, which are critical for today’s modern electric grid with its non-dispatchable renewables, increasing levels of demand response, and deployment of energy storage.

There exists a need for next generation power system management tools that use unified geo-spatial models, more efficiently handle massive scenario evaluation such as contingency analysis, and provide look-ahead visualization capabilities. This report focuses on attempting to address some of these limitations.

Chapter II of this report provides a summary of advances in fast communication systems for PMU applications, and it proposes a novel communication architecture that would support further deployment of PMUs.

In Chapter III, the unified network applications framework is extended to the study of contingencies that result in bus splits and bus mergers. Dynamic pointer assignment and incremental subnet processing allow seamless realization of consolidated representations necessary to model arbitrary post-contingency topologies. Numerical simulation results for a large scale ISO node-breaker model are used to quantitatively determine the advantages of the proposed framework.

Chapter IV introduces the concept of using distribution factors to directly transition from the power flow solution of one system state to the contingency analysis results for a similar but different state is introduced. A new time-dependent PTDF is defined and combined with OTDFs to estimate post-contingency transmission line flows for a scenario that deviated slightly from the base case. Representative results for an illustrative 7-bus example and the IEEE 24-bus reliability test system are presented and compared against traditional distribution factor-based contingency analysis.

In Chapter V, a novel 3D navigational method of visualization for the exploration of past, present, and future power system states is presented. The 3D stacking approach is illustrated on the PowerWorld 7-bus test system. As a complement to the 3D overview visualization, a 2D visualization interface that shows more detailed information is also discussed. Color was used to emphasize components that are threatened while the components under normal operation were grayed out. The user interaction was especially designed with detailed navigation and exploration of system changes over time in mind. Interactive pop-up bus labels, line charts, and detailed tabular views are all essential to achieve those goals.

In Chapter VI we provide concluding remarks, and Chapter VII discusses possible future directions of research.

# Table of Contents

1	Introduction to Seamless EMS Systems .....	1
2	Decentralized Communication Systems to Support Seamless Energy Management .....	4
2.1	Introduction.....	4
2.2	Overview of the Design Process .....	5
2.2.1	Architectural Considerations.....	5
2.2.2	Centralized Communication Architecture.....	6
2.2.3	Decentralized Communication Architecture.....	7
2.2.4	Choice of Protocol Stack.....	9
2.2.5	Process for Design of Communication Architecture.....	9
2.3	Design and Simulation of Communication Network.....	10
2.3.1	Choice of metric: MegaBit-Hops .....	10
2.3.2	Communication Delay and the Number of Hops .....	11
2.3.3	Design and Simulation of the Communication Network .....	13
2.3.4	Calculation of Delay in Communication.....	14
2.3.5	Conclusion on Centralized and Distributed Topologies .....	16
2.4	Pre-Processing: Design of Wide Area Damping Controller .....	16
2.4.1	System Description .....	17
2.4.2	Small Signal Analysis .....	18
2.5	Simulation of Controller for the Power Network.....	19
2.6	Conclusion .....	21
3	Unified Geo-Spatial Model.....	22
3.1	Toward Operations and Planning Modeling Unification .....	22
3.2	Issues with Model Unification .....	23
3.3	Toward Temporal Interoperability.....	27
3.4	Modeling Switching Contingencies .....	28
3.4.1	Device Terminal Pointer Assignment .....	28
3.4.2	Contingency Modeling.....	31
3.4.3	Incremental Subnet Processing .....	32
3.5	Case Example: Unified Contingency Analysis Interoperability .....	34
3.5.1	System Properties.....	34
3.5.2	Incremental Processing vs. Low-X Branch Solution .....	34
3.5.3	Incremental Subnet Processing vs. Scripting.....	34
4	Seamless Computational Framework .....	38
4.1	Database Structures for Temporal Data Handling .....	38

4.2	Development of Smart Contingency Analysis Algorithm .....	40
4.2.1	Motivation.....	40
4.2.2	Distribution Factors.....	42
4.2.3	Problem Formulation and Algorithm.....	43
4.2.4	Results.....	45
4.2.5	Illustrative 7-Bus Example.....	45
4.2.6	IEEE 24-Bus Reliability Test System.....	46
4.2.7	Computational Complexity .....	47
4.2.8	Exploration of Other Approaches .....	48
4.3	Application of High Performance Computing .....	50
4.3.1	Introduction.....	50
4.3.2	Literature Review.....	51
4.3.3	Incorporation of HPC in Power System Analysis.....	52
4.3.4	System Architecture Overview .....	52
4.3.5	Investigation of HPC Applications .....	53
5	Look-Ahead Visualization .....	55
5.1	Visualizing Power System Data.....	55
5.2	Literature Review.....	56
5.2.1	2D Dynamic Network Visualization.....	56
5.2.2	3D Dynamic Network Visualization.....	56
5.3	Visualization Data Management.....	56
5.3.1	Database Schema .....	56
5.3.2	Converting the Data .....	57
5.4	Initial Concept.....	57
5.4.1	3D Visualization Concept .....	57
5.4.2	2D Visualization Interface Concept.....	59
5.5	Navigational 3D Power System Visualization Prototype .....	60
5.5.1	Implementation Overview.....	61
5.5.2	Buses .....	61
5.5.3	Lines.....	63
5.5.4	User Interaction.....	64
5.5.5	Interface .....	65
5.6	Complementary 2D Power System Visualization Prototype .....	66
5.6.1	Implementation Overview.....	66
5.6.2	Interfaces.....	66
5.6.3	Main Power System View.....	66

5.6.4	Line Charts.....	68
5.6.5	Detail Tables .....	68
6	Conclusions.....	70
6.1	Unified Geospatial Model.....	70
6.2	Multi-Scenario Contingency Analysis .....	70
6.3	Navigational Visualization.....	71
7	Future Work.....	72
7.1	Multi-Scenario Contingency Analysis .....	72
7.2	Navigational Visualization.....	72
8	References.....	75

## List of Figures

Figure 1.1: Prototype system architecture.....	3
Figure 2.1: Centralized communication architecture .....	6
Figure 2.2: Decentralized communication architecture .....	7
Figure 2.3: Distributed communications architecture for power system control .....	8
Figure 2.4: Process for design of communication architecture .....	10
Figure 2.5: Relationship between communication delay and number of hops for various flows with different link capacities .....	12
Figure 2.6: Optimal location of control center for Topology 1 and data routing hubs for Topology 2. ....	14
Figure 2.7: Communication Network Overlay for IEEE 118 Bus System with 3 areas.. ....	15
Figure 2.8: Excitation system with WADC with remote signal.....	17
Figure 2.9: Dynamic Response of Generators .....	19
Figure 2.10: Dynamic Response of Generators .....	20
Figure 3.1: Node-breaker model showing groups that correspond to buses in the bus-branch model .....	24
Figure 3.2: Bus-branch model corresponding to the node-breaker system and its breaker statuses .....	25
Figure 3.3: New bus-branch model needed to represent and study a bus split contingency .....	26
Figure 3.4: Device pointer assignment using a unified application framework.....	29
Figure 3.5: Power flow on a node-breaker model.....	30
Figure 3.6: Device pointer assignment for post-contingency state (breakers 3-6 and 4-5 open).....	31
Figure 3.7: Solution process for a list of contingencies using a node-breaker model.....	33
Figure 3.8: Detailed view of a section of the ISO node-breaker model.....	35
Figure 3.9: Resulting bus-branch model .....	35
Figure 3.10: New bus-branch model after the switching contingency.....	36
Figure 4.1: Proposed system architecture of the prototype.....	38
Figure 4.2: Grid database structure .....	39
Figure 4.3: AC contingency analysis for two arbitrary time steps.....	41
Figure 4.4: Conventional DC distribution factor-based contingency analysis for two time steps .....	41
Figure 4.5: Proposed approach for two arbitrary timesteps, assuming no topology change.....	45
Figure 4.6: Time distribution for the PowerWorld 7-bus test system across 30 runs .....	49
Figure 4.7: Time distribution for the IEEE RTS 24-bus test system across 30 runs .....	49
Figure 4.8: Time distribution for the IEEE 118-bus test system across 30 runs .....	49
Figure 4.9: Grids, threads and blocks in the CUDA layout for GPUs .....	51
Figure 4.10: High level overview of the system architecture .....	53
Figure 5.1: 3D prototype representation .....	57
Figure 5.2: 3D top view .....	58
Figure 5.3: 3D side view .....	59

Figure 5.4: 2D user interface .....	60
Figure 5.5: 2D run mode .....	60
Figure 5.6: Side view of a bus volume.....	62
Figure 5.7: Visualization of bus volumes for a 7-bus system.....	63
Figure 5.8: Visualization of line loading for a 7-bus system.....	64
Figure 5.9: Collapsible user interface in the visualization.....	65
Figure 5.10: Main power system view.....	67
Figure 5.11: Sample pop-up label.....	68
Figure 5.12: Sample line chart.....	68
Figure 5.13: Detailed table view .....	69
Figure 7.1: Schema of the “diagram view” approach to visualizing contingency analysis results .....	74

## List of Tables

Table 2.1: Reduction in Number of Hops .....	13
Table 2.2: Calculation of Delay in Communication for Topology 1 .....	15
Table 2.3: Calculation of Delay in Communication for Topology 2 .....	16
Table 2.4: Centralized and Decentralized Topology Communication.....	18
Table 2.5: Small Signal Analysis Results for IEEE 118 Bus System.....	18
Table 3.1: Properties of the Sample Case .....	34
Table 3.2: Solution Times for the Sample Case.....	34
Table 4.1: Different Bus Injections over Time .....	46
Table 4.2: Comparison of Contingency Analysis Results for PowerWorld 7-Bus Example.....	46
Table 4.3: Representative Changes in Bus Injection over Time .....	46
Table 4.4: Comparison of Contingency Analysis Results for IEEE 24-Bus Test System .....	47
Table 4.5: CPU vs. GPU for Matrix Multiplication.....	54

# 1 Introduction to Seamless EMS Systems

Decision makers for the bulk electric grid have come to rely on various computational methods and software applications to help plan and operate the power system. Due to its large size, complex emerging behavior (such as increased integration of wind and demand response), and possibility of events, even experienced personnel can struggle with making decisions quickly while meeting the objectives of system security and reliability. Hence, the decision-making process must be equipped with enhanced applications that address emerging behavior and that exploit new data and computation capabilities.

Currently, at the ISO level, there are two primary groups of decision makers: one that deals with control and operations, and a separate group that deals with transmission and generation planning. Both perform similar functions albeit with different objectives and on different time scales. The primary objective of the control and operations group is to meet the current demand and ensure the minute-to-minute reliability of the overall system, including voltage management, constraint handling, etc. The primary objective of the planning group is to meet the future demand and ensure the future reliability of the system.

Although they share similar responsibilities and often run similar algorithms, planning and operations each utilize their own specific software applications, models, and data formats [1]. This practice of each group working in isolation has led to two seams or rifts in the ISO community. The first seam is the difficulty in comparing application results due to the utilization of different models in operations and planning. A unified network model for both planning and operations is necessary in order to achieve interoperability of the two groups at the ISO level [2]. The second seam that is a focus of this project is the repetitiveness of simulations. For instance, contingency analysis (CA) needs to be performed across multiple time points for operations and multiple scenarios for planning. For instance, in the case of operations, the same list of contingencies (which may include thousands of plausible events) is resolved every 2-5 minutes while the system may have experienced small changes in generation and demand. An integrated multi-temporal, multi-scenario algorithm that takes advantage of existing information from the system and that is less “brute force” is required. Furthermore, this algorithm must be applicable to both planning and operations to more efficiently handle studies of large number of scenarios or contingencies. Throughout this report we use contingency analysis as an example of an application that requires massive scenario analysis. Other applications, such as Voltage Security Assessment (VSA), Dynamic Security Assessment (DSA), and Available Transfer Capability (ATC) fall within the same category and are examples of security applications. Future power system operation may also require the calculation of economic optimization applications, such as Security-Constrained Economic Dispatch (SCED), Security Constrained Optimal Power Flow (SCOPF), and Security-Constrained Unit Commitment (SCUC) to be used under massive scenario analysis for look-ahead operation. While we do not include specifics of these

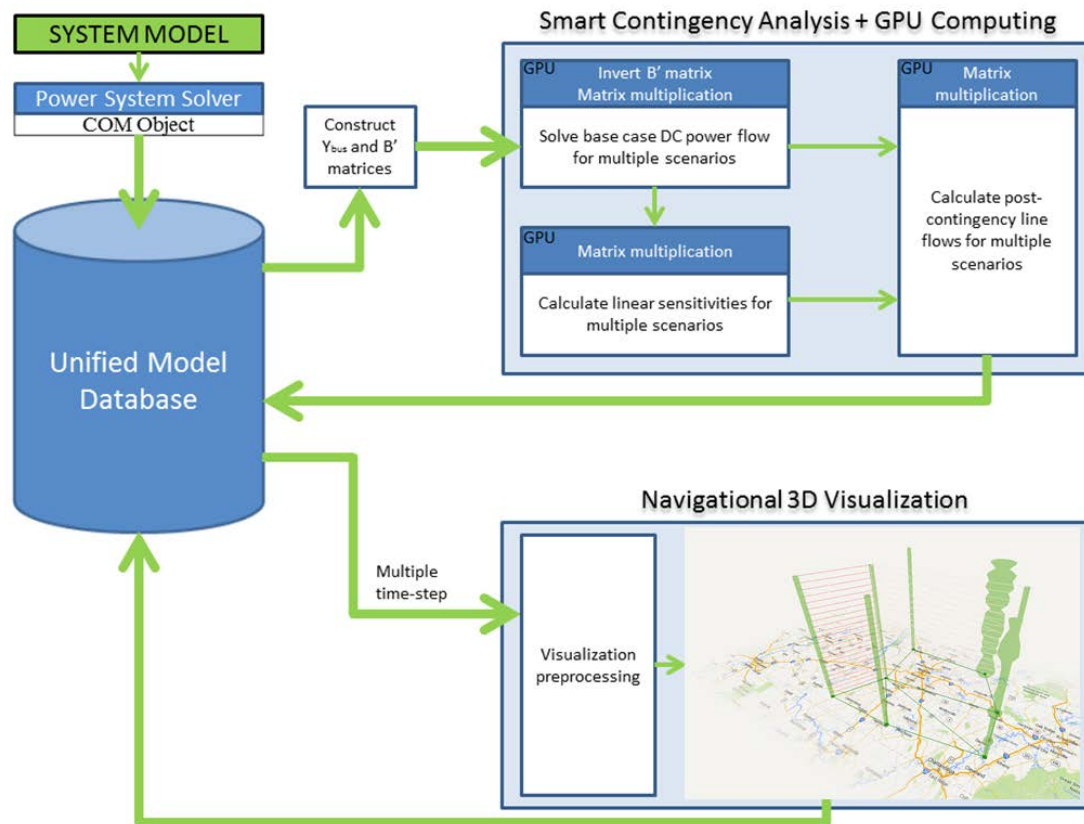
applications in this project, we use contingency analysis as an example to provide insight into software architecture issues of seamless EMS equipped with look-ahead capabilities.

The third seam is the continued utilization of legacy code in today's software applications, which rely primarily on sequential computing rather than taking advantage of modern high performance computing technology. The fourth and final seam is the lack of look-ahead visualization capabilities, which are critical for today's modern electric grid with non-dispatchable highly variable renewables, increasing levels of demand response, and deployment of energy storage.

There exists a need for next generation power system management tools that use more efficient computational algorithms and provide look-ahead visualization capabilities. Seamless Energy Management Systems refer to both the tools that address these issues as well as new operational frameworks that support and exploit such tools. This report introduces a seamless, modular prototype (shown in Figure 1.1) with the described features:

- **Multi-scenario contingency analysis.** Contingency analysis (CA) is an essential tool used by both the planning and operations functions of the ISO. Operations typically perform full contingency analysis every few minutes to check for N-1 system security despite just small deviations from the normal system state. Similarly planning performs full contingency analysis for a wide variety of scenarios with only a few changes from one scenario to another. This module introduces the concept of using linear sensitivities to move from one scenario to another without duplicating full CA. Also, recent advances in GPU technology have paved the way for large computations in the power system arena that were previously impossible to achieve in a near real-time scale. Highly parallel tasks such as state estimation, power flow studies, and contingency analysis are particularly good candidates for GPU applications.
- **Navigational 3D visualization.** Visualizing massive amounts of data is a major challenge in power systems. Most modern visualization tools are limited to displaying static data for one time instance, which is inadequate for the increasingly complex nature of the grid with its needs for quantity plotting and trend spotting. This module demonstrates a navigation-based approach to visualizing power system data in both the time and spatial dimensions.

Additionally, a database with versioning capabilities is needed to keep track of different scenarios, their associated contingency analysis results, and the visualization elements. The prototype leverages the model handling capabilities of PowerWorld, a commercial power system solver package. Via the COM interface, relevant system topology and line parameter quantities are first extracted from PowerWorld and stored in the unified database. Information is exchanged between the smart contingency analysis and navigational visualization modules via the database. Each module retrieves the information, processes it, and stores new results in the database.



**Figure 1.1: Prototype system architecture**

## 2 Decentralized Communication Systems to Support Seamless Energy Management

### 2.1 Introduction

Due to the rapid deployment of phasor measurement units (PMUs) on large power grids the system operators now have access to high speed high resolution data. New classes of monitoring and control applications are possible with the PMUs. Although PMU based monitoring systems have been well developed, implementations of PMU based fast acting closed loop wide area control systems are relatively rare. To meet the stringent latency requirements of a wide area controller the communication and power infrastructures have to collaborate strongly. In this research, a combined process for design and simulation of both communication network and power network has been presented with the objective of damping inter-area oscillations. A method to determine the optimal location of data routing hubs so as to minimize the volume of communications is also proposed.

It is an established observation that real-time sub-second measurements are necessary to gain insights about the dynamic behavior and to take fast automatic control actions in a modern power grid operated close to its margins [43]. PMUs obtain synchronized measurement of voltage and current phasors at rates of about 30 to 120 samples per second. Smart grid of the future is expected to have PMU data available widely across the grid. To meet the latency requirements and to handle the huge amounts of data, the need for a real time information infrastructure has been proposed [44]. Smart grid applications are designed to exploit these high throughput real-time measurements. Most of these applications have a strict latency requirement in the range of 100 milliseconds to 5 seconds [45], [46]. Among the other delays [47], communication delay also adds to the latency and needs to be minimized.

The communication delays on the network are comprised of transmission delays, propagation delays, processing delays, and queuing delays [43]. Each of these delays must be looked into to understand the complete behavior of the communication network for a given network. With the advances made in ubiquitous computing systems, the notion of distributed data and distributed analytics becomes amenable. The question then becomes what should be the design of the new communications architecture? Given that the data and computations are going to be distributed, which data should reside where? How data is to be moved to the applications efficiently meeting the latency requirements? This research attempts to answer these questions by presenting a possible design of communications architecture for wide area control and protection of the smart grid.

In reference [48] a detailed survey of smart grid applications based on latency and bandwidth requirements has been presented. Latency is a measure of time delay experienced in a communication system. Whereas, bandwidth is the rate of data transfer in bits per second, that

can be achieved by a communication resource. According to [48] applications pertaining to power system operation can be classified in the increasing order of their latency requirement as follows: transient stability ( $< 100$  milliseconds), small signal stability ( $< 1$  sec), state estimation ( $< 1$  sec), voltage stability (1 - 5 sec), post-mortem analysis of grid disturbances ( $>$  few minute).

Due to the requirement of fast response, it is difficult for human operators to respond to problems pertaining to transient and small signal stability of a large power grid. Typically, the stability is achieved by local controllers operating with local information. However, due to large interconnections of power grids, disturbances in one region can spread to other region. Hence, wide area controllers, which rely on remote signals, become necessary.

## 2.2 Overview of the Design Process

### 2.2.1 Architectural Considerations

The following are some of the factors considered in design of the communication architecture.

*1) Location of Data:* In order to minimize the data traffic on the communication network, the choice of data that is put on the network will have to be determined by the application which will consume that data. Although the PMUs can sample the phasors at a rate of 30 to 120 samples per second, every application may not require data at such high rates. Hence each substation stores the data measured at that location in a local database and makes this data available. The approach here is to keep the data distributed and close to the power network components from which the data is measured.

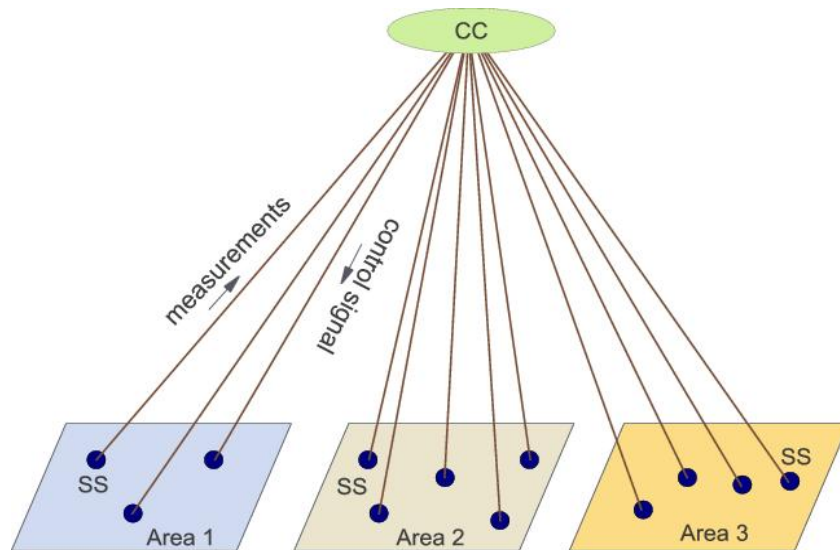
*2) Location of Applications:* It can be observed from the latency requirements, that only the class of applications concerned with transient or small signal stability of the system need faster data with higher detail. Other applications are relatively less stringent in the need for real time data. Thus, there is no need to centralize all the applications at one place. The fast control applications can be located closer to the controllable equipment.

*3) Movement of Data:* Since the data and applications are defined to be distributed, a communication infrastructure is needed which can identify a specific subset of data and transfer to the required application. The characteristics of such an infrastructure are described in [44]. A middle-ware system forms the heart of such an infrastructure, which can perform the functions of efficient routing of data packets while conforming to the quality of service (QoS) constraints. An architectural paradigm known as publish/subscribe is suitable for such a middle-ware. The sources of data need not be aware of the consumer of data. The sources simply publish their data to the middle-ware. The applications which require specific data will subscribe to the middle-ware. A list of all received subscriptions is maintained by the middle-ware. When the data is published the middle-ware notifies the receiving application and forwards the data.

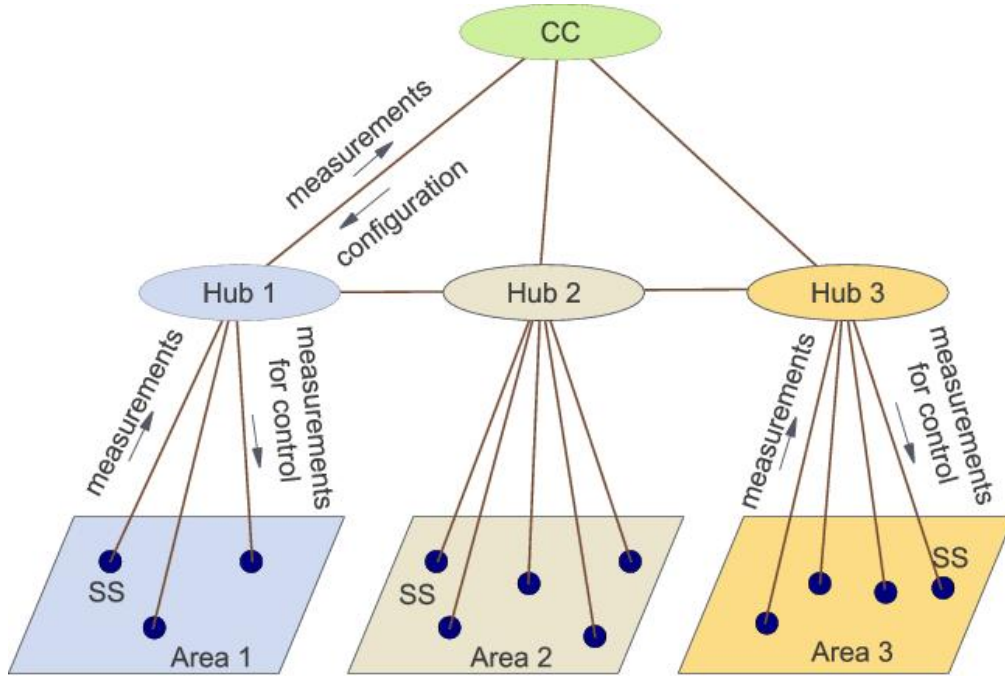
4) *Format for Data and Control Commands:* The PMUs are being manufactured by multiple vendors and interoperability among equipment from different vendors is ensured by using standard formats. The standard C37.118 is used in practice for communication of PMU data [49]. Among the four frames that are defined in C37.118, the data frame is one that is sent out from the substation under normal conditions. The command frame defined in C37.118 can be used to send commands to the PMUs for controlling the associated power system equipment.

### 2.2.2 Centralized Communication Architecture

Figure 2.1 illustrates the typical communication architecture with a centralized control center (CC) which receives all the measurements from each substation (SS). The structure shown in Figure 2.1 represents the logical connection, whereas, the substations are interconnected to the control center through a physical network which can have meshed structure similar to the power network. This data is useful for system wide energy management applications such as, state estimation, operator visualization, security analysis, contingency studies, and voltage stability. The results of these studies would determine required control actions such as, switching of circuit breakers, capacitor banks, and transformer taps. These control actions would have be implemented typically in the time scale of few seconds to minutes. Hence a centralized architecture is suitable for such applications. Although this centralized architecture is simple, it may not be scalable for all applications. It can be observed that the substations belonging to different control areas have to send all their data to a same location, which may be quite far away from some substations. Because of the huge amount of data generated at each substation, as the size of the system increases not all the data needs to be sent to one central location in real-time. To achieve scalability, only the data required for energy management applications can be centralized at a relatively slower data rate. This gives rise to a decentralized topology as discussed below.



**Figure 2.1: Centralized communication architecture**



**Figure 2.2: Decentralized communication architecture**

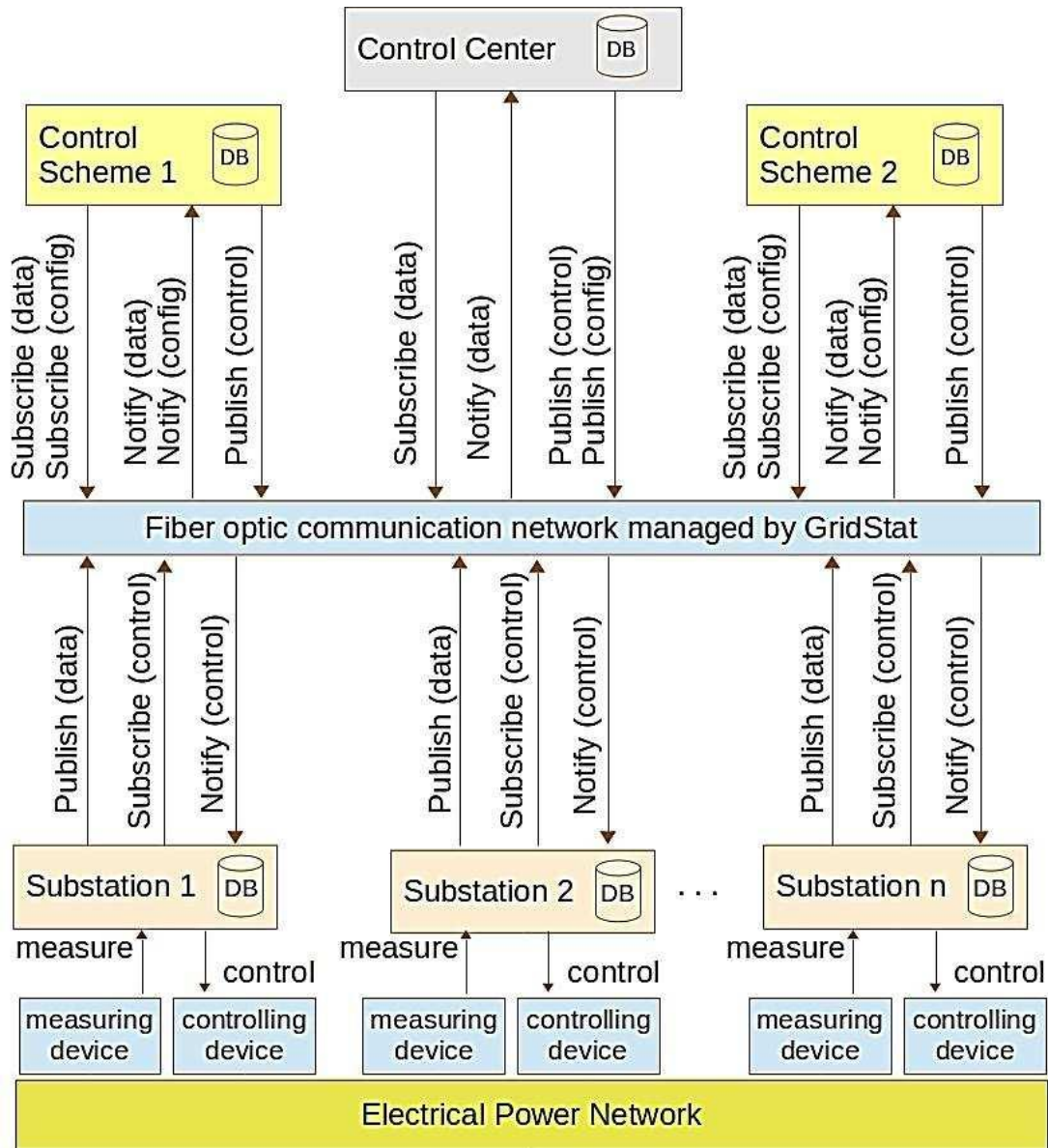
### 2.2.3 Decentralized Communication Architecture

As shown in Figure 2.2, a new layer of communication nodes which act as *data routing hubs*, is added between the substation and central control center. The main function of *data routing hubs* (or simply referred as *hubs*) is to receive the measurements from all the substations in the area and route them to main control center and / or to substations with controllable devices. The control algorithms themselves run in parallel at respective substations, whereas the hubs only route the measurements as per their offline configuration. The configuration itself is flexible and can be remotely changed time to time, by the main control center as per the changing state of power system. Hubs are also interconnected in a peer to-peer fashion to exchange information across areas. In this distributed architecture, the total time between the substation sending measurements and receiving control signal is reduced and hence the applications which have more stringent latency requirements, such as transient stability and small signal stability can be supported.

Development of similar middle-ware systems to provide the latency and other quality of service (QoS) requirements is one of the objectives of the North American synchrophasor initiative (NASPI) [50], [51] and some research initiatives like Gridstat [52], [53].

Based on the above considerations some of the applications needing lower latency should be decentralized. As a consequence of this decentralized or distributed approach a need arises for storing the data at various levels. Since, only a subset of data is communicated as per the

requirement of the applications, effective data management strategies are needed to define the movement of the data across the various nodes of the network. To address this need, an information architecture for power system operation based on distributed controls using a publish/subscribe communication scheme and distributed databases is described in Figure 2.3.



**Figure 2.3: Distributed communications architecture for power system control**

The key feature of the proposed architecture is that the databases are distributed at each level. Each substation stores the measured data locally. Applications that need real-time data for transient stability monitoring and control are not located in the control center but can be located on a computing node near the substations, identified as “control schemes” in Figure 2.3. The special protection schemes (SPS) being used in power systems are one example for such control schemes.

At this level the data can also be stored for future use in computations. The data and control frames, as described by the C37.118 standard, can be exchanged via publish/subscribe based middle-ware which manages the fiber optic communication network. The communication network can be physically laid along with the power system network. The control center has its own set of applications and associated databases. While the focus is on optimizing the latency of time critical data, the data which is non-time critical can also be moved around with appropriate QoS attributes using the same communication network. The objective is to achieve a configuration of communication network which is most efficient and compatible with the operation of power system network in a decentralized way.

#### **2.2.4 Choice of Protocol Stack**

A protocol stack is a layered conceptual model that standardizes the internal functions of a communication system. Considering the latency requirements, the user datagram protocol (UDP) is a preferred protocol at transportation level, as opposed to transportation control protocol (TCP). Time-sensitive applications often use UDP because it is preferred to drop the packets instead of waiting for delayed packets, which may not be an option in a real-time system [54]. UDP is also faster, as there is no error checking, no ordering of messages, has light weight header (8 bytes), and no need to setup prior transmission connection and handshake. As the PMUs are sending out a stream of data frames on the network, at the application layer, the constant bit rate (CBR) is a good choice to carry the continuously generated data frames. The maximum transmission unit (MTU) size of the link layer can help in the design of application level software to receive a complete C37.118 packet and not a broken one. Optical fibers and broadband over power line (BPL) are the suitable choices at the link layer. It is assumed that optical fiber is present throughout the network for uniformity. Hence the protocol stack can be summarized as follows: link layer – optical fiber, data layer - Ethernet, network layer - internet protocol (IP), transportation layer - UDP, and application layer - CBR. Adoption of this protocol stack will make the implementation, platform independent and hence achieve interoperability among different hardware and software operating systems.

#### **2.2.5 Process for Design of Communication Architecture**

The flow chart of Figure 2.4 describes the proposed process for the design of communication architecture for a large power system. In the pre-processing block small signal analysis is carried out to determine the parameters of the damping controller. The second block, focuses on the design and simulation of the communication network for both centralized and decentralized

topologies. The optimal location for the hubs is also determined based on minimization of volume of communication. The simulation also determines the time delay for communication network for various values of bandwidth. The third block, focuses on the non-linear time domain simulation of the power network incorporating the time delay in the WADC. This process of block II and block III can be iterated until a suitable overall solution for stability is achieved. The subsequent three sections elaborate the process and explain the various steps and results obtained for the design and WADC for IEEE 118 bus system.

## 2.3 Design and Simulation of Communication Network

Since, it is assumed that the communication network will be overlaid over the power network both will have a similar topology. However the data routing hubs can be connected to any substation in the area. Different locations of hubs will result in different amount of traffic volume. Thus the notion of optimal hub location arises which would result in minimum amount of communications.

### 2.3.1 Choice of metric: MegaBit-Hops

In order to define a criteria to evaluate optimal location of hubs, we introduce a metric, namely Mega-bit-hop.

$$Mbh_c = \sum_{i=1}^N (h_i * p_i) \quad (1)$$

where,  $c$  is the location of hub,  $i$  represents the flow ID,  $N$  is the total number of flows,  $h_i$  is the number of hops taken by the packets on  $i^{\text{th}}$  flow and  $p_i$  is the packet size of  $i^{\text{th}}$  flow in Mega bits.

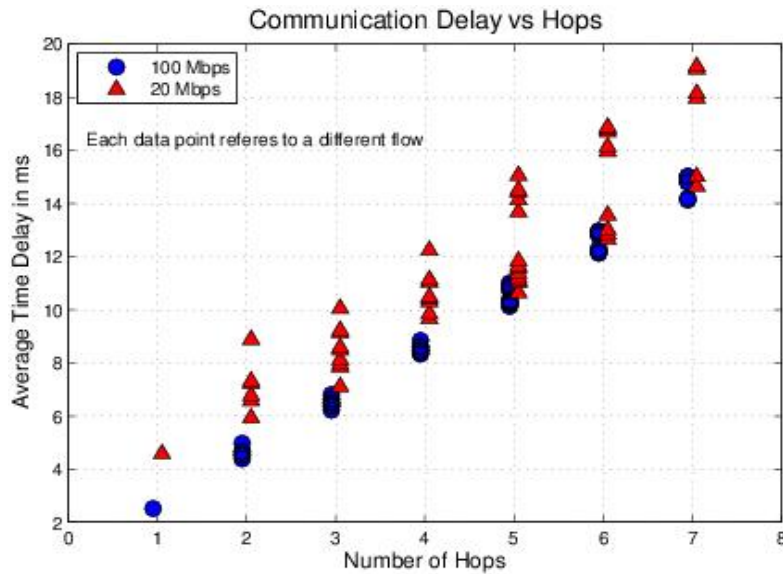


Figure 2.4: Process for design of communication architecture

The propagation delay is assumed to be negligible as information flows at the speed of light, whereas the majority of the time delay occurs when the packets are enqueued and dequeued at the intermediate nodes.

Each substation sends all its measured data to the control center, hence the packet size depends on the number of feeders connecting to the substation. Each of such flow from substation to control center can be assigned a flow ID. Thus the packet size  $p_i$  for each flow, is calculated based on the number of feeders and the data frame format defined by IEEE C37.118 standard for each substation. For the IEEE 118 bus system it has been calculated that the packet sizes for flows from various substations, vary from a minimum of 234 bytes to a maximum of 1440 bytes, with an average packet size of 409 bytes. Using the same communication network the control center also sends data to the local controllers which may be located at all generation substations. These flows are also assigned flow IDs and the packets only contain the predefined remote measurements useful for wide area control. Hence, the packet size for these controller inputs is assumed to be 200 bytes.

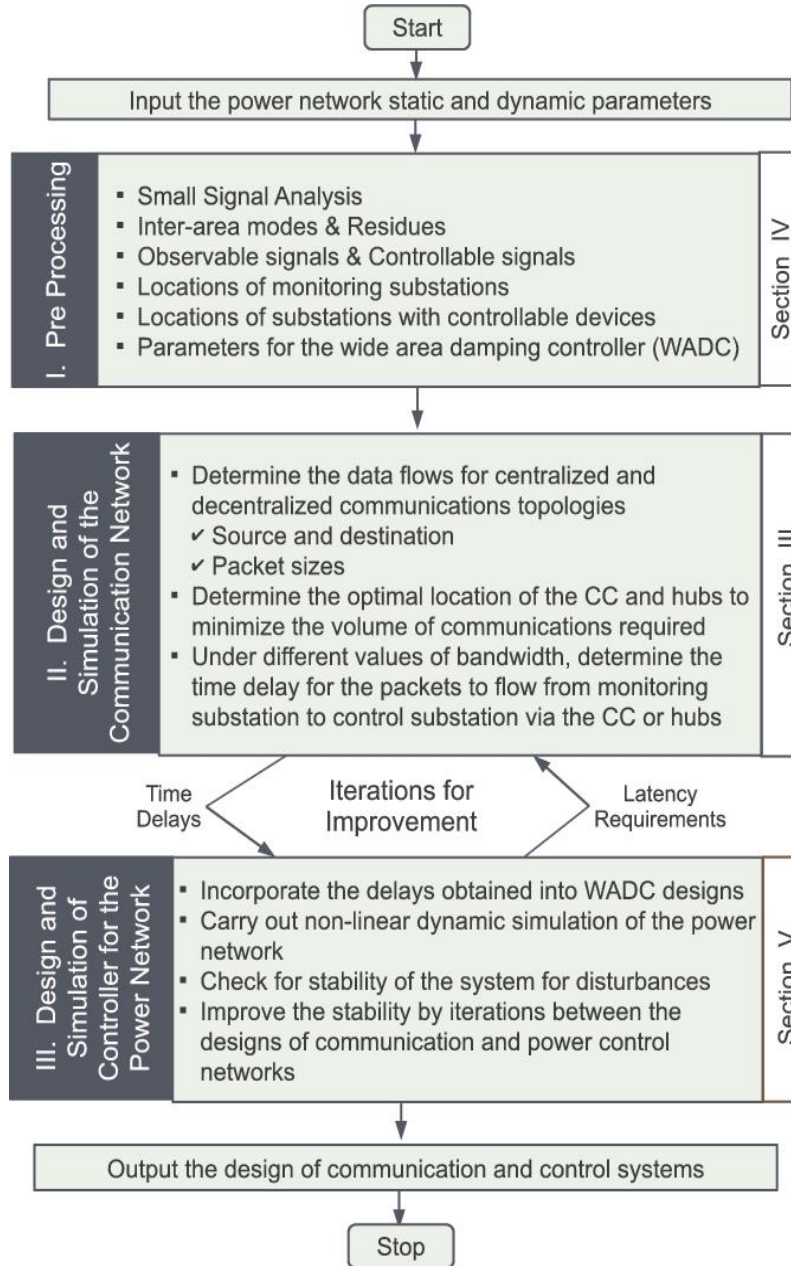
### 2.3.2 Communication Delay and the Number of Hops

To further establish the relationship between communication delay and the number of hops a simulation was carried out using the NS3 network simulator for a communication network topology on the IEEE 118 bus system. Each substation is modeled to send its data to the central control center which is connected to one of the substations. The communication links are modeled as two cases, with capacity of 100 Mbps and 20 Mbps. The results are shown in Figure 2.5. From this study, following observations can be made. Firstly, the communication delay varies linearly with number of hops, indicating that minimizing the hops results in minimizing the communication delay. Hence, the Mega-bit-hops (Mbh) is an appropriate metric to quantify the volume and delay for a given topology. Secondly, in the case where the link itself is overloaded due to lower capacity, an additional delay is experienced on account of buffering of the packets on the queues at the intermediate nodes. This buffering delay is not captured in the Mbh metric hence it should be ensured that the link bandwidths are sufficiently high.

The communication network can be considered as a connected graph where the nodes represent substations and links represent the communication lines. Each flow is between substation node and the chosen control center node or a hub node.

Since, IP based communication is used, the path taken by each flow is the shortest path in terms of links used from the sending node to the receiving node. The shortest path between two nodes for a given graph can be determined by Floyd-Warshall algorithm [55]. In networking, the hop count represents the total number of links a given packet passes through between the source and destination nodes. The more the number of hops the greater is the transmission delay incurred. In this work, the Floyd's algorithm has been implemented using Matlab code, and the Mbh count (Mbhc) is calculated by connecting a control center node to each substation node one at a time. Then we identify the optimal location of control center or the hubs based on the lowest value of

the Mbhc. Since, the Floyd's algorithm only calculates the shorted path, the communication network is also simulated in NS3 to confirm the path and to determine the time delays for each flow. It is verified, that the shortest path found by Floyd's algorithm is same as that of NS3 for all flows, as long as the communication network is not overloaded. Floyd's algorithm is convenient for solving multiple scenarios in a single program loop, whereas NS3 simulation also calculates the delays in communication.



**Figure 2.5: Relationship between communication delay and number of hops for various flows with different link capacities**

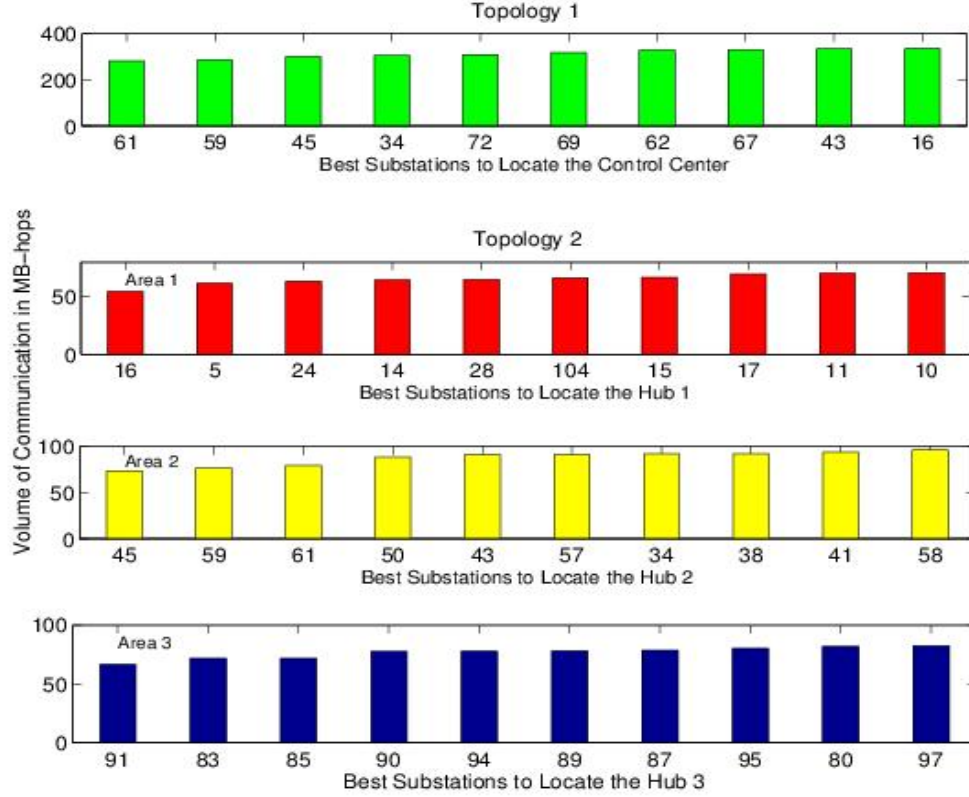
### 2.3.3 Design and Simulation of the Communication Network

Having defined a metric for volume of communications, we proceed to determine the optimal location for control center. For this the communication network topology is derived from the power network of IEEE 118 bus system. Two topologies are considered, namely, (i) Topology 1 (centralized), where there is a single central control center (CC) connected to one of the substations, (ii) Topology 2 (decentralized), where there are three hubs which communicate among themselves and an overall high level control center. Each of the three hubs are located respectively in each control area of the IEEE 118 bus system, as shown in Figure 2.7. In the process of deriving communication network from power network, the buses connected by transformers are merged in to a single communication node at the substation. As a result the substation numbers are different than the bus numbers of the IEEE 118 system. Assuming the location of control center at each of the substation, the Mbh is calculated using Floyd's algorithm. A sample of results showing the hops taken by the information for individual data flows are shown in Table 2.1.

The results are sorted in the increasing value of Mbh to determine the optimal location of control center. The best 10 locations along with the volume of communication, are shown in Figure 2.6. It can be observed, that for the case of topology 1, the best location for control center is at substation 61 resulting in total Mbh = 282.3. Whereas for the topology 2 the best location for 3 hubs are substations 16, 45, and 91 (which translates to bus numbers 17, 49 and 100) resulting in Mbh values of 54.5, 73.9, and 66.6, respectively, with the total adding up to 195. It should also be noted that, by moving from centralized topology to decentralized topology the volume of communication reduced significantly from 282.3Mbh to 195Mbh.

**Table 2.1: Reduction in Number of Hops**

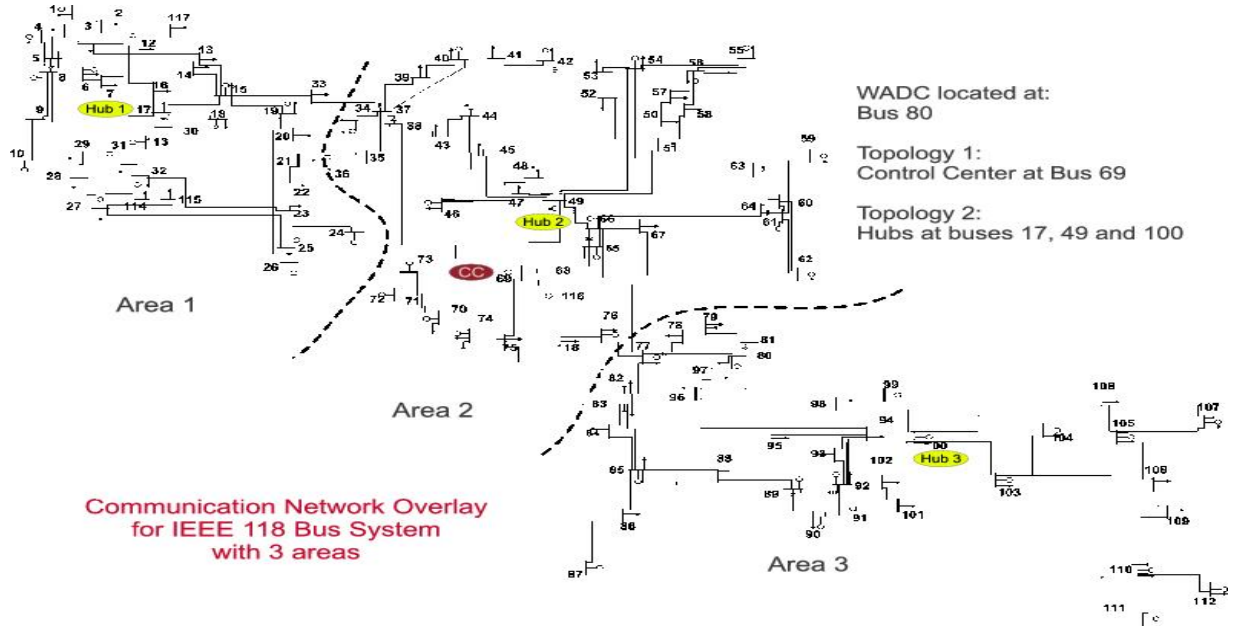
Flow ID	Topology 1 with 100 Mbps links (Centralized)				Topology 2 with 100 Mbps links (Distributed)			
	Path	Hops	Pkt'size	Bit-hops	Path	Hops	Pkt'size	Bit-hops
1	0,2,4,15,33,56,60,109	7	336	2352	0,2,4,15,27,109	5	336	1680
2	1,10,14,15,33,58,60,109	7	336	2352	1,10,14,15,27,109	5	336	1680
3	2,4,15,33,58,60,109	6	440	2640	2,4,15,27,109	4	440	1760
...	...	...	...	...	...	...	...	...
100	60,70,88,90,93,99,100,109	7	336	2352	90,93,99,100,111	4	336	1344
...	...	...	...	...	...	...	...	...
160	15,33,58,60,103,109	6	200	1000	90,93,100,111	3	200	600
...	Total Mega-bit-hops (Mbh) =			275.68	Total Mega-bit-hops (Mbh) =			199.97



**Figure 2.6: Optimal location of control center for Topology 1 and data routing hubs for Topology 2.**

#### 2.3.4 Calculation of Delay in Communication

Having determined the volume of communication with optimal locations for control centers and hubs, the next step is to determine the communication delay. For determining the delay, a simulation on NS3 has been carried out for both topologies 1 and 2. The results are shown in Tables 2.2 and 2.3. For the topology 1, two kinds of flows are involved, namely flow of measurement from SS9 to CC, and then from CC to SS80. It is configured in this way, because the small signal analysis (discussed later) of the system identified that the observable state is in substation 9 and controllable state is in substation 80 for damping a particular inter-area oscillatory mode. The obtained values of communication delay will be later used in the simulation of wide area damping controller in Section 2.4.



**Figure 2.7: Communication Network Overlay for IEEE 118 Bus System with 3 areas. The single line diagram is from [56].**

**Table 2.2: Calculation of Delay in Communication for Topology 1**

Bandwidth (Mbps)	Time delay (ms)			% Increase
	SS9-CC	CC-SS80	Total	
100	14.1	12.8	26.8	0.0
60	14.2	13.3	27.5	2.3
50	14.3	13.6	27.8	3.4
40	14.8	14.0	28.8	6.9
30	17.5	14.6	32.1	19.3
25	39.8	15.2	54.9	104.0
20	178.5	16.0	194.4	622.2

**Table 2.3: Calculation of Delay in Communication for Topology 2**

Bandwidth (Mbps)	Time delay (ms)				% Increase
	SS9- hub1	Hub1- hub3	Hub3- SS80	Total	
100	8.07	2.02	6.13	16.22	0.00
60	8.12	2.03	6.21	16.37	0.91
50	8.15	2.04	6.26	16.44	1.36
40	8.18	2.05	6.32	16.55	2.04
30	8.25	2.06	6.43	16.74	3.18
25	8.31	2.07	6.52	16.90	4.16
20	8.88	2.09	6.64	17.62	8.63
10	12.79	2.18	7.29	22.26	37.26
5	193.26	2.37	8.58	204.20	1158.9

### 2.3.5 Conclusion on Centralized and Distributed Topologies

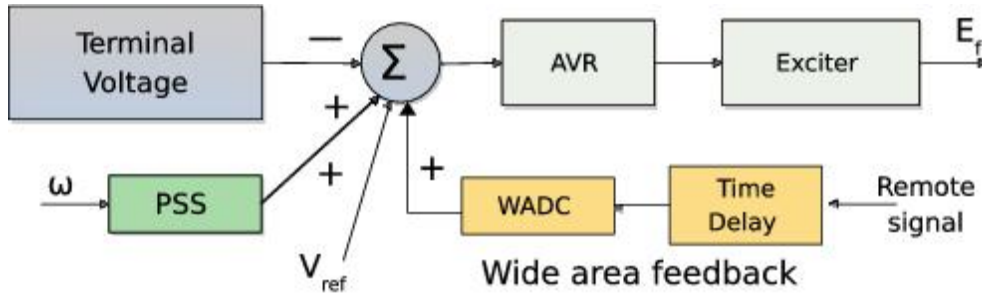
Based on the simulation results, it can be concluded that the decentralized topology results in lesser volume of communication (occurring in real-time) and also lesser amount of time delay. The reduction in volume of communication can be attributed to the decentralized localization of data in real-time. The reduction in delay can be attributed to the reduction in length of communication path, due direct peer-to-peer communication between neighboring hubs, instead of transfer of the monitoring data all the way to the central CC and subsequent relay of control commands back from central CC to the control substation. Based on the off-line linearized analysis of the power system dynamic model, the inter-area oscillatory modes of the system can be determined. With further study on observability and controllability analysis, the remote monitoring signal and location of wide area controller can be determined. These results can be used to configure the communication flows when simulating the communication network for determining the delays in the signal.

## 2.4 Pre-Processing: Design of Wide Area Damping Controller

In this section, the design and implementation of a wide area damping controller (WADC) incorporating the delay in the remote signal is presented using the IEEE 118 bus system. Apart from the monitoring applications, the real-time data received over the communication links can also be used to excite close-loop control applications such as WADC. The controllers using remote signals have certain constraints on latency. The objective of this study is to demonstrate the effect of communication time delay on the performance of closed-loop WADC.

### 2.4.1 System Description

In the IEEE 118 bus system considered, there are 118 buses, 186 branches, 3 control areas, and 19 generators. The generators belonging to area-1 are at bus numbers 10, 12, 25, 26, and 31, whereas for area-2 are at 46, 49, 54, 59, 61, 65, 66, and 69, and for area-3 are at 80, 87, 89, 100, 103, and 111. This allocation of generators into three control areas is taken from reference [56]. The system dynamics consists generator machine parameters, excitation systems, and power system stabilizers (PSS) for each of the 19 generators. The dynamic parameters as well as the parameters for local controllers (PSS) and the excitation systems are taken from the reference [57] and [58]. As described in the previous section, the WADC uses the measurement from SS9 and the controller is installed at SS80. In order to test the performance of PSS and WADC the damping coefficients of the machines themselves is set to zero [59]. Thus, all the damping has to be provided by the controllers. Conventional PSS using  $\Delta\omega$  as an input ( $\omega$  is the local generator speed) is installed on all plants to damp local modes. WADC is added in the form of an additional input to the exciter at generator SS80. The design of controller is shown in Figure 2.8. The input signal is the changing part of differential speed signal between the local generator and the remote generator  $\Delta(\omega_{89} - \omega_{10})$ . The transfer function of the controller is calculated based on residues compensation [60].



**Figure 2.8: Excitation system with WADC with remote signal**

The parameters are as follows:

$$H(s) = 26 \left( \frac{10s}{1+10s} \right) \left( \frac{1+0.503s}{1+0.0288s} \right) \quad (2)$$

For simulation of dynamic behavior, the disturbance scenario considered is a step change of automatic voltage regulator (AVR) set point by 0.05 pu at generator at SS10 for duration of 0.2 sec. It was observed in the simulations, that a higher value of controller gain would result in

faster damping, but also requires stricter limit on allowable time delay. Thus, there is a trade-off between damping ratio and time delay margin.

While the power network and controller parameters are discussed above, for the communication network parameters, two topologies are considered. For the topology 1 all the sub-stations send their data to CC and from there the specific pre-defined measurements needed by the controllers is forwarded from CC to the controller located at GSS. In the case of topology 2, each area has a hub which can be configured remotely with the measurement routing specifications. The description of communication topologies is summarized in Table 2.4.

**Table 2.4: Centralized and Decentralized Topology Communication**

Parameters	Topology 1: Centralized	Topology 2: Distributed		
		Area 1	Area 2	Area 3
Buses	118	37	45	36
Substations(SS)	109	34	40	35
Control Center	1	0	1	0
Data Routing Hubs	0	1	1	1
Generator SS	52	16	19	17
Communication Links	161	50	59	52

### 2.4.2 Small Signal Analysis

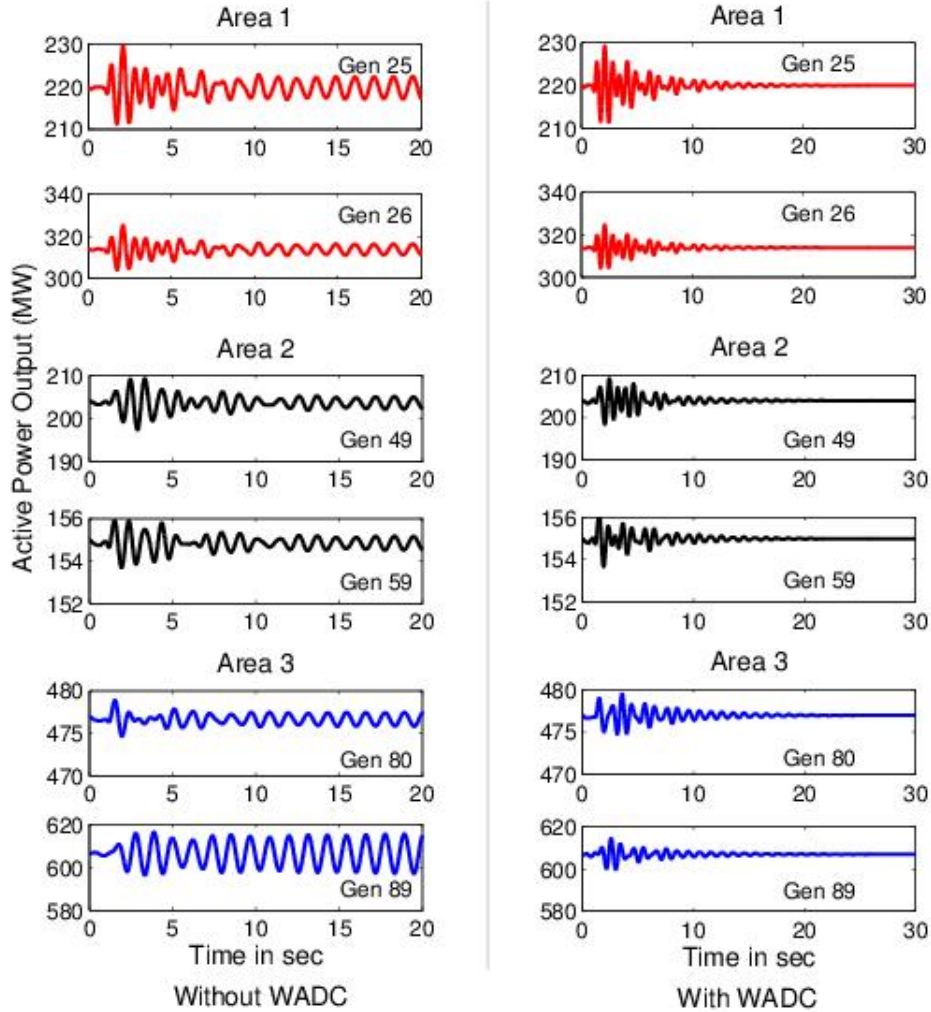
In order to determine the inter-area modes existing in the base case of the system, a linearized small signal analysis was carried including the excitation systems, PSS but not including the WADC. Out of the many eigenvalues of the system, a few have been identified as inter-area modes, which are summarized in Table 2.5. The last column of Table 2.5 shows the mode type, which indicates the number of generators exhibiting the presence of the mode out of the total number of generators.

**Table 2.5: Small Signal Analysis Results for IEEE 118 Bus System**

Eigen Value		Frequency (Hz)	Damping Ratio	Mode Type
Real	Imaginary			
-0.277	$\pm j7.26$	1.15	3.81%	13/19
-1.475	$\pm j14.67$	2.33	10%	7/19
-1.113	$\pm j11.71$	1.86	9.47%	7/19
-0.2.2	$\pm j5.86$	0.933	3.45%	11/19

The system is then simulated without the WADC. Following the disturbance, it was observed that, while the local modes are damped by around 5 seconds, the inter-area modes persist beyond 20 sec. The first column of plots in Figure 2.9 shows the active power generation of two generators from each area. It can be observed that the generators in area 3 are oscillating against

the generators in areas-1 and 2. Now, the WADC is introduced with time delay of the remote signal set to zero. The response of generators with the WADC is shown in the second column of Figure 2.9.



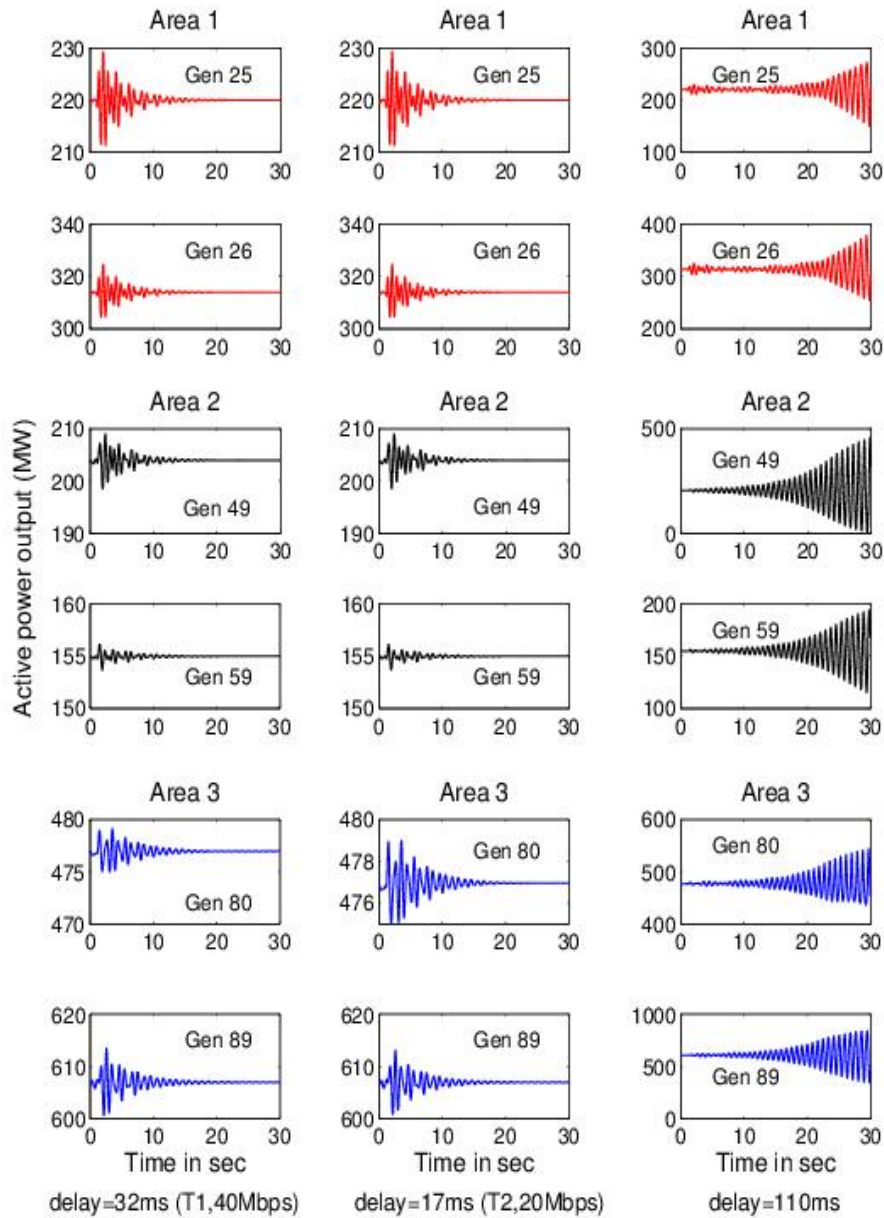
**Figure 2.9: Dynamic Response of Generators**

## 2.5 Simulation of Controller for the Power Network

The non-linear time domain simulation of the power network is carried out for both topology 1 and topology 2 incorporating the time delays obtained respectively. For the simulation of topology 1, a case with 30 Mbps bandwidth having 32.1 msec delay is chosen. And for topology 2, a case with 20 Mbps bandwidth having 17.6 msec delay is chosen.

These values are chosen from the results shown in Tables 2.2 and 2.3. The design of the controller has been described earlier in Section 2.4.

Figure 2.10 presents the results of non-linear time domain simulation of power network. The plots in the first column shows the dynamic response of two generators selected from each area for topology 1. Similarly, the plots in the second column represent the dynamic response for topology 2. To illustrate the effect of time delay on the controller performance the time delay has been successively increased, and it was observed that the WADC results in unstable system for a time delay more than 110 msec, as shown in the plots of the third column of the Figure 2.10. Since for most cases the time delay of our communication network for both topologies is well less than the 110 msec latency requirement of the controller, it can be concluded that the WADC can successfully provide the required damping of the inter-area oscillations.



**Figure 2.10: Dynamic Response of Generators**

## 2.6 Conclusion

This research presented a process for combined design and simulation of communication and control systems for the IEEE 118 bus system. In the pre-processing step, the wide area damping controller (WADC) parameters have been determined which also models the time delay in communication. Then a procedure is proposed for optimal location of control center and data routing hubs to minimize the volume of communication. The time delays and bandwidths for centralized and decentralized topologies are determined through NS3 simulations. These time delays are then incorporated into the WADC and a non-linear time domain simulation of the power network is carried out. Based on simulation results of two communication topologies, it has been demonstrated that the distributed architecture has more advantages than the centralized one. Decentralized topology can achieve shorter time delays even with lower network bandwidth, there by reliable and suitable choice for wide-area damping controller spanning multiple control areas.

## 3 Unified Geo-Spatial Model

### 3.1 Toward Operations and Planning Modeling Unification

As we have described a number of algorithms are used in planning and operations stages of power system management in order to ensure that the system can handle various events. One of the most important tasks in power system long-term planning and operations planning is Contingency Analysis, which consists of automated what-if-type simulation of large numbers of possible system events under a variety of future scenarios. Contingency analysis contributes to ensure that the operation of the system is secure. Contingency analysis uses planning cases, which are bus-branch level models, representing a simplified version of the physical system [3]. The physical system is represented in detailed by a node-breaker model.

The application corresponding to Contingency Analysis, but in real-time is Security Assessment. This application performs essentially the same function as contingency analysis with the following differences:

- a. It operates integrated into a real-time loop as part of the Energy Management System (EMS),
- b. It provides contingency screening functions which enable processing the list of contingencies in the shortest time, suitable for a real-time loop, and
- c. It acts on a full-topology operations model of the grid, i.e., a model that includes detailed switching device information at the node-breaker level.

The bus-branch (planning) and node-breaker (operations) models used pervasively in the industry today are largely incompatible. The network models used by the power industry for planning do not contain information about the switching devices or substation topology configurations. They are simplified, bus-branch models where the normal operation statuses of breaker and disconnects has been assumed [4], and where each group of nodes connected by normally closed switching devices has been merged into a single bus. Consequently, planning models cannot be used to fully analyze contingencies that involve arbitrary switching devices actions [5]. In particular, it has been recognized that many plausible contingencies create bus mergers or bus splits that result in network topologies not realizable with a given bus-branch model [6]. Contingencies involving switching devices studied in the planning environment as part of expansion studies or in designing special procedures for substation maintenance. They are usually not studied routinely as part of day-to-day operations planning despite the fact that such contingencies can actually occur and that their effect on the system can be significant. This limitation of bus-branch models continues to expose the operation of the system to possible unstudied events.

Beyond models, at the application level, the existing EMS security assessment and planning contingency analysis applications are incompatible too, i.e., despite the fact that their core solution algorithm is essentially the same (an AC power flow), neither can read the other's model or solve the other's computational problem [7, 8].

It is clear that in order to ensure system security, all plausible events must be studied in the planning environment, including contingencies that involve bus mergers and bus splits. Comprehensive study of such contingencies using existing planning tools involve developing the equivalent of a large number of bus-branch models to realize the post contingency topology through contingency-specific scripting, which is cumbersome and prohibitive for a large number of contingencies.

Switching-level contingency analysis in planning could be accomplished in an efficient and comprehensive manner if:

- a. The network model used in planning had the necessary detail at the physical level, i.e., it is based on a full-topology, node-breaker model, and
- b. Existing software would be enhanced so that switching contingencies are modeled seamlessly and post-contingency topologies are realized automatically.

The work developed in reference [3] has resulted in a novel framework that realizes *Unified Network Applications* and resulting tools that provide seamless exchange of models and network applications at the power flow level, addressing literal a) above. The present work on the other hand, addresses the issue described in literal b): the enhancements to software for automatic study of arbitrarily number and arbitrary topological complexity of contingencies that include switching device actions.

Section 3.2 describes contingencies that are non-realizable with bus-branch models and the challenges that arise in their modeling and simulation. Section 3.3 discusses the interoperability implications of the proposed method. Section 3.4 describes efficient methods for handling switching device contingencies. Section 3.5 presents computational time results and analysis using a large-case realistic Independent System Operator (ISO) model.

### **3.2 Issues with Model Unification**

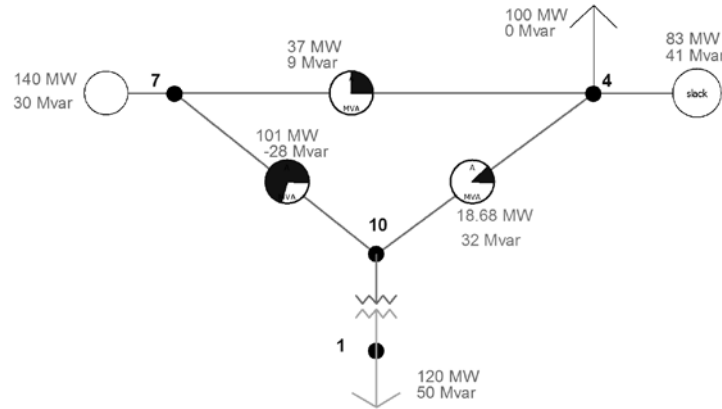
A bus in bus-branch models typically used in planning is in actuality a set of (three-phase) bus-bars and junctions connected through (normally) closed switching devices (breakers and disconnects). A bus-branch model provides an accurate representation of the electric network, except when the breakers or disconnects have statuses different from their assumed normal operation conditions. In many cases, the topology that results due to changes in breaker statuses can be represented by the bus-branch model. For instance, the outage of a transmission line may involve opening the normal and bypass circuit breakers at both ends of the transmission line. In

the bus-branch model this is realized using an “open transmission line” contingency action. This is an example of a contingency that is *realizable with a bus-branch model*.

To illustrate a simple case where contingencies cannot be realized with the bus-branch model alone, consider Figure 3.1, which shows the node-breaker model of a very small power system. For simplicity we present the example of a network considering circuit breakers only (not disconnects). This power system contains 11 nodes. The system contains 8 circuit breakers showing their normal operating statuses.

**Figure 3.1: 11-node node-breaker model showing bus groupings that correspond to buses in the bus-branch model**

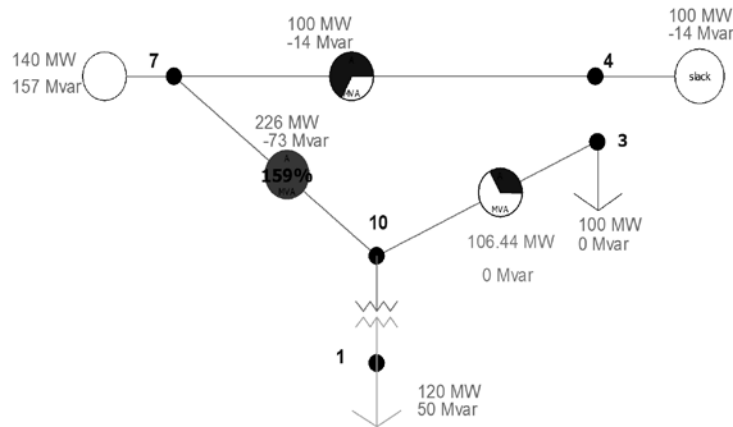
buses. In this example there is a relation between the bus numbers that are preserved in the bus-branch model and the node numbers in the node-breaker model: the formed bus retains the number of one of the nodes in the corresponding node group. This relation is seldom the case in realistic models used in operations and planning, where the node and bus numberings are not related. Bus numbers are the result of historical assignment of number ranges for various reliability regions in a system.



**Figure 3.2: Four-bus bus-branch model corresponding to the 11-node node-breaker system and its breaker statuses**

All the planning studies of a (physical) system as the one shown in Figure 3.1, would take place using only the bus-branch model shown in Figure 3.2. Planners do not have a case with all the information about the detailed topology or how each bus was originated [9, 10].

Contingencies such as single or double line outages in the bus-branch model are realizable and can be simulated using the bus-branch model, by “disconnecting” the transmission lines. On the other hand, consider a contingency that involves simultaneously opening breakers 3-6 and 4-5 in Figure 3.1. This will result in bus 4 being split, a contingency that cannot be realized and hence cannot be studied using the bus-branch model shown in Figure 3.2. The bus-branch model would have to be modified to obtain a new bus-branch model with the topology shown in Figure 3.3, where a new bus (bus 3) has been created, the terminal 4 of transmission line 4-10 has been moved to bus 3, and the load at bus 4 has been moved to bus 3 as well.



**Figure 3.3: New bus-branch model needed to represent and study a bus split contingency**

In order to realize this new system with existing software, a script such as the following would be required.

```

Begin
  Create New Bus (3)
  Move Line (Line 4-10) to be Line (3-10)
  Move Load (4) to be Load (3)
  Solve Power Flow
  Tabulate Results
  Move Load (3) to be Load (4)
  Move Line (Line 3-10) to be Line (4-10)
  Delete Bus (3)
end

```

Similar scripts would need to be developed for the bus-branch model in Figure 3.2, in order to model bus split or merger contingencies. It must be noted that the step of determining whether the bus is split and how it is split used information that is not contained in the bus-branch model and required visual analysis done by a human. In realistic cases this is not trivial. By exploring realistic ISO node-breaker (operations) and bus-branch (planning) models, we have identified that the relation between the number of nodes and the number of buses is about 8:1 in models that have breakers only, and it is about 20:1 in models that have both breakers and disconnects. Automating this step leading to the development of scripts for a large number of complex contingencies in large-scale systems is a highly involved task. Developing such scripts is not only prone to error, but it causes significant overhead in the computation. In addition, such method has the following fundamental drawbacks:

- a) The line that needs to be moved (Line 4-10 in the previous example), has a different name in the post-contingency state: it becomes Line 3-10. The software (or the analyst) must keep track of the new name to correctly tabulate the contingency analysis results, such as line thermal limit violations.

- b) The moved load has a new name: Load 3. This will also need to be tracked to provide consistent result tabulation [11, 12].
- c) The newly created bus, Bus 3, which does not exist in the pre-contingency system, may present contingency violations, such as low voltage magnitude. Because bus 4 (the bus in the original model) has a different post-contingency voltage and may not exhibit a violation, this creates the impossibility of tracking such low voltages and associating them with a single bus in the result tabulation.

Clearly, the approach of developing scripts for bus split or merger contingencies is highly involved and results are difficult to tabulate. The determination of how many buses result from each combination of switching actions, and how the resulting devices need to be connected needs to be automated.

### 3.3 Toward Temporal Interoperability

A key objective of grid application interoperability is the seamless exchange of data and system models enterprise-wide among all business processes and analytical stages of real-time operations and off-line activities. Because modern power systems are operated more stressed and closed to their security limits, and because of the need to analyze more complex operation with growing renewable energy, it is important to develop methods that enable comparing system studies established across all stages of power systems. The existing incompatibility of models, formats, and network applications used in planning and operations represents a barrier to achieving interoperability and higher security [13].

This research demonstrates a method that enables simulating all the EMS contingencies, including bus merges or splits in the planning environment, something currently cumbersome with existing planning tools, which are limited to a few small impedance branches or switching devices. Furthermore, the proposed method allows seamless processing, without requiring any model conversion or complex scripting. The implications of the proposed method are the following:

- a. It enables direct and comprehensive comparison of planning and operations studies for massive number of contingencies.
- b. It allows synchronizing and making compatible real-time and off-line models. By being able to perform detail comparison of real-time and off-line simulations, model, network parameters, or input data errors can be identified and corrected. With current computational power, storing switching devices in the model is no longer a problem, and we note that the number of generators, loads and shunts is the same in node-breaker and bus-branch models.
- c. It enables quick analysis of potential events in operations planning, directly benefiting system security.

## 3.4 Modeling Switching Contingencies

### 3.4.1 Device Terminal Pointer Assignment

Existing planning tools are able to model a few switching devices by treating them as (very) low impedance branches. However, as the number of switching devices increases, the power flow Jacobian becomes ill-conditioned and numerical instability problems arise. Reference [14] describes the low impedance branch numerical instability problem. Several efforts by researchers have taken place towards modeling switching devices in network applications [15, 16]. In particular, topology error detection in state estimation benefits from including breaker statuses as part of the estimation, the Generalized State Estimation (GSE) [17, 18].

After the unified network application approach proposed in [3], a single, node-breaker model can be utilized for all network applications. In this approach a consolidated representation of the node-breaker model, electrically equivalent to the corresponding bus-branch model, can be obtained dynamically without altering the node-breaker model and without requiring creating a separate bus-branch model. This is achieved by assigning pointers associated with nodes that are connection terminals of power system devices. The fundamental idea of the method is that any topological configuration due to changes in switching device status (either during normal operation or to simulate contingencies) can be efficiently realized by assigning node pointers.

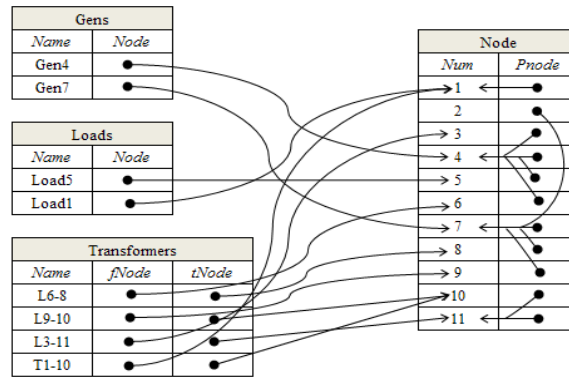
We now describe in detail the solution of a power flow on a node-breaker model using the mentioned pointer assignment method. We use as an example the node-breaker system shown in Figure 3.1. The algorithm starts by identifying the groups of nodes connected by closed switching devices. For instance, if the breaker statuses shown in Figure 3.1 are assumed, then each group of nodes {1}, {3,4,5,6}, {10,11} and {2,7,8,9} correspond to nodes that are the same electric point (have the same voltage phasor). Once the groups are identified, a node from each group, the primary node or pnode, is selected. Although any node in the group can be chosen as pnode, rules are used to determine what nodes are selected. A simple rule is to select the node with the smallest number, but a more convenient implementation uses a priority rule where voltage regulated buses and device terminals have priority over regular nodes. In the example, nodes 1, 4, 7, and 10 are selected as pnodes of each one of the node groups.

The next step is to have each node in the system contain a pointer that points to the selected pnode for the node group. To illustrate the implementation, we use object oriented programming: a node in the node-breaker model is a TNode class. This class has a pnode property that is of the same TNode type:

```
class TNode
  properties:
    num: integer
    pnode: TNode
```

Figure 3.4 shows the resulting pointer assignment for the node-breaker system of Figure 3.1. Let us consider the terminals of each power device (transmission line, transformer, load, generator, etc.) physically correspond to nodes. For instance the terminal nodes of Line 3-11 are nodes 3 and 11. If instead of the terminal nodes the pnodes of the terminal nodes (node.pnode) are used, we can realize the devices that would be present in the bus-branch model. For instance, in Line 3-11, Node 3 has node 4 as pnode and node 11 has node 10 as pnode. Hence Line 3-11 appears as connected between nodes 4 and 10, resulting in a transmission line equivalent to Line 4-10 in the bus-branch model shown in Figure 3.2.

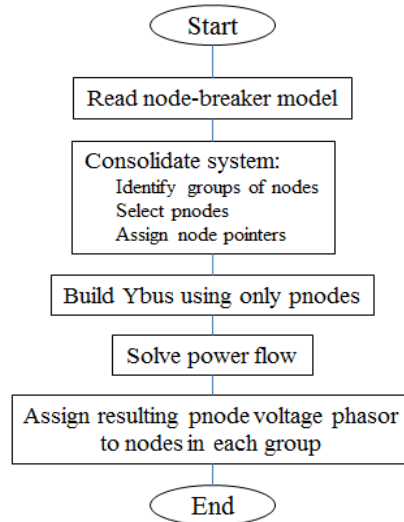
If not the terminal nodes, but the terminal node pnodes are used for all the devices in the system, we would realize the devices connected as in Figure 3.2, the bus-branch model. Once the pnodes are used, the non-pnode nodes have no devices attached to them and they and the circuit breakers in the model become irrelevant. A consolidated representation of the node-breaker model electrically equivalent to the bus-branch model has been obtained without requiring generating a bus-branch model.



**Figure 3.4: Device pointer assignment using a unified application framework: normal operation condition corresponding to Figures 3.1 and 3.2**

We note that the pnode property in the Node Table is a pointer to a record in the same Node Table. The proposed method uses a single class, TNode, without requiring a bus (Tbus) class. Traditionally, separate data structures have been used to store objects of the node-breaker and bus-branch models. For instance, the Common Information Model (CIM) [14,15] considers a Connectivity Node class to represent nodes in the node-breaker model and a Topology Node class to represent buses. Because our method uses a single class, no model conversion is required.

Figure 3.5 illustrates the process to obtain a power flow solution on a node-breaker model, by using device terminal pointer assignment.



**Figure 3.5: Power flow on a node-breaker model**

The Consolidate System step in Figure 3.5 is achieved by the following algorithm, which for illustration purposes assumes the rule of the pnode being the node with the smallest node number:

```

Procedure Consolidate System
Begin
  changed = true
  while changed do begin
    changed = false
    pLine = FirstLine
    while pLine <> nil do begin
      if is Switch and Status = closed
      then begin
        if fnode.pNode.num < node.pNode.num
        then begin
          fnode.pNode = tnode.pNode
          changed = true
        elseif fnode.pNode.num > node.pNode.num
        then begin
          tnode.pNode = fnode.pNode
          changed = true;
        end
      end
      pLine = pLine.next
    end
  end
end

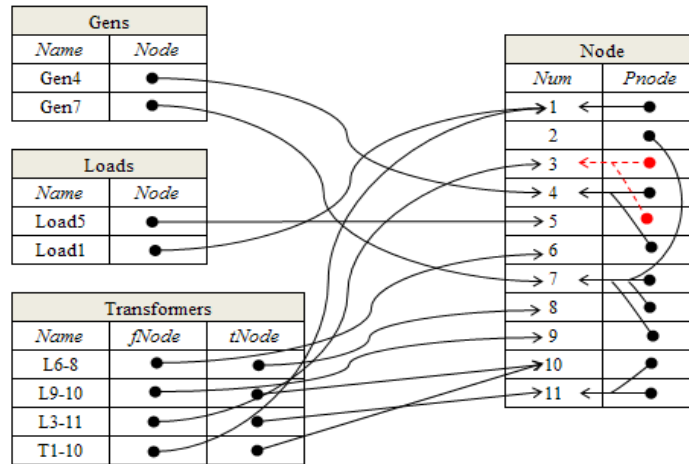
```

This algorithm reportedly converges in about 4-5 iterations for realistic systems with more than 10,000 nodes, taking less than 0.1 sec with a modern computer. The corresponding deconsolidation or expansion algorithm, which resets the pnodes is trivial and consist of setting all the pNodes to themselves:  $\text{node}(i).\text{pnode} = \text{node}(i)$ .

As switching devices change their status, preserving the correct system topology is hence a matter of maintaining the correct pointer assignment. This will be the basic of the method to model contingencies of arbitrary complexity.

### 3.4.2 Contingency Modeling

During contingency analysis, the node groups connected through closed breakers that were originally determined must be altered, possibly creating new groups (bus split) or merging groups (bus merging). The corresponding pnodes involved are modified accordingly. For instance, let us consider the contingency we described in Section II, due to simultaneously opening circuit breakers 3-6 and 4-5 in the system in Figure 3.1. This contingency is not realizable with the bus-branch model. The contingency actions will result in a post-contingency topology with two bus groups of nodes {4,6} and {3,5}, which become buses 4 and 3, as shown in Figure 3.3. Node 4 remains being a pnode and node 3 becomes a new pnode. The device pointer assignment required to realize the post-contingency condition is shown in Figure 3.6.



**Figure 3.6: Device pointer assignment for post-contingency state (breakers 3-6 and 4-5 open), corresponding to Figure 2.3**

By comparing Figures 3.4 and 3.6 it is observed that:

- No changes in generator, loads, lines or other power devices names or connection terminals are needed.
- The only changes needed are associated with changes to the pnode pointers. In this example, node 3 pnode pointer was changed from node 4 to node 3 (itself), and the pnode of node 5 was changed from 4 to 3.
- Because of a), tabulating the violations in any device across contingencies is possible, without requiring a special tracking procedure.

### 3.4.3 Incremental Subnet Processing

In this section we describe a generic contingency analysis algorithm, which allows us to automate the process leading to the pointer assignment for a large number of contingencies. Such algorithm must identify what existing node groups (buses) are affected by a contingency, and how the pnode pointers must be changed. Because switching actions may result in bus mergers it is not enough to look “inside” the node groups.

We start by defining as a *subnet* the groups of nodes that are connected thorough breakers without considering their status. The groups of nodes {1}, {2,7,8,9},{3,4,5,6}, and {10,11} are also subnets. When all the breakers are closed the subnets are equal to the resulting groups of nodes that form buses in a bus-branch model. However, the set of nodes that belong to a subnet don’t change when the statuses of switching devices change. Subnets are static and represent the maximum size of the node groups. In addition, it is noted that subnets are bounded by the terminals of transmission lines and transformers. Thus subnets provide an invariant region to determining “what has changed” in the system as a result of a contingency. In realistic systems, the number of subnets is about 90 to 95% of the number of buses, i.e., there are a small percentage of subnets that contain two or more node groups identified as buses at a given time.

Subnets can be determined in a straightforward manner using the same Consolidate System algorithm used to determine node groups, except that the line of code where the status of the switching devices is checked

```
if is Switch and Status = closed
```

must be replaced by

```
if is Switch
```

Each node and switching device can be assigned a reference to the container subnet. Because switching contingencies involve breakers status changes as contingency actions, determining the affected subnets due to a contingency becomes a trivial process. Typical contingencies would affect 1 to 4 subnets, while more involved contingencies (such as double or common-mode) can easily affect 8 or more subnets. In any case, the number of subnets to be re-processed during a contingency is a very small fraction compared to the number of subnets or buses in the system.

Contingency analysis requires sequential processing of a large number of plausible contingencies. The system state must be returned to the pre-contingency state after each contingency has been solved. Instead of resolving each time the pre-contingency pointer assignment for the entire system, it is easier to save the pre-contingency pnode pointer assignment. Thus the TNode class would be extended with a pre-contingency pnode property (preCTGpnode) also of type TNode. It is also required that each subnet contains a list of breakers and a list of nodes inside of it. With these pre-requisites, we are ready to propose the following

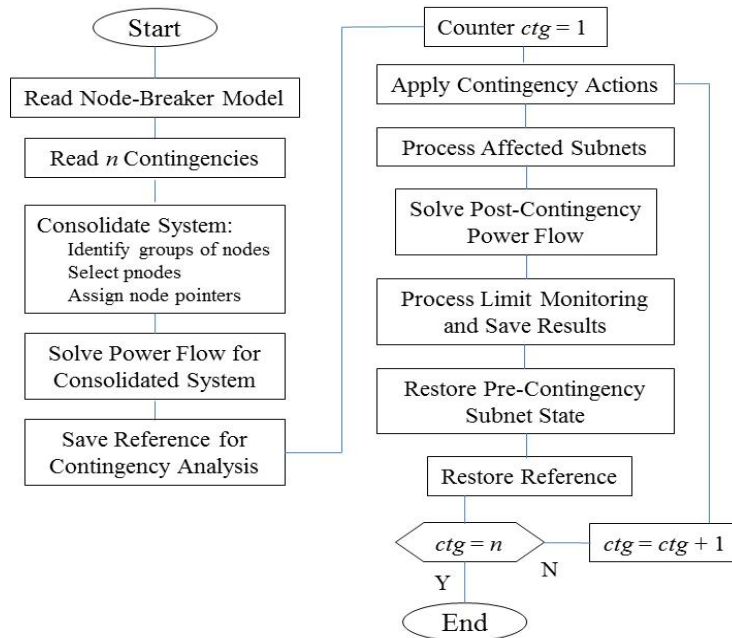
algorithm to incrementally solve contingencies on a node-breaker model that includes arbitrary complexity of the substation topology and arbitrary number of switching contingency actions:

```

Procedure IncrementallySolveContingency
Begin
  For each ContingencyAction do begin
    Apply ContingencyAction
    If ContingencyAction.Type = Switch then
      ContingencyAction.Switch.
      .Subnet.Affected = True
    end
  End
  For each affected Subnet do begin
    Subnet.StorePreContingencyState
    Subnet.Expand
    Subnet.Consolidate
  End
  SolvePowerFlow
  StoreContingencyResults
  For each affected Subnet do begin
    Subnet.RestorePreContingencyState
    Subnet.Affected = False;
  End
End;

```

This process is extremely fast for a subnet, because Subnet.Expand and Subnet.Consolidate consists only of the switching devices that belong to the subnet, requiring a few milliseconds in a large-scale system. Figure 3.7 illustrates the solution process for an arbitrary number of contingencies.



**Figure 3.7: Solution process for a list of contingencies using a node-breaker model. Contingencies can include arbitrary switching device configurations and substation topologies.**

### 3.5 Case Example: Unified Contingency Analysis Interoperability

#### 3.5.1 System Properties

In order to test the performance of planning contingency analysis with switching devices, PowerWorld Simulator was utilized. This tool implements the unified network application framework. In this section, we demonstrate contingency analysis using a large-scale ISO node-breaker model. The relevant characteristics of the model are listed in Table 3.1. The system includes modeling of breakers only and corresponds to the exact network model used by the control center EMS system.

**Table 3.1: Properties of the Sample Case**

Property	Large ISO Model
Number of nodes	10,436
Number of branches	11,668
Number of breakers	9,074
Equivalent consolidated buses	2,054

#### 3.5.2 Incremental Processing vs. Low-X Branch Solution

In this section we compare the solution times for the consolidation process with the corresponding solution times on the node-breaker model without consolidation. This requires modeling breakers as low-impedance branches using a 0.0001 pu default reactance value [11,12].

Critical single transmission line outages were simulated and the computational time needed to solve the entire contingency set was recorded. Contingency analysis was performed using two different solution methods: full AC and DC power flow. The computational time results of the simulation are presented in Table 3.2.

**Table 3.2: Solution Times for the Sample Case**

Property	Incremental Subnet Processing	Low Impedance Branches
Num. Contingencies	793	793
Full AC (mm:ss)	00:46	2:16
DC (mm:ss)	00:08	1:37

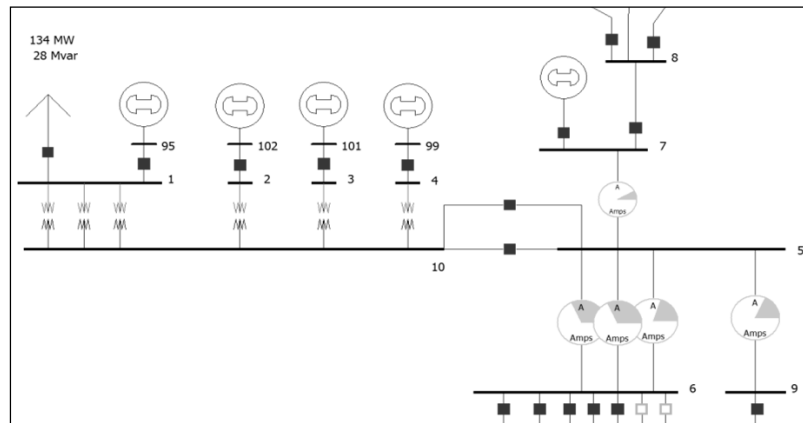
All contingencies were solvable. Both methods presented virtually identical numerical contingency results. For both AC and DC power flow methods, using incremental subnet processing significantly reduced the contingency solution time.

#### 3.5.3 Incremental Subnet Processing vs. Scripting

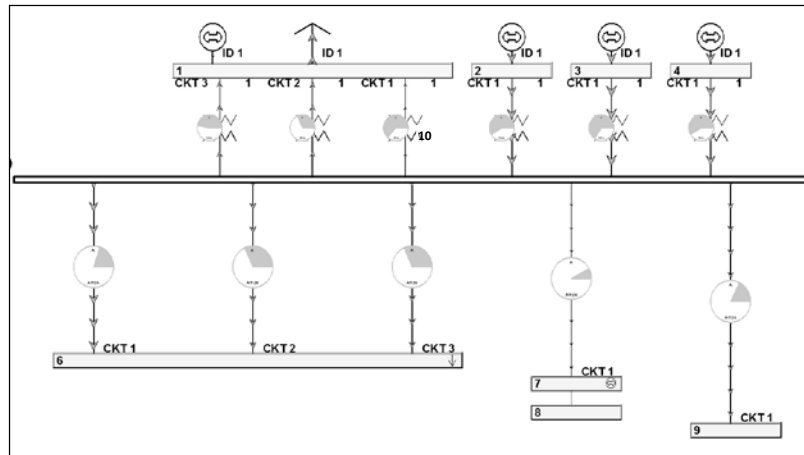
As discussed in Section II, using existing software, a planning model would have to be modified in order to realize contingencies that include bus merges or splits. A quantitative comparison can

be developed between the solution times needed to solve such contingencies using a bus-branch model and scripting versus incremental subnet processing.

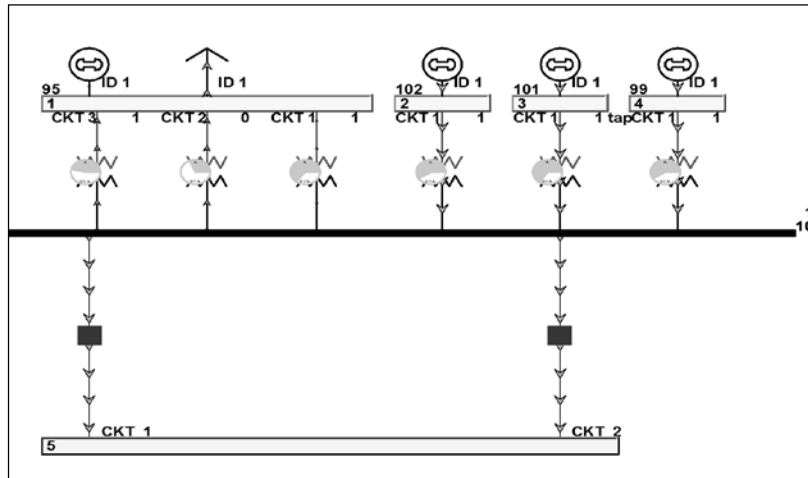
Consider a region of the ISO full-topology described in Section IV.A, as shown in Figure 3.8. Given the status of the breakers in this system, buses 5 and 10 consolidate into bus 10, and buses 7 and 8 consolidate into bus 7, resulting in Figure 3.9. This is the corresponding bus-branch model for the section of the full-topology model. Consider now a contingency which causes the simultaneous opening of the two breakers connecting buses 10 and 5. This contingency is clearly not realizable using the existing bus-branch model of Figure 3.9: the contingency results in the current bus 10 being split. The node-breaker model was manually explored to determine a) which buses would split or merge and b) which devices needed to be relocated due to the bus split or merge. The existing bus-branch model was modified to the one shown in Figure 3.10 where a new bus (bus 5) has been created.



**Figure 3.8: Detailed view of a section of the ISO node-breaker model**



**Figure 3.9: Resulting bus-branch model**



**Figure 3.10: New bus-branch model after the switching contingency**

The terminals buses of five transmission lines: 10-6 circuits 1, 2, and 3; line 10-7 and line 10-9 were moved to the newly created bus. In order to make these changes to the bus-branch model and to solve the contingency, the following script was developed.

```
Start
  Create New Bus (5)
  Move Line 10-6-CKT1 to Line 5-6-CKT1
  Move Line 10-6-CKT2 to Line 5-6-CKT2
  Move Line 10-6-CKT2 to Line 5-6-CKT2
  Move Line 10-7 to Line 5-7
  Move Line 10-9 to Line 5-9
  Solve Contingency
  Save Results
  Move Line 5-9 to Line 10-9
  Move Line 5-7 to Line 10-7
  Move Line 5-6-CKT1 to Line 10-6-CKT1
  Move Line 5-6-CKT2 to Line 10-6-CKT2
  Move Line 5-6-CKT3 to Line 10-6-CKT3S
  Delete Bus 5
  Reset Contingency Reference
End
```

The time required to solve this contingency using scripting was 10.5 times slower than solving the contingency using incremental subnet processing. The same process was repeated for ten different bus merges and split contingencies using the large-scale ISO model. The results varied from about 8.4 times to 17.2 times slower using scripting. Although the commercial software scripting is highly optimized for speed, the fact that devices need to be created and moved caused significant computational overhead. It must be mentioned that the engineering effort required to manually understand the topologically effects of complex contingencies and the time required to write the scripts was not considered as part of this simple benchmark.

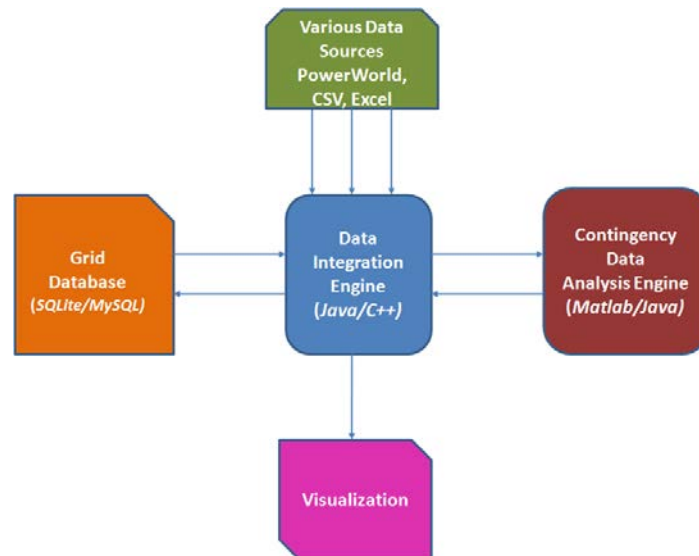
Comprehensive development of numerous bus split or merging contingencies in realistic case is extremely time consuming and cumbersome. The proposed method entirely avoids the need of such analysis and automates the process reducing the total time and effort required by several orders of magnitude. The detailed analysis of contingencies that involve detailed substation topology for applications such as substation maintenance constitutes a project which may take engineers weeks to complete. The proposed method can provide the same results in the order of seconds.

## 4 Seamless Computational Framework

### 4.1 Database Structures for Temporal Data Handling

Once a unified model has been established, it is necessary to develop the data structures and repositories needed to support data access by the computational applications. The data and computational framework are the second element of the seamless energy management system prototype.

The information architecture for the prototype uses Java and a SQL-like database system, SQLite, to organize the processing of initial-state power system data, interfacing with the contingency analysis algorithm in MATLAB, and organizing the output of the analysis back into the database system. This architecture is shown in Figure 4.1.



**Figure 4.1: Proposed system architecture of the prototype**

In developing the Grid Database, the following work has been done:

- A database was designed to meet the temporal contingency analysis data store needs and visualization.
- A parser was created to extract data from PowerWorld database in a format that is easily plugged into the grid database.
- The designed database schema was implemented with Java/JDBC and SQLite and was populated to do initial testing.

The initial database schema is quite simple in nature, consisting of only three tables. From its current design, it is trivial to retrieve the state of the buses, transmission lines, generators, and other system elements at any given time step, under any control action, and according to any specific scenario (e.g., contingency). This allows visualizing the components of the system in their respective states. Several revisions led to the database configuration in order to meet the contingency analysis information requirements. Sample snapshots of the current database are presented in Figure 4.2.

The important attributes added to the regular power system schema (Bus and Branch Tables) are **TIME\_STAMP** that records the UTC time of when the event happened and **STATE\_ID** to uniquely identify each node of the temporal contingency analysis tree. To traverse through the tree and to identify what contingency caused the shift from one state of the system to another state, the third table (State Table) was introduced. This table aids in extracting various statistics of the system state after performing contingency analysis on it. It can be used to:

1. Find effects of all similar kind of control/contingency actions
2. Make a temporal traversal on the tree to trace all possible path of actions
3. Visualize the system state (all possible attributes) on a particular time, etc.

#	TIME_STAMP	STATE_ID	BUS_NUM_1	BUS_NUM_2	LINE_CIRCUIT	LINE_STATUS	LINE_R	LINE_X	LINE_C	LINE_AMVA
1	1383834263576	0	1	2	1	Closed	0.02	0.06	0.06	150.0
2	1383834263576	0	1	3	1	Closed	0.08	0.24	0.05	65.0
3	1383834263576	0	2	3	1	Closed	0.06	0.18	0.04	80.0

**Branch Table**

#	TIME_STAMP	STATE_ID	BUS_NUM	BUS_CAT	GEN_ID	GEN_MW	LOAD_ID	LOAD_MW
1	1383834263576	0	1	null	1	101.8534	null	null
2	1383834263576	0	2	null	1	170.0779	1	40.0
3	1383834263576	0	3	null	null	null	1	110.0

**Bus Table**

#	TIME_STAMP	STATE_ID	PARENT_STATE_ID	CONTINGENCY_ACTION	CONTROL_ACTION
1	1383834263576	0	-1	null	null

**State Table**

**Figure 4.2: Grid database structure**

The system implemented thus far is fully Java-based. We hope to create the control engine in Java, to which different components would be plugged in. For the amount of expected to be produced by the contingency analysis component, a traditional RDBMS system would be able to easily handle even the most complex query. We used SQLite, a very light weight RDBMS

system and communicated with it via JDBC from the central engine. The central engine first extracts relevant contingency analysis information from a PowerWorld case which is then loaded into the database system as well as passed to the MATLAB analysis engine. Once the analysis is completed, the output is loaded back into the database system and retrieved from there later to supply to visualization engine.

## **4.2 Development of Smart Contingency Analysis Algorithm**

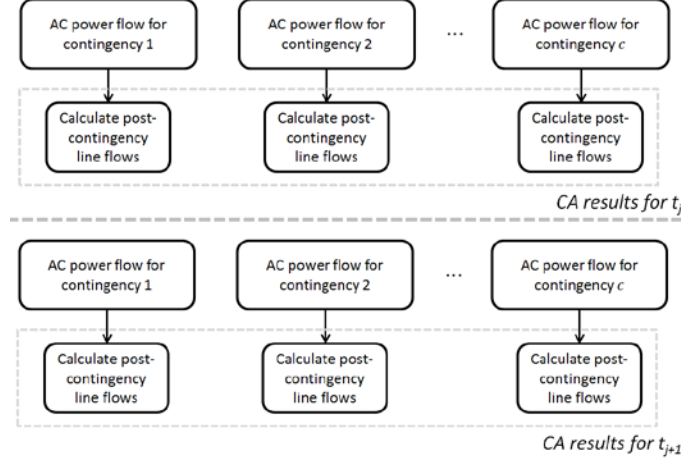
### **4.2.1 Motivation**

In this section we use the example of contingency analysis for the need to simulate a number of scenarios for the present system state and for the future seconds, minutes and hour operations. This “look-ahead” capability is not only associate with visualization, but with the computation necessary to produce system level information to be presented to the operator.

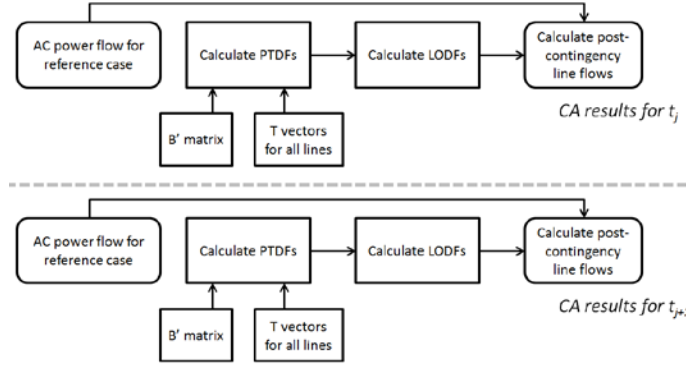
The ability of a power system to provide reliable service to its consumers at all times is critical. This operational reliability is ensured by contingency analysis. Possible contingencies include sudden shifts in load and the outage of system elements such as transmission lines and generators. Specified quantities of interest (e.g., flows across certain transmission lines) are calculated for the reference case and each post-contingency scenario. These results subsequently inform the alteration of the current system state in operational regimes or the comparison of transmission and generation expansion alternatives during planning stages.

Currently there are several solution methods for contingency analysis. AC Newton power flow provides the most accurate solution but is computationally demanding [19]. Under this approach, first an AC power flow solution is obtained and used as the reference case. For each contingency of interest, the contingency actions are implemented, a new AC power flow solution is obtained, the limit violations are reported, and the reference is restored. The whole process is repeated at every time step, as illustrated in Figure 4.3.

In studies where approximate solutions for post-contingency scenarios are acceptable, fast methods such as decoupled power flow [20] have been used instead to speed up the overall computation time for contingency analysis. In power systems where linearized load flow methods provide sufficient accuracy, linear sensitivities such as power transfer distribution factors (PTDFs) and line outage distribution factors (LODFs) may be used to achieve even greater speed [21], as illustrated in Figure 4.4. Once again the first step is to obtain a single power flow solution as the reference case. However, rather than solving many power flows, active power linear sensitivities are calculated under DC assumptions, and these distribution factors are used to compute the approximate line flows for each contingency and identify limit violations [22]. Active power distribution factors are widely used in the industry since they are regarded as a good tradeoff between accuracy and speed [23].



**Figure 4.3: AC contingency analysis for two arbitrary time steps. AC power flow must be computed for each contingency and time step.**



**Figure 4.4: Conventional DC distribution factor-based contingency analysis for two arbitrary time steps. At each time step, a single AC power flow solution for the reference case is obtained, and linear distribution factors are used to determine the post-contingency flows on lines of interest.**

In the real-time operational setting, a sizeable number of contingencies need to be evaluated every few minutes. In most cases, the operational system state changes only slightly between time steps. Similar expansion alternatives are likewise considered in the planning setting. The number of currently considered scenarios in both planning and operations is already substantial, and that number is growing steadily as technologies like variable distributed generation, demand response, and energy storage are integrated into the existing power system infrastructure [24].

Rather than calculating each post-contingency scenario separately and from scratch, this research proposes a new algorithm that uses distribution factors to directly move from the power flow solution for one scenario to the contingency analysis results for similar but different scenarios, thus reducing the number of times AC power flow solutions need to be found. This approach, described in detail in the following sections, has potential applications for large-scale look-ahead

system state exploration, which may ensure future power system security [25].

The evaluation of every possible contingency in large systems is computationally infeasible, so methods to reduce the number of contingencies under consideration are utilized [26]. Since the focus of this research is not contingency selection, the authors assume that the list of contingencies to be studied is provided as an input. Similarly, the choice of metric used to quantify the severity of contingencies is assumed to be given.

#### 4.2.2 Distribution Factors

Consider a power system with  $n$  buses and  $m$  transmission lines that is characterized by the sparse and symmetric  $n \times n$  bus admittance matrix  $Y_{bus}$ . Each diagonal element of the  $Y_{bus}$  matrix is equal to the sum of the admittances of all devices incident to that bus. Each off-diagonal element of the  $Y_{bus}$  matrix is equal to the negative sum of the admittances joining the two buses. The  $Y_{bus}$  matrix can be decomposed into real and imaginary components as shown below:

$$Y_{bus} = G + jB. \quad (1)$$

Under DC assumptions, all bus voltages are assumed to be 1.0 pu, and line resistances are neglected. Reactive power flows are also ignored. Hence, the full AC power flow equations simplify to

$$\theta = [B']^{-1} P \quad (2)$$

where  $P$  is the net active power injection at each bus and  $\theta$  is the bus angle. The  $B'$  matrix is the imaginary component of the  $Y_{bus}$  matrix formed without any shunt terms so that incremental currents are contained in the transmission lines rather than shorted to ground [27]. These assumptions are used to derive the following distribution factors.

The two most commonly used linear sensitivities in DC contingency analysis are the PTDF and the LODF. The PTDF measures the sensitivity of the real power line flow to an increase in real power transfer. A given transfer  $\psi$  from bus  $i$  to bus  $j$  can be represented as a  $(n - 1) \times 1$  zero vector, where the  $i$ -th element is 1, and the  $j$ -th element is  $-1$ . Let line  $a$  be a monitored transmission line between buses  $k$  and  $l$ . This relationship can be expressed as a  $1 \times (n - 1)$  zero vector  $\phi$ , where the  $k$ -th element is 1, and the  $l$ -th element is  $-1$ . The reactance of line  $a$  is known to be  $x_a$ . According to [28], the PTDF for line  $a$  impacted by a transfer from bus  $i$  to bus  $j$  is denoted as

$$PTDF_{a,\psi} = \frac{1}{x_a} \phi [B']^{-1} \psi. \quad (3)$$

Then the post-transfer flow on line  $a$ ,  $P_a$ , is a function of the pre-transfer flow on the line  $P_a^0$ , the transfer amount  $\rho_\psi$ , and the PTDF for a transfer from bus  $i$  to bus  $j$ :

$$P_a = P_a^0 + PTDF_{a,\psi} \times \rho_\psi. \quad (4)$$

The LODF measures the percentage of pre-outage flow on line  $b$  that will show up on line  $a$  after a contingency occurs on line  $b$ . By modeling the outage of line  $b$  as a transfer, the LODF calculation becomes a simple relationship of PTDFs [29].

$$LODF_{a,b} = \frac{PTDF_a}{1 - PTDF_b} \quad (5)$$

The post-contingency flow on line  $a$ ,  $\tilde{P}_a$ , is then a function of the pre-contingency line flow on line  $a$ , the LODF, and the pre-contingency line flow on line  $b$ :

$$\tilde{P}_a = P_a^0 + LODF_{a,b} \times P_b^0. \quad (6)$$

Another well-known and commonly used linear sensitivity is the outage transfer distribution factor (OTDF), which measures the fraction of a transfer that will show up on branch  $a$  after a contingency occurs on line  $b$ :

$$OTDF_{ab,\psi} = PTDF_{a,\psi} + LODF_{a,b} \times PTDF_{b,\psi}. \quad (7)$$

The post-transfer post-contingency power flow on line  $a$ ,  $\tilde{P}_a^{pc}$ , can be expressed as

$$\tilde{P}_a^{pc} = P_a^0 + LODF_{a,b} \times P_b^0 + OTDF_{ab,\psi} \times \rho_\psi. \quad (8)$$

#### 4.2.3 Problem Formulation and Algorithm

Consider again the power system with  $n$  buses and  $m$  transmission lines, this time with  $g$  generators and  $l$  loads. The net bus injection  $P_{inj}$  at any given time is a function of the load usage  $P_L$  and the generation dispatch  $P_G$ :

$$P_{inj}(t) = P_G(t) - P_L(t) \quad (9)$$

Assume at the present time  $t_j$ , the state of the system  $S_j$  is known and can be completely described by  $m \times 1$  line statuses (either ‘open’ or ‘closed’) and  $n \times 1$  bus injections. Also, all line parameters and limits are known. From these quantities, the power flow solution for  $t_j$  can be found. The traditional distribution factor approach described in Section I can then be used to determine the contingency analysis results for  $t_j$ . For the next time step  $t_{j+1}$ , the topology of the system remains the same, but the load usage  $P_L(t_{j+1})$  and generation dispatch  $P_G(t_{j+1})$  have changed. Now the contingency analysis results for  $t_{j+1}$  are needed. If the system changes are minor, we propose using the following algorithm to determine the contingency analysis results for  $t_{j+1}$  rather than recomputing the power flow solution and repeating the contingency analysis process.

1. Assume that the changes in generation and load are provided as inputs:

$$\Delta P_G(t_{j+1}, t_j) = P_G(t_{j+1}) - P_G(t_j) \quad (10)$$

$$\Delta P_L(t_{j+1}, t_j) = P_L(t_{j+1}) - P_L(t_j) \quad (11)$$

2. Find the scalar transfer amount  $\rho$ :

$$\rho = \sum_i \Delta P_G(i) \quad (12)$$

3. Normalize  $\Delta P_G$  and  $\Delta P_L$  to get a  $(n - 1) \times 1$  time-dependent transfer vector  $T_{time}$ :

$$T_{time} = \frac{1}{\rho} (\Delta P_G(t_{j+1}, t_j) - \Delta P_L(t_{j+1}, t_j)) \quad (13)$$

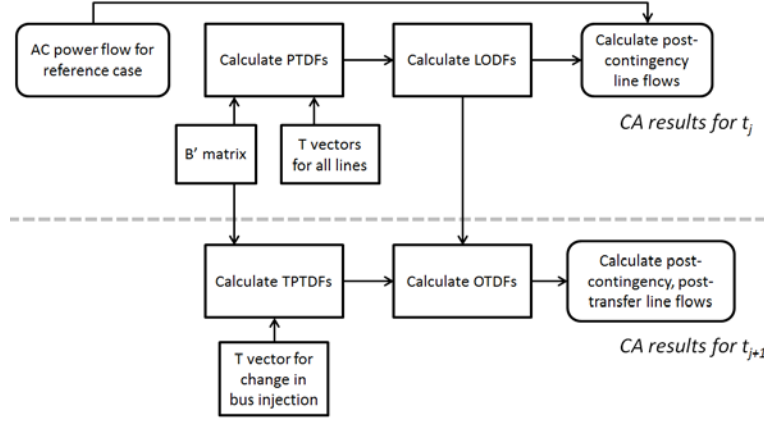
4. Calculate the time-dependent PTDFs (TPTDFs) for the transfer vector in (13):

$$TPTDF_{a, T_{time}} = \frac{1}{x_a} \phi^T [B']^{-1} T_{time} \quad (14)$$

5. Calculate the OTDFs (7) using the LODFs calculated during  $t_j$  and the TPTDFs from (14).
6. Finally, evaluate the impact of the combined transfer and outage on each monitored line using (8). If the monitored line exceeds its limits, the indices of the overloaded line and the outaged line are stored along with the percentage overload.

This algorithm is illustrated in Figure 4.5. Since the TPTDFs are based purely on bus injections, they can be used to capture a variety of different system changes, such as sudden shifts in generation or load as well as planned outages.

In the event that a change in system topology under normal operating conditions occurs at  $t_{j+1}$ , the  $B'$  matrix would need to be updated, and the LODFs should be recalculated. Then these quantities would be used to calculate the TPTDFs and the OTDFs.



**Figure 4.5: Proposed approach for two arbitrary timesteps, assuming no topology change. AC power flow may be omitted for steps after  $t_j$ .**

#### 4.2.4 Results

The aforementioned algorithm was applied to two test systems to compare its accuracy against conventional AC and DC contingency analysis. The first case is a small illustrative 7-bus example developed by PowerWorld [30]. The topology, line parameters, and line limits for this system are described in the appendix. The second case is the 1996 IEEE 24-bus reliability test system [31]. For both systems, each line is outaged in turn and its effects on the other lines are quantified.

The AC contingency analysis results were obtained via the approach described in the introduction. For both DC contingency analysis methods, first an AC power flow solution was found for the reference case at time  $t_j$ . At time  $t_{j+1}$ , the system experienced a planned change. The distribution factors were calculated and applied to the active power line flow results from time  $t_j$  to find the post-contingency active power line flows for the system at time  $t_{j+1}$ . The reactive power was assumed to be constant from one time step to the next. The apparent power line flow was calculated using the active and reactive power line flows. Then it was checked against the MVA line limit for each line.

#### 4.2.5 Illustrative 7-Bus Example

At time  $t_j$ , five generators and six loads are active. Their respective MW and MVAR injections are shown in Table 4.1. At time  $t_{j+1}$ , the same generators and loads are active, but the loads at bus 3 and bus 5 have increased by 20 MW and 10 MW respectively. The generators at bus 1 and bus 2 correspondingly increased by 12.34 MW and 18.7 MW.

**Table 4.1: Different Bus Injections over Time**

Bus	time $t_j$				time $t_{j+1}$	
	Gen [MW]	Gen [MVAR]	Load [MW]	Load [MVAR]	Gen Change [MW]	Load Change [MW]
1	101.85	5.25	0	0	+12.34	
2	170.08	33.24	40	20	+18.70	
3	0	0	110	40		+20
4	95.03	19.99	80	30		
5	0	0	130	40		+10
6	200.33	-6.59	200	0		
7	200.65	51.29	200	0		

Table 4.2 offers a comparison between the accuracy of the traditional distribution factor approach and the new proposed TPTDF approach. Since the system change was small, the errors are comparable between the two methods.

**Table 4.2: Comparison of Contingency Analysis Results for PowerWorld 7-Bus Example**

Outaged Line	Affected Line	AC % Limit	DC % Limit	New % Limit	DC % Error	New % Error
1	2	153.9	176.2	176.3	14.5	14.6
3	2	102.0	99.8	99.8	2.2	2.1
6	5	121.0	117.8	118.1	2.7	2.4
7	2	108.1	99.7	99.8	7.8	7.7
9	5	127.3	118.2	118.1	7.1	7.2

#### 4.2.6 IEEE 24-Bus Reliability Test System

For the IEEE 24-bus case, the representative changes in bus injection are shown in Table 4.3. In the second scenario, all seventeen loads in the system increased by 10%, which represents the summer peak load. Four generators increased their output correspondingly.

**Table 4.3: Representative Changes in Bus Injection over Time**

Bus	time $t_j$		time $t_{j+1}$	
	Gen [MW]	Gen [MVAR]	Gen Change [MW]	Load Change [MW]
101	151.51	94.34	+40.49	All +10%
107	48.24	46.86	+99.05	All +10%
113	436.58	164.69	+99.02	All +10%
115	184.51	9.37	+30.49	All +10%

Table 4.4 shows a comparison between the accuracy of the traditional distribution factor approach and the new proposed TPTDF approach when the system change is significant. AC contingency analysis flagged four line MVA violations, while both DC-based methods only flagged two,

which is a consequence of relying on direct linear methods to estimate the post-contingency line flows. The accuracy of the traditional DC method and the new DC method are again comparable, but there is a small loss of accuracy in the new approach since the system deviation from the reference case was fairly large.

**Table 4.4: Comparison of Contingency Analysis Results for IEEE 24-Bus Test System**

Outaged Line	Affected Line	AC % Limit	DC % Limit	New % Limit	DC % Error	New % Error
10	1	136.2	53.0	44.8	61.1	67.1
10	5	109.0	88.9	86.6	18.4	20.6
12	13	110.5	104.0	102.5	5.9	7.2
13	12	111.0	104.8	103.9	5.6	6.4

#### 4.2.7 Computational Complexity

In computer science, big- $O$  notation is frequently used to compare the performance of different algorithms. This type of analysis is more instructive than simple timing in assessing an algorithm's efficiency, since it considers only the relationship between the number of operations and the overall problem size and abstracts the algorithmic performance from confounding factors such as software- or platform-specific optimization.

Consider the power system with  $n$  buses and  $m$  transmission lines. To simplify the analysis of computational complexity for different contingency analysis algorithms, assume that the number of single line outages to be considered is equal to  $m$ .

According to [32], the computational cost of Newton (AC) power flow is  $O(j^{1.4})$  for factorization and  $O(j^{1.2})$  for each iteration of the solution when the Jacobian is a sparse  $j \times j$  matrix. Since the number of iterations necessary for convergence is usually small and the cost of factorizing the Jacobian dominates, the overall complexity of obtaining an AC power flow solution is  $O(j^{1.4})$ . The computational complexity of  $N - 1$  AC contingency analysis is a function of the complexity for AC power flow, the number of impacted lines  $m$ , the number of contingencies  $m$ , and the number of time steps. For each time step and contingency, AC power flow is solved at a cost of  $O(j^{1.4})$ , and  $O(m)$  transmission line flows are calculated. Hence the overall complexity is  $O(j^{1.4}mt + m^2t)$ .

For traditional  $N - 1$  DC contingency analysis, the complexity is a function of the cost to obtain an AC power flow solution for the reference case and computation of the distribution factors for each time step. The cost of obtaining the reference AC power flow solution is  $O(j^{1.4})$ . To find the cost of calculating the PTDFs, we first assume that the  $B'$  matrix is size  $k \times k$ , where  $k$  is equal to  $n - 1$  which is approximately  $n$ . Note that the number of buses  $n$  is smaller than the size of the Jacobian matrix  $j$ . The  $B'$  matrix is factorized at a cost of  $O(n^{1.4})$  and that factorization is used  $m$  times at a cost of  $O(n^{1.2})$  for each  $T$  vector. Under this assumption, the cost of

computing the PTDFs becomes  $O(n^{1.4} + mn^{1.2})$ , which reduces to  $O(mn^{1.2})$ . The cost to calculate the LODFs and the post-contingency line flows is  $O(m^2)$  since these calculations are done for every pair of lines. Then the overall complexity is  $O(j^{1.4}t + mn^{1.2}t + m^2t)$ .

For the new proposed method, the same DC contingency analysis process is repeated for the first time step. Hence, that cost is  $O(j^{1.4} + mn^{1.2} + m^2)$ . For each subsequent time step, we reuse the factorized  $B'$  matrix to calculate the TPTDFs at a cost of  $O(n^{1.2})$ . The cost of calculating the OTDFs and post-contingency line flows is  $O(m^2)$ . The sum of the terms becomes  $O(j^{1.4} + mn^{1.2} + m^2 + n^{1.2}t + m^2t)$ , but the smaller terms can be absorbed. Overall the algorithm's complexity is  $O(j^{1.4} + mn^{1.2} + n^{1.2}t + m^2t)$ . Compared to traditional DC contingency analysis, the cost of calculating additional AC power flows was eliminated, and only one TPTDF vector is necessary for each time step. Since its order of complexity is lower than that of traditional DC contingency analysis, the new algorithm has the potential to be faster.

#### 4.2.8 Exploration of Other Approaches

Another approach to solving DC contingency analysis is to consider each line outage as a perturbation to the original  $B'$  matrix. In traditional DC power flow, the  $B'$  matrix is first factored into LU form. When the prefactored matrix is Hermitian positive-definite, Cholesky factorization which is faster than LU can be used. For each single line outage considered during contingency analysis, only four entries in the  $B'$  matrix are altered. Hence, this problem can be formulated as a low-rank downdate to the initial factors and then used to solve for the subsequent bus angles and line flows.

Three standard power systems were used as test cases to compare timing results between four different approaches. The PowerWorld 7-bus system is considered a highly-interconnected network whereas the IEEE 24-bus and 118-bus systems are relatively sparsely-interconnected. All three have HPD "A" matrices.

This project used Tim Davis's CHOLMOD package to perform a series of low-rank downdates to the initial sparse LDL' factorization of A. This was tested on the aforementioned test systems. The timing results for 30 runs were compared against the results from several other methods: sparse LL' factorizations, dense updates, and dense LL' factorizations for each line outage. The time distributions for each system are captured in Figures 4.6-4.9.

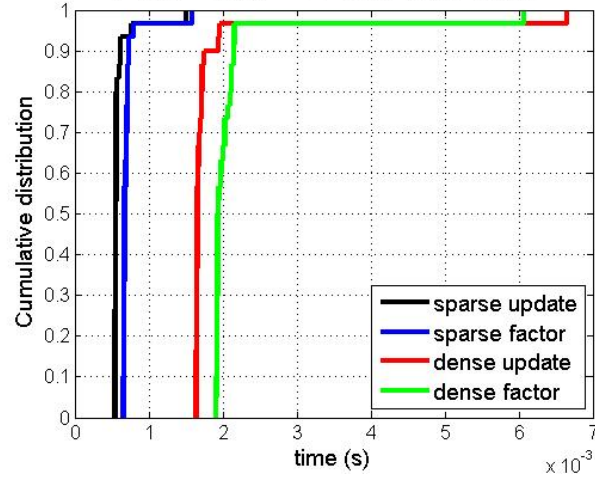


Figure 4.6: Time distribution for the PowerWorld 7-bus test system across 30 runs

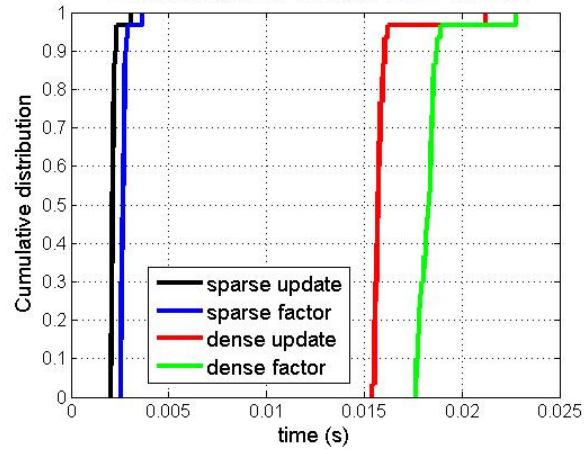


Figure 4.7: Time distribution for the IEEE RTS 24-bus test system across 30 runs

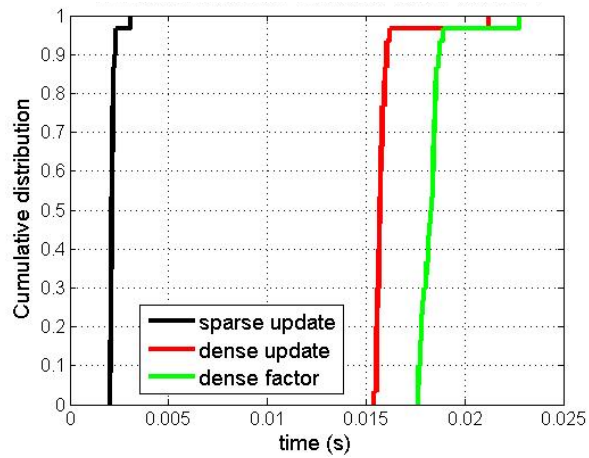


Figure 4.8: Time distribution for the IEEE 118-bus test system across 30 runs

The objective was to investigate whether sparse Cholesky downdates would work well for DC contingency analysis. This method is only applicable when the original  $A$  matrix is HPD and the resulting new  $A$  matrices from the downdates are also positive-definite. For all three considered systems, the original  $A$  matrices were HPD, but in the 24-bus and 118-bus cases, the resulting new  $A$  matrices from the downdates were not all HPD. However, timings were still included for general comparison.

This update approach could be extended to DC contingency analysis, but Cholesky factorization specifically has limited applicability. After further literature review, it seems that the two necessary conditions for Cholesky factorization are frequently not met. One issue arises from how large shunt terms are included in  $B'$ , which causes the matrix to be indefinite. This was the case for a larger 1000-bus system. Another issue occurs when the network is not interconnected enough. In that scenario, the outage of a tie line essentially cuts the network into multiple pieces, and this method of analysis is no longer accurate. This was illustrated in the 24-bus and 118-bus cases.

In general, sparse updates are slightly faster than sparse factorizations, while dense updates are somewhat faster than dense factorizations. Both sparse methods are considerably faster than the dense ones.

## **4.3 Application of High Performance Computing**

### **4.3.1 Introduction**

High Performance Computing (HPC) is a fast growing domain that has influenced work in many other domains as more and more people begin to realize its importance and necessity. Although HPC is vastly employed to help improve the performance of applications, its use also leads to many other benefits such as efficient use of resources and broadening the usage domain of an application.

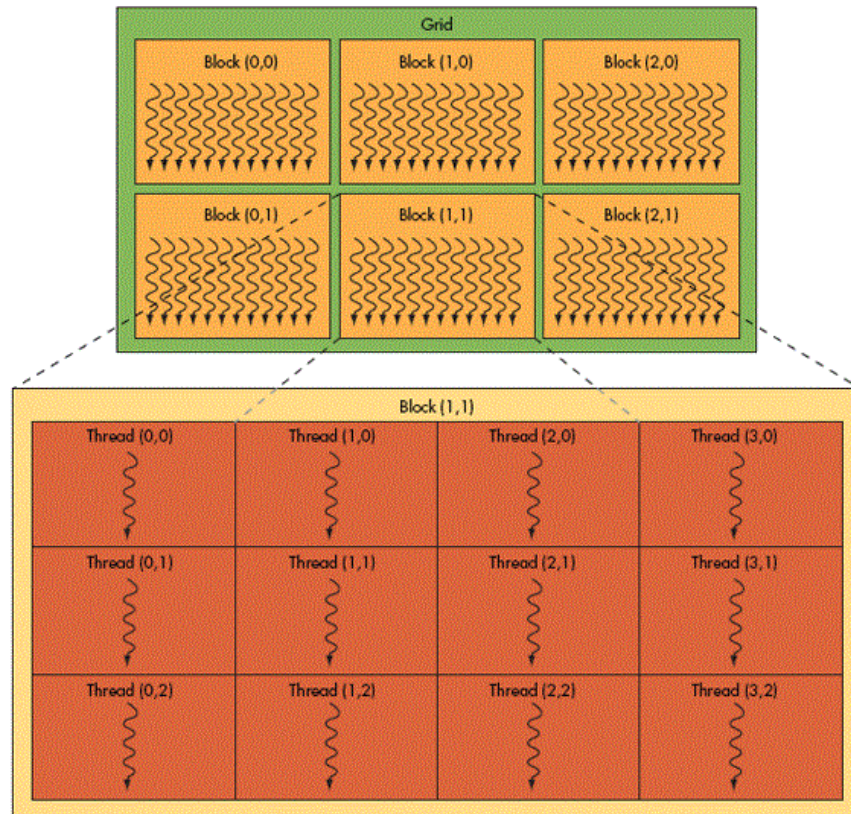
In this regard, the domain of power systems is not far behind in leveraging the benefits of HPC for performance enhancement. As the literature review will discuss in more detail, a great deal of research has been conducted in this area. The reason why HPC has become important in this field is the sheer size of the data sets that power systems have to deal with. Different types of data come in from different sources and computations have to be done in real time. Most of these computations are fairly complex. Energy being a resource that drives so many systems around the world, one can imagine how critical it is for an Energy Management System to run grid algorithms in real-time based on field measurements. In such a scenario, EMS can highly benefit from the use of HPC resources.

### 4.3.2 Literature Review

For years now people have been trying to leverage the benefits of HPC in power systems. The interest in HPC had begun to develop very early on and people had recognized power systems as an engineering domain that could demand more computational power in the future. HPC architectures such as vector processors and SIMD massively parallel machines were considered in the 1990s. Thereafter, for many years, this area was not given enough attention by the power system community, and vendors had reservations regarding use of parallel architectures. However, as the high performance computing industry continued to gain momentum, newer HPC technologies were developed and people again began to investigate ways of incorporating this growing trend into power systems with the interests rising with respect to grid computing, multicore and many-core architectures <sup>[3]</sup>. The latter was popularized by the development of GPUs.

Previous work in this area has been targeted at applications such as contingency analysis and state estimation. For example, GPUs have been applied to solving contingency analysis. Contingency analysis has also been implemented using MPI and PVM.

We further explore this area and here we investigate the application of CUDA, a parallel programming language used to program GPUs, to power system analysis.



**Figure 4.9: Grids, threads and blocks in the CUDA layout for GPUs**

### 4.3.3 Incorporation of HPC in Power System Analysis

It has been possible to integrate GPUs into existing platforms as long as certain criteria are satisfied. Some algorithms can be parallelized fairly easily, while others need to be reworked substantially in order to enable use of parallel processors. In doing so, it is necessary to ensure correctness of the parallel algorithm.

The algorithm that performs contingency analysis using a DC power flow includes computations on matrices such multiplication and inversion. The matrices could be fairly large and the computations are carried out on floating point data. Thus, the use of a GPU in the implementation of this algorithm could lead to faster execution.

The main elements of the algorithm that we wish to focus on are the portions that deal with computation on matrices. Consider the following equation for DC power flow:

$$\theta = -[B']^{-1} P$$

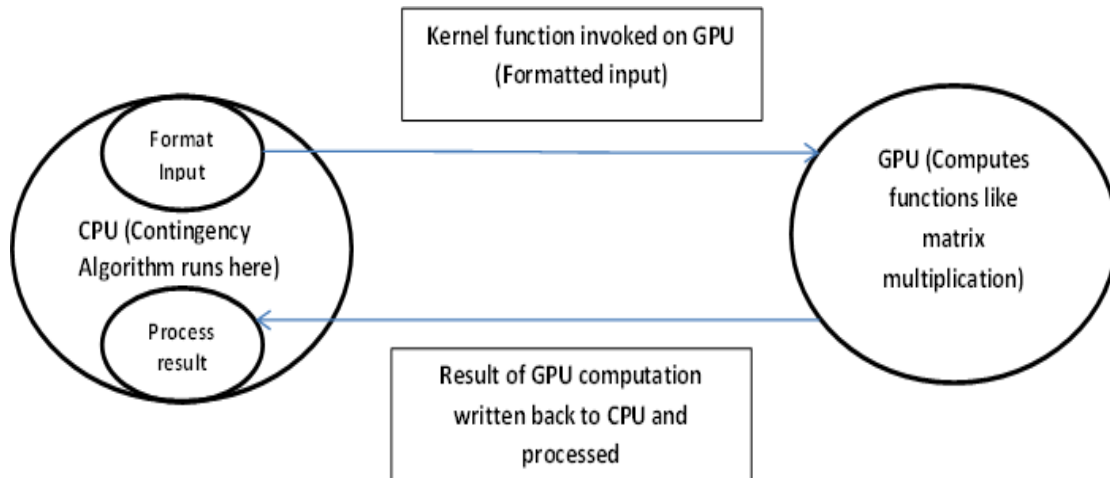
In this equation  $P$  is an  $n \times 1$  vector and  $B'$  is an  $n \times n$  matrix. The computation performed in the equation is an inversion of  $B$  followed by a multiplication with  $P$ . In the event of the value of  $n$  being fairly large and the elements of the matrix being floating point numbers such a computation can be expensive. This is where we propose to export the computation to a GPU where each thread would independently compute an element of the final vector  $\theta$  and could significantly save execution time. The execution of the equation below,

$$P(\text{line}) = \frac{1}{X_{jk}} \sin(\theta_j - \theta_k) \approx \frac{1}{X_{jk}} (\theta_j - \theta_k)$$

follows for  $m$  lines. This computation could again be done using a GPU where each thread would calculate the value of  $P$  for a single line, and  $m$  threads would run simultaneously.

### 4.3.4 System Architecture Overview

A broad overview of the system consists of two primary components: the CPU that invokes the kernel and the GPU that computes the kernel function. Before the function call to the GPU takes place, the data that is sent to the GPU has to be in an appropriate format. For example, the flattening of two dimensional arrays is required in case the data is structured in that form. Once the GPU computation is done, the result is copied to the CPU where again some post-processing may be required. A basic diagram that outlines the steps can be viewed in Figure 4.10.



**Figure 4.10: High level overview of the system architecture**

#### 4.3.5 Investigation of HPC Applications

The goal of the HPC investigation is to identify areas where performance enhancement could be introduced, because our application has requirements of scalability and high performance due to real time computational constraints and large data sizes that will only get larger with time.

During our initial research, we had identified two major components of the contingency analysis algorithm that had potential for parallelism and were hence candidates for performance enhancement, namely, matrix multiplication and matrix inversion. Besides being operations that could easily be parallelized, there are other reasons why we choose to focus on these components:

- We realized that these two operations were performed within loop bodies and comprised a major portion of the execution time. Any operation with a loop body is carried out multiple times and an optimization of such an operation results in a major reduction in execution time.
- We were able to identify that there existed no dependency between different iterations of the loop body. To elucidate, consider matrix multiplication in contingency analysis. During each iteration, a separate element of the matrix is computed, and this computation does not depend on any other element of the matrix. Thus, we were able to identify independent computations in the contingency analysis algorithm.
- These operations work with big data, i.e., the matrix sizes are large typically of the order of  $10^4$  in one dimension. The size of this data justifies the use of parallelism and frameworks that speed up computation along with introduction of additional overhead, as the latter is amortized as the data sizes increase.

We have implemented matrix multiplication for GPUs using CUDA and have compared the results with a naïve CPU implementation. The results can be seen in Table 4.5.

As can be seen from the results, the GPU implementation provides a definite speed-up in all scenarios and specifically for large matrices where the GPU utilization is maximum and therefore the overhead due to transfer of data between the host and the device is amortized due the large matrix size. It is these cases that clearly depict the contrast between the CPU and GPU implementations.

**Table 4.5: CPU vs. GPU for Matrix Multiplication**

<b>Matrix Size</b>	<b>CPU (s)</b>	<b>GPU (s)</b>
40 x 40	0.000713	0.000111
100 x 100	0.0098	0.000906
1000 x 1000	10.04	0.98
5000 x 5000	1702.83	10.84
10000 x 10000	14542.59	11.31

Besides matrix multiplication, another component of contingency analysis that could possibly be an opportunity for parallelism is matrix inversion. The steps in matrix inversion are as follows:

- Calculate the matrix determinant
- Find the transpose of the matrix
- Calculate cofactors and represent them as a matrix
- Multiply the resultant matrix with the determinant.

Looking at these steps, we were able to identify two components in this computation that could be ported on a GPU. The calculation of cofactors could be independently performed by each individual thread and similarly, each thread could perform the multiplication of the determinant with each matrix element. While exploring these options, we realized that our implementations were not as efficient as some other tools that were available for such optimizations. Also, we were performing all operations on dense matrices. However, some computations in the contingency analysis algorithm involved sparse matrices. This motivated us to study tools available that were already optimized to perform such operations not only on dense but also on sparse matrices. Our research led us to a tool CULA, a suite of functions that perform operations such matrix multiplication and inversion on dense as well as sparse matrices. For example, the GETRI function in CULA performs matrix inversion using LU factorization.

## 5 Look-Ahead Visualization

### 5.1 Visualizing Power System Data

Power system operators, planners, and managers need to analyze vast quantities of data in order to make effective decisions for the bulk electric grid. The analysis process is very complex for large systems containing thousands of buses and transmission lines. Hence, the challenge is to present information in a manner which facilitates intuitive and rapid assessment of the system state [33]. To mitigate the information overload experienced by grid operators and decision makers, it is necessary to develop a visualization platform capable of quickly transforming large datasets into visual representations that are easy to understand at a glance.

Of special importance is the ability to analyze large amounts of temporal (both look-ahead and past) data. Look-ahead capabilities are crucial for understanding the emerging stochastic nature of the grid due to the integration of non-dispatchable renewable energy, demand response, and storage. These capabilities can be used by operators to assess the likelihood of possible future problems in the network. Similarly, they can be used by planners to predict the effect of network expansion. Being able to view past system states can help operators pinpoint how an event began and to develop insight for future actions, as well as to support overall post-operational analysis.

Various visualization methods have been developed in the past to aid in the interpretation of power system data, although only some have actually made their way into the hands of practicing engineers. A few of these techniques include animated flow arrows [34], bus voltage and transmission line contours [35], and 3D bar graphs [36, 37].

Most of the existing work has focused on representing power system data for only one point in time. This type of static large-scale 2-D visualization does not support the aforementioned requirements for temporal analysis. One way to fill this void is a visualization that allows users to see multiple time steps simultaneously. The natural reaction is to consider simply stacking 2D system states from different time points on top of one another in order of time progression. However, this concept has a key disadvantage. Users cannot clearly see the relevant power system data, due to the unnecessary repetition of the system one-line diagram and the cluttering effect of so many rendered elements.

This project offers a hybrid 3D and 2D approach to power network visualization. We created a prototype tool to allow fast network analysis and detailed reading. The prototype is designed as a side-by-side visualization tool that concurrently operates on time-varying network measurements. By leveraging visual analytics techniques, we can display two simultaneous views of complex network data. With both the high-level exploratory power of the 3D view and the fine-grained traditional 2D view, our visualization concept supports the best of both worlds.

## 5.2 Literature Review

In this section, we briefly review relevant state-of-the-art work for visualizing time-varying networks in the area of information visualization and visual analytics.

### 5.2.1 2D Dynamic Network Visualization

Using time sliders to explore time-varying networks is generally more effective rather than providing animations. Farrugia et al. [38] conducted studies to compare two common approaches: animation and static snapshots. They concluded that static snapshots are generally more effective in terms of task completion time. To tackle the limitation of the snapshot-based approach, DiffAni [39] integrated several interaction techniques into a snapshot-based visualization.

Instead of having only one view, having multiple coordinated views is effective for exploring diverse aspects of different systems. GraphDiaries [40] presented a visualization interface for time-evolving graphs and focused on animated transitions for highlighting changes in graphs. The interface consists of several small views, including the main graph view, timeline, and history. Similarly for our 2D visualization, we also designed and implemented several coordinated views, including the main graph view, timeline, and details.

### 5.2.2 3D Dynamic Network Visualization

Recent research has investigated the use of three-dimensional space for visualizing time-varying networks. Itoh et al. [41] stacked 2-dimensional planes on a horizontal axis to represent time-varying networks. Tominski et al. [42] presented a similar approach for visualizing spatiotemporal data, but they used a vertical axis for time. We also chose to use this stacking approach for visualizing the evolution of power system states.

Previous work has also attempted a hybrid approach of using 3-D and 2-D. MatrixCube represents a time-evolving graph as a 3D cube by stacking adjacency matrices. Each graph can be represented as a matrix. This type of cube allows users to easily explore graph in various perspectives using simple operations, such as filtering by slicing the cube (e.g., show only for 3-6pm).

## 5.3 Visualization Data Management

This section briefly explains how data is managed and used in the visualization. We store all the necessary visualization data in a relational database. SQLite was chosen for its reputation as a popular lightweight relational database system.

### 5.3.1 Database Schema

Our database consists of a series of tables. First, the following three tables store the basic static information about buses and transmission lines, including their position.

- bus\_info: basic bus information
- line\_info: basic line information
- display: position information about generators, loads, etc.

Secondly, the following tables store time-varying data for each element.

- bus\_data: time-varying data for bus values
- line\_data: time-varying data for line values

### 5.3.2 Converting the Data

When developing and testing our prototype, we used the IEEE 7 bus case and synthetic time-varying data. We obtained system data from PowerWorld, and we wrote a collection of programming scripts in Python and Java to transform the raw data format into an appropriate format for our relational database. Then we generated synthetic time-varying data, which contained 24 temporal values for each element.

## 5.4 Initial Concept

### 5.4.1 3D Visualization Concept

The initial 3D visualization concept is shown in Figure 5.1. One dimension can be used to indicate past and future timelines. The other two can be used to represent the system topology. Because the system one-line diagram remains constant across time, it only needs to be displayed once. Instead the emphasis is on shifting system quantities, such as line thermal limits. This technique can be extended to other values such as generation or load levels.

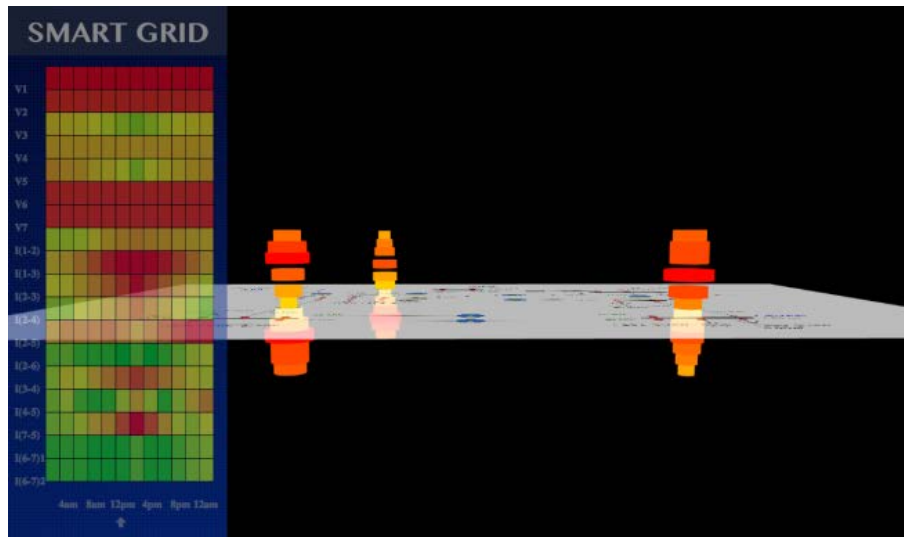
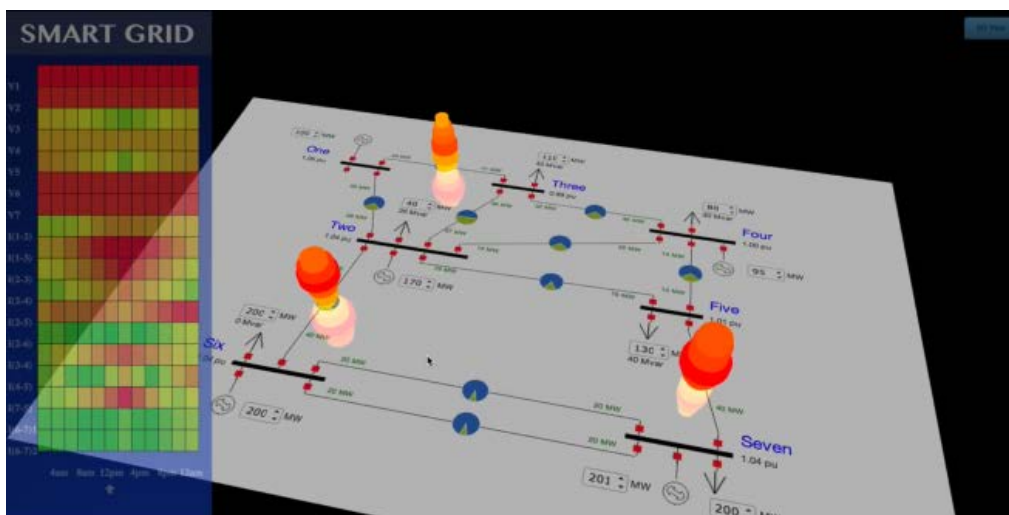


Figure 5.1: 3D prototype representation

The system one-line, shown as a semi-transparent floor plan in the middle of the screen, is used to indicate the present time. The vertical z-axis indicates different system states over time. The +z direction represents future predicted system states. A point higher along the +z axis indicates a time further into the future. The -z direction represents historical system data. A point lower on the -z axis indicates a time further in the past. The stacked cylinders represent line thermal limits in the network at different time points. The size and color indicate the severity of the problem. The redder the cylinder and the larger its radius, the more severe the line overload is.

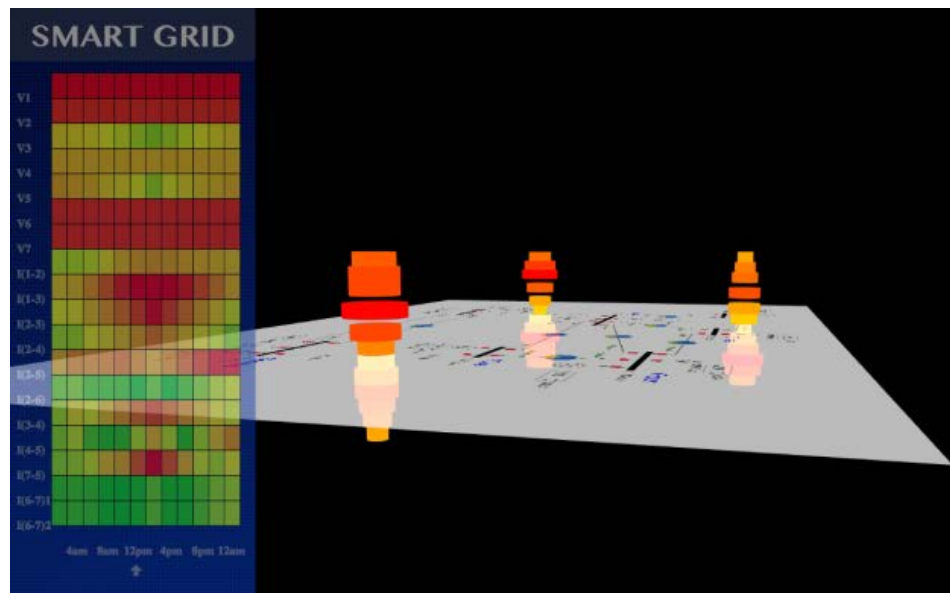
This 3D representation gives users the ability to view system conditions across a window of time, so that they are able to determine whether a problem is present in the system, as well as when it began and when it is expected to end, which may change as the problem evolves. The interactive nature of this visualization allows the user to navigate between system states one step at a time. To help distinguish between the current time and any past or future states that the user is navigating, the cylindrical slice corresponding to the current time is separated from the overall stack.

The heat map on the left panel shows the user more detailed information about the network. Each column along the horizontal dimension shows an interval of time, the length of which depends on the use case. Each row indicates the state of a different power system quantity in the network, such as bus voltage or line limit. The color shows whether the value of the attribute at a specific time falls within the normal range. Accounting for human factors, green was chosen for normal operation while red indicates that urgent action is required. Also, a small white arrow points to the present time.



**Figure 5.2: 3D top view**

As shown in Figures 5.2 and 5.3, the user has the ability to pan around the 3D visualization to take a closer look at any point of interest. For example, in Figure 5.2, the user may want to simultaneously view the system topology and also see the trend of line thermal limit violations. In Figure 5.3, the user may want to take a closer look at the transmission line between Buses 5 and 7.



**Figure 5.3: 3D side view**

#### 5.4.2 2D Visualization Interface Concept

Sometimes more detailed information is needed for further network analysis. Hence, the ability to navigate between the 3D view and a 2D view is essential. When a specific time instance is selected in the 3D view, the user can transition seamlessly to the 2D view showing the system state for that point in time. Figure 5.4 presents the 2D user interface concept.

The 2D representation in this visualization corresponds to the traditional one-line diagram that users are accustomed to seeing, which increases the ease of personnel training. Buses are represented by horizontal lines. Each bus is labeled with its name. Bus voltage and line flow values are placed next to their corresponding elements. Users can increase or decrease generation and load levels at each bus by clicking the arrows located next to the appropriate element. The simulation is initiated when the user clicks the run button in the upper right hand corner, and flow arrows will appear as shown in Figure 5.5. The top panel contains a dashboard where users can modify the system topology. The left panel displays the 3D view so that users can continue to monitor emerging system trends.

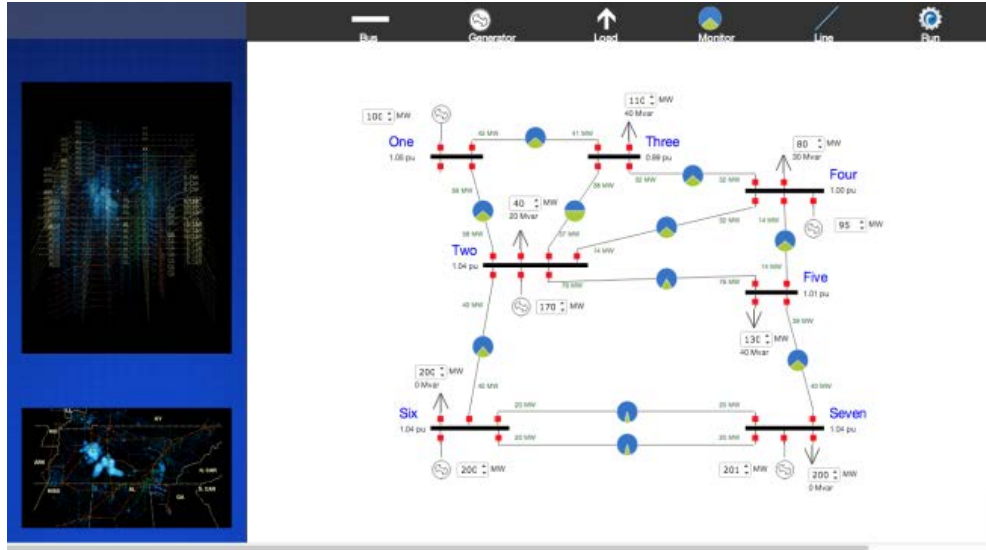


Figure 5.4: 2D user interface

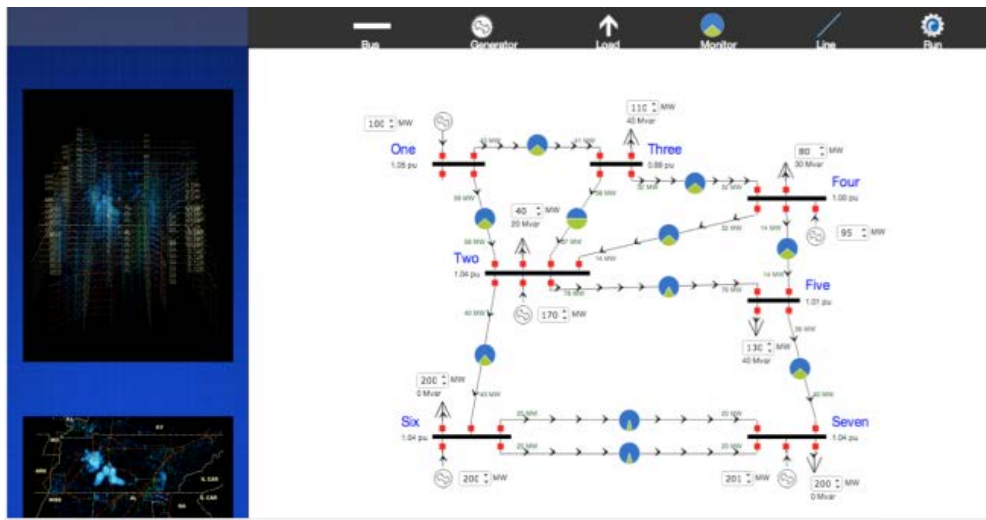


Figure 5.5: 2D run mode

## 5.5 Navigational 3D Power System Visualization Prototype

We created a 3-dimensional exploratory tool to investigate the evolution of a power system over time. The tool visualizes the system state (voltages, max violations, etc.) over time, using color and shape to denote changes in system behavior.

The major goal of this approach is to facilitate fast investigation and exploration of dynamic behavior in a power system. Because the changes in state are summarized by shape and color, the

overall state of the system can be understood by the user without the need to read individual values.

Many contemporary power system visualization tools such as PowerWorld only employ 2D layouts that summarize exact values at an instantaneous point in time. Usually these tools leverage animation to show how the system changes over time. Often comparison of system-wide behavior at two time points is challenging, and comparison of three or more time instances is nearly impossible. This is one of the natural limitations when using animation to show trends over time.

We chose to implement a tool that would allow the user to visually compare several consecutive measurements at once. This objective is facilitated by visualizing the bus and line quantities with color and volumes that change with the associated values. Instead of using animation to show changes, we use 3D space to display each consecutive time step. By allowing multiple comparisons, the user will be able to understand the overall system behavior with greater clarity.

### 5.5.1 Implementation Overview

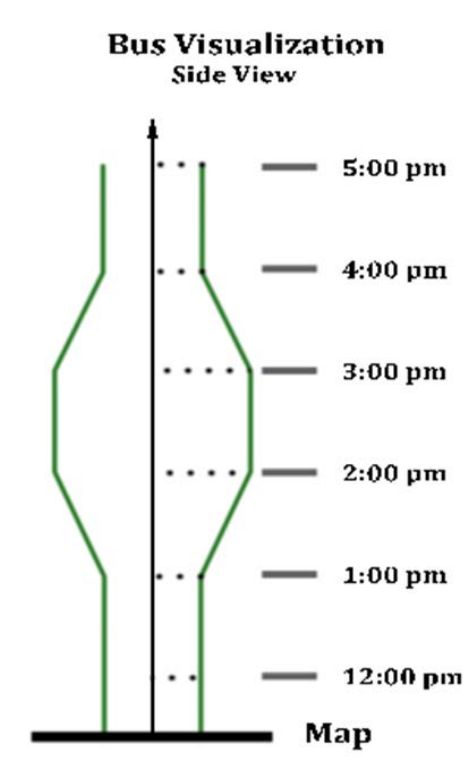
We implemented our prototype visualization tool using WebGL. WebGL is a web browser based 3D rendering environment. WebGL was selected for 1) its performance and 2) its compatibility.

- 1) WebGL offers excellent 3D performance by leveraging graphics hardware at a low level. It allows graphically complex scenes to run smoothly and enables real-time interaction and exploration. The size, content, and aesthetics of our visualization are built using WebGL as their foundation.
- 2) WebGL allows 3D environments with rich user interaction directly in a web browser. Currently most mainstream web browsers (e.g., Chrome, Firefox, Internet Explorer 11, Safari) can support WebGL. By choosing WebGL, our 3D visualization can be run on many systems with a wide variety of graphics hardware with very little additional setup.

### 5.5.2 Buses

Each bus in the system is visualized as a vertical volume. The volumes are created by modifying the radius of a vertical column based on the desired quantity to be visualized at each time step.

Consider Figure 5.6 which shows a cross-sectional side view of a single bus value being visualized with our approach. The timeline next to the volume indicates the time associated with each volume slice. The map marks the beginning of the time frame of interest, and time values increase as they move up the volume. In this example, it is easy to see that the visualized bus quantity from 2-3pm is considerably larger in magnitude than at any other time.



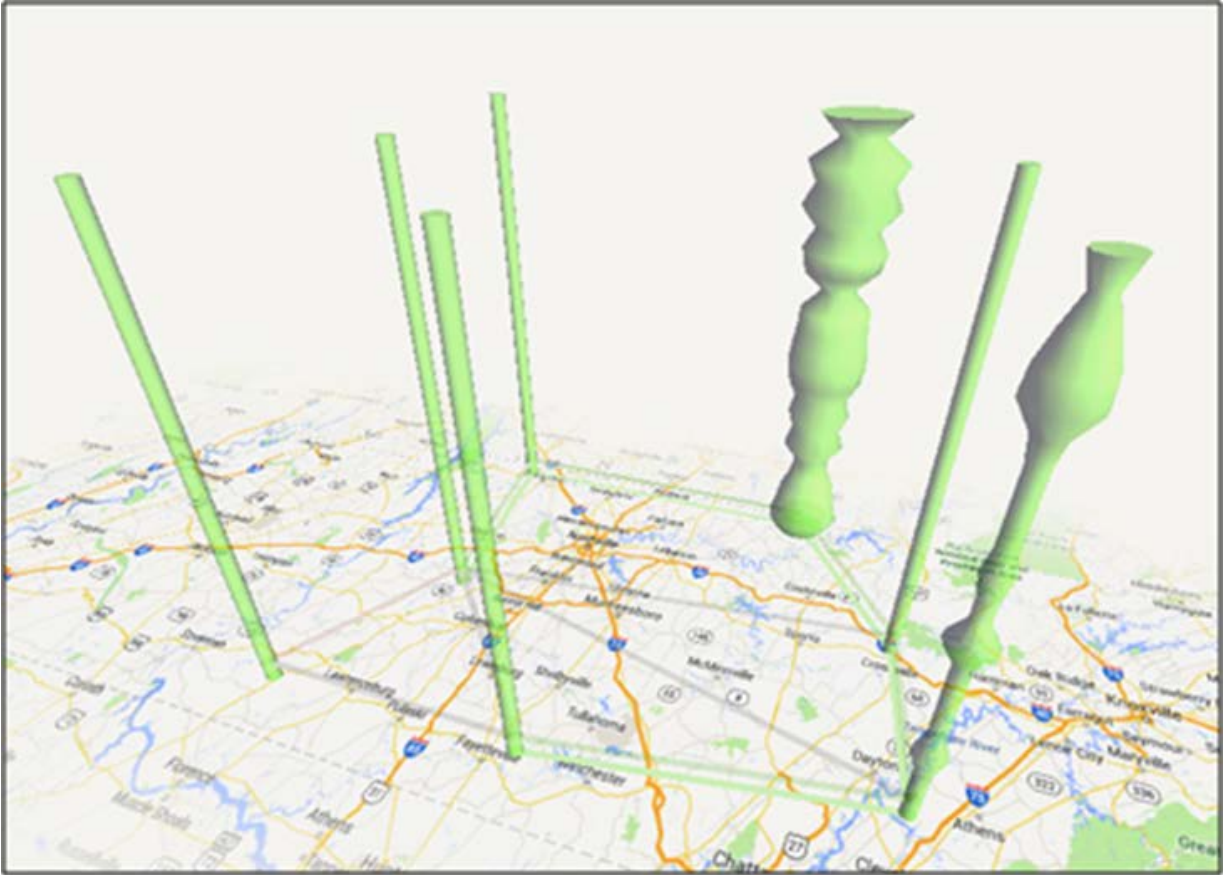
**Figure 5.6: Side view of a bus volume**

Each bus in the system can be visualized with such a volume. For additional customization, the user has the ability to select the bus quantity to be visualized.

Because certain quantities may have extremely small values or little deviation, we also allow the user to view a modified version of the volume. This scales the data and enhances any data variation so that these trends can be captured by a user-friendly visual representation.

A snapshot of a 7-bus test network that displays only the bus volumes during the study period is pictured in Figure 5.7.

In order to show dramatic changes over time, we decided to investigate the reactive power generation at each bus. Most of the buses appear to be a constant-width column, which implies that their values do not change much over the study period. However, two of the volumes (on the right) clearly indicate a lot of variation in the studied attribute. With this approach we can also detect the nature of the changes in behavior. The center-right bus volume experienced the most change over time, as shown by its repeating ridges.

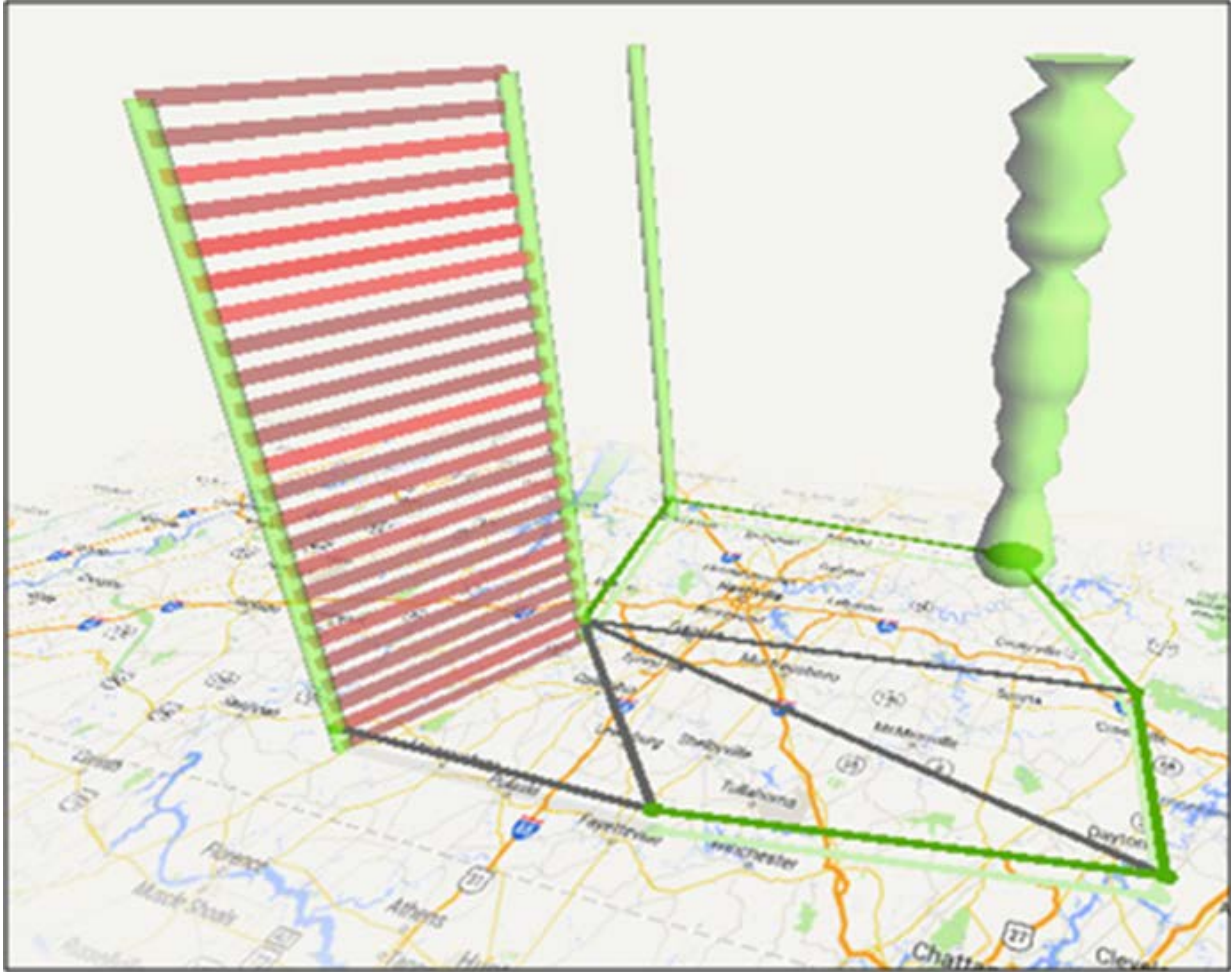


**Figure 5.7: Visualization of bus volumes for a 7-bus system. A quantity represented could be MW generation connected to the bus.**

### 5.5.3 Lines

Transmission lines are visualized with color-coded lines that run between the buses they connect. The color of the line is based on the value of the transmission line at that time. When the percent thermal limit violation is very low, the associated line is not drawn. As the loading increases, the transmission lines become more and more visible. In addition to the change in visibility, the lines also change from grey to dull red once they pass 90%. As the line flow increases, the red color becomes brighter and brighter until it becomes a stark red at above 105%. We chose to hide the lightly loaded lines by default, because they generally operate as expected and are less critical than the lines that are near or above their maximum thermal limits.

The lines in Figure 5.8 illustrate the changes in color and opacity as the system evolves over time. In this scenario, the shown line was over capacity for the entire study time. Each “rung” on this ladder represents the line’s percent limit violation for the simulated time step.



**Figure 5.8: Visualization of line loading for a 7-bus system**

#### 5.5.4 User Interaction

This system was built to allow the exploration of our 3D visualization. Currently our prototype supports:

- Panning – move the scene with the mouse by clicking and holding.
- Rotation – rotate the scene to investigate buses and lines from various angles, which yields a plethora of views of the system state.
- Zoom – if the system is too large, the mouse wheel can be used to shrink or magnify the scene, allowing the user to zoom into an interesting area or capture a higher-level view of the system.
- Item highlighting – when a user hovers over a line or bus, the object becomes selected with its colors bolded to help it stand out visually.
- Click-hiding – the user can hide any of the buses or lines by clicking on them in the visualization.

### 5.5.5 Interface

To keep the interface unobtrusive, we created a collapsible interface (shown in Figure 5.9) to allow control over several aspects of the visualized data.



**Figure 5.9: Collapsible user interface in the visualization**

The Bus and Line tabs allow the user to select aspects of the visualization related to the buses and lines. The user can visualize any numerical quantities associated with each line or bus by selecting them in the dropdown menus. Once a new attribute is selected, the color of the lines and the bus volumes will automatically change to suit the behavior of the attributes over time. For example, if users want to make the buses more transparent, they can use the opacity slider or type a number into the box to set the opacity of all currently visible buses.

If users want to hide all buses or lines, they can simply click the “Hide—Show” button. This action will hide the appropriate elements, and when pressed again the buses or lines will reappear.

The section labeled “Slice Selector” allows users to select a horizontal slice of the volumes presented above. They can either type in the time value or use the slider to update the slice in real time. The slice is drawn in bold to stand out against the other drawn elements. The selection shows the state of the system at the specified time across all of the lines and buses within the system. In the case pictured in Figure 5.8, the system is stable (line are grey) across all lines except one, which clearly stands out in red.

The user can scrub or quickly scroll through the timeline, which will cause the slice to animate. All of this interaction can be viewed at almost any angle and zoom with whichever volumes desired displayed to the screen.

## **5.6 Complementary 2D Power System Visualization Prototype**

The proposed 3D navigational visualization is powerful for high-level overview and exploration of the evolution of a power system, but it is insufficient for inspecting the details of the system. In order to support the detailed readings of various systems, we designed a complementary 2D visualization interface to work alongside the 3D overview visualization.

### **5.6.1 Implementation Overview**

Our visualization interface is implemented with D3.js, a popular JavaScript-based visualization toolkit. Like WebGL, this interface also works on any modern Web browser (e.g., Google Chrome, Firefox, etc.). As previously mentioned, web browser-based systems are multi-platform and thus can run on many operating systems, such as Windows, Mac, Linux, and even tablets or smartphones.

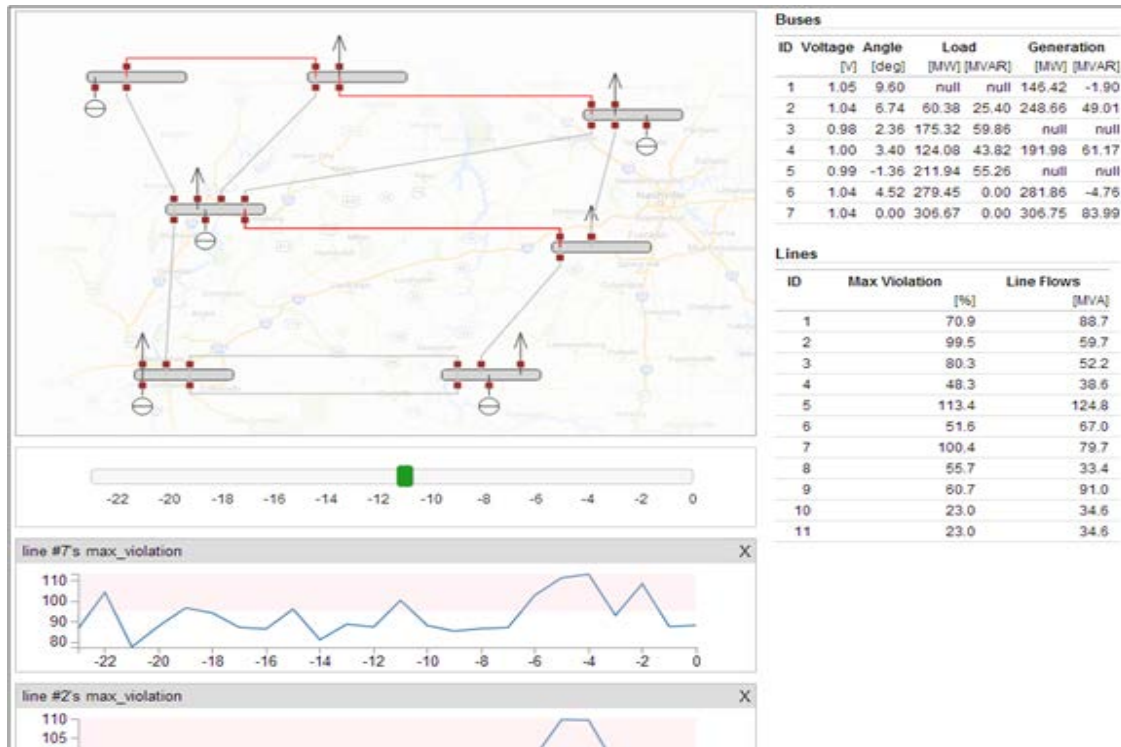
The 2D interface needs to communicate with a server to interactively get the data from our databases. Currently, we are using web.py, a Python-based web framework, to support this communication. Using this framework, the client-side, which is in charge of visualization, can easily request the necessary visualization data from the server. Then the server-side program can load the requested data from our databases and deliver the information to the client-side so that the client-side can process the data and then visualize the data.

### **5.6.2 Interfaces**

This interface consists of three coordinated views which enable analysts to see multiple aspects of the system in great detail. The three views are: 1) the main power system view, 2) line charts, and 3) detail tables. Each view is explained in greater detail below.

### **5.6.3 Main Power System View**

The main power system view is shown in Figure 5.10. The visualized system has 7 buses and 11 transmission lines. Each bus has a generator or load connected with breakers. There is a geographic map in the background to help the user physically locate the equipment.



**Figure 5.10: Main power system view**

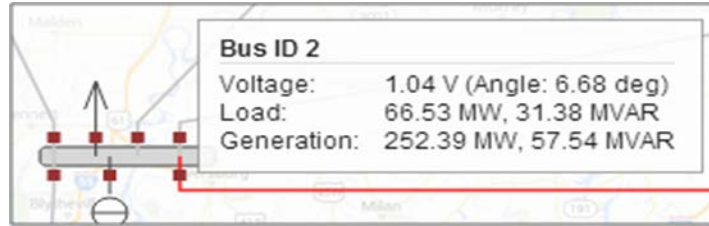
The color of the transmission lines and buses represents their values in terms of maximum line violation or bus voltage. We discretized the values into three categories:

- Gray – normal condition; no action required.
- Orange –warning; may pose an issue further on.
- Red – severe; immediate action needed.

Green was avoided as the default color for normal operating condition since red-green color blindness is very prevalent. This color mapping helps focus the attention of the users and aids in the immediate perception of buses or lines that are threatened.

The slider located right below the network view allows users to change the selected timestamp. Whenever users move the slider, the data is dynamically updated. The colors of the elements or numbers in the detailed table are also automatically updated.

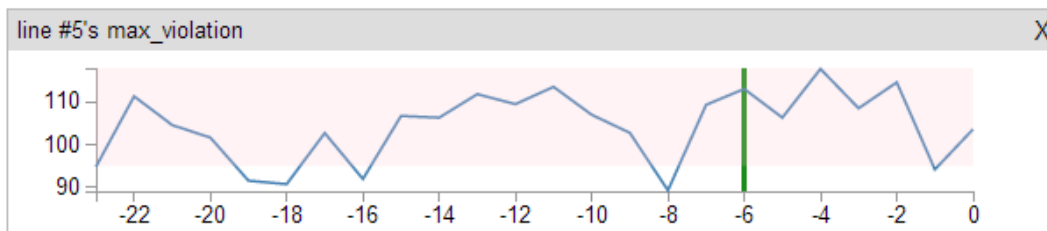
Labels pop up when users hover their mouse over a bus or line, as depicted in Figure 5.11. The label shows the relevant information associated with the bus or line for the selected timestamp. The bus label contains the bus voltage magnitude and angle, as well as the generation and load level. The line label contains the real power and reactive power line flow, as well as the percent thermal violation.



**Figure 5.11: Sample pop-up label**

#### 5.6.4 Line Charts

The line charts located at the bottom of the screen show the trends of selected buses or lines over time. For example, Figure 5.12 shows how the maximum thermal violation for line 5 has changed over time. The x-axis represents time. In this scenario, 24 time steps were simulated. The y-axis represents the value of the selected quantity. The background color for normal operation is white, while the background color for abnormal operation is light red, in this instance for values higher than 95%.



**Figure 5.12: Sample line chart**

Users can easily add or remove line charts from the interface. By clicking the buses or lines on the main network view, the line chart will be added to the interface. This feature allows users to easily compare the characteristic behaviors of several different elements in the same interface. Users can remove a line chart from the interface by clicking on the 'X' button in the top right corner.

#### 5.6.5 Detail Tables

The detail tables show all of the data for the buses and lines at the current selected time. An example is illustrated in Figure 5.13. Each bus has six attributes: voltage, angle, real and reactive generation, as well as real and reactive load. Each transmission line has two attributes: max violation and apparent power flow in each line.

With these tables, users can compare the information for all of the buses and lines at once. The comparison is not difficult in our test system of only 7 buses and 11 lines. However, if we had a

much larger system with thousands of buses and lines, we can provide filtering or sorting to help users explore the large dataset more easily.

Buses						
ID	Voltage	Angle	Load		Generation	
	[V]	[deg]	[MW]	[MVAR]	[MW]	[MVAR]
1	1.05	9.60	null	null	148.34	-1.95
2	1.04	6.68	66.53	31.38	252.39	57.54
3	0.98	2.35	175.60	64.37	null	null
4	1.00	3.38	126.27	42.74	194.83	64.10
5	0.99	-1.35	211.82	59.21	null	null
6	1.04	4.44	343.57	0.00	344.39	-4.31
7	1.04	0.00	308.70	0.00	310.42	85.72

Lines		
ID	Max Violation	Line Flows
	[%]	[MVA]
1	72.4	90.4
2	99.9	60.0
3	80.3	52.2
4	47.9	38.4

Figure 5.13: Detailed table view

## 6 Conclusions

We proposed a new framework for power system decision-making, integrating the concepts of a unified geospatial model, smart contingency analysis, and look-ahead visualization. A unified network model for power system operations and planning was identified as the requirement for seamless multi-scenario security assessment. Two multi-scenario contingency analysis techniques were introduced. GPU computing was proposed to further speed up contingency analysis. Preliminary results for the GPU matrix multiplication operation were presented. In addition, a new seamless spatio-temporal visualization designed for operators, planners, and managers was discussed.

### 6.1 Unified Geospatial Model

The unified network applications framework is extended to the study of contingencies that result in bus splits and bus mergers. Such contingencies are not realizable using a bus-branch model alone. Dynamic pointer assignment and incremental subnet processing allow seamless realization of consolidated representations necessary to model arbitrary post-contingency topologies. The consolidated representation obtained in this manner is electrically identical to the bus-branch model that would correspond to the real-time system given the post-contingency switching device statuses.

Numerical simulations were performed using a large scale ISO node-breaker model to quantitatively determine the advantages of the proposed framework:

- a) The method results in the same numerical results as the EMS security assessment solutions,
- b) Numerical instability problems that occur when solving cases with breakers modeled as low-impedance branches are resolved
- c) The complex and cumbersome procedure which involves manual inspection of the node-breaker model and script development needed to modify an existing bus-branch model to model switching contingencies is no longer necessary, and
- d) The method enables direct comparison of real-time and off-line contingency results, achieving full interoperability between planning and EMS security assessment models and applications.

### 6.2 Multi-Scenario Contingency Analysis

The concept of using distribution factors to directly transition from the power flow solution of one system state to the contingency analysis results for a similar but different state was introduced. A new time-dependent PTDF was defined and combined with OTDFs to estimate post-contingency

transmission line flows for a scenario that deviated slightly from the base case. Representative results for an illustrative 7-bus example and the IEEE 24-bus reliability test system were presented and compared against traditional distribution factor-based contingency analysis.

Similar to the conventional DC method, this approach is also based on bus injections and can be used to represent different types of contingencies. Its accuracy was comparable to the accuracy of traditional DC contingency analysis for small system changes, but there is a slight loss of accuracy for significant deviations such as large changes in injection. Intuitively, the accuracy of the new method depends on the relative size of the deviation from the reference system state. Based on the computational complexity analysis, this method has the potential to be faster than traditional DC contingency analysis. Further improvement may be possible with algorithmic optimization.

An alternate perturbative approach was also discussed. Timing results for a 7-bus example, the IEEE 24-bus reliability test system, and the IEEE 118-bus test system were presented.

### **6.3 Navigational Visualization**

The need for better methods of power system visualization was motivated, and a novel 3D navigational method of presentation was designed for the exploration of past, present, and future power system states. The 3D stacking approach was illustrated on the PowerWorld 7-bus test system. Some issues with visualizing small changes or deviations in the dataset were discussed. The user interaction, an important component of our design, has been developed to include five different features. A collapsible user interface was also added to enhance the user experience and allows users some level of customization.

As a complement to the 3D overview visualization, a 2D visualization interface that shows more detailed information was also developed. Similar to other power system visualization tools, our interface included a system one-line diagram. Color was used to emphasize components that are threatened while the components under normal operation were grayed out. The user interaction was especially designed with detailed navigation and exploration of system changes over time in mind. Interactive pop-up bus labels, line charts, and detailed tabular views are all essential to achieve those goals.

## 7 Future Work

### 7.1 Multi-Scenario Contingency Analysis

Possible future work in this area includes the development of cross-temporal and cross-scenario voltage sensitivities for systems that consistently encounter undervoltage or overvoltage issues. Additionally this method may be extended from  $N-1$  to  $N-k$  contingency analysis. Also, an in-depth analysis of the method's accuracy would be of interest to operators and planners.

### 7.2 Navigational Visualization

Future work in this area should be focused on finding a specific application that would benefit from this type of 3D visualization. One possible application is to use this 3D technique to visualize systems with large penetrations of non-dispatchable renewables. The forecast data can be updated in blocks dynamically. When wind or solar production drops suddenly, the tool should be able to automatically alert operators so that they can take immediate action. Of even greater interest is if the tool could suggest optimal solutions that operators could take to resolve flagged issues.

Another possible extension of our work is to visualize contingency analysis results. After reviewing more literature on existing contingency analysis visualization techniques, we have refined our view of this project's visualization system requirements. From the literature review, we highlight three main findings: 1) the traditional technique used to visualize contingency analysis results is tabular, where each row in the table displays information about a given contingency and its consequences on some system security parameters, which is not suitable for large amounts of data and overloads the analyst with information that is difficult to navigate through; 2) in recent years, power researchers have proposed more advanced 2-D and 3-D visualization techniques that take into account the geographical coordinates of the power system's constituents, and present the large amount of contingency analysis data in a summarized fashion using color-coding, shapes, visual effects (e.g., transparency levels), etc.; 3) to our knowledge, no previous work has addressed our visualization setting, in which the contingency analysis is extended over multiple time steps, as opposed to a static single-step setting as in the literature review. This makes our problem more challenging, yet interesting. To address this, we could design a visualization system with two main components as illustrated in Figure 7.1:

1. Temporal scenario tree: this is a simple tree in which each level represents one time step, each node at level  $i$  represents a possible system state (or a scenario) at time  $i$ , and each link (or edge) between a node  $u_i$  at time  $i$  and another (child) node  $u_{i+1}$  at time  $i + 1$  represents the a given control action that, if taken, may shift the system state from the configuration in  $u_i$  to that of  $u_{i+1}$ . Each node in the temporal scenario tree will be clickable, and the clicked node will become the configuration of interest.

2. Contingency visualization deck: the main component of the visualization system, where the results of the contingency analysis are displayed, with a central view on the system state corresponding to the node that is clicked on in the temporal scenario tree described above. We are currently considering two possible options for this view:
  - a. Aggregate view: the power system is drawn according to the traditional one-line technique. To account for the temporal scale in our contingency data, each element of the system is coupled with a tower of disks, visualized in 3-D, where each disk corresponds to the value of a certain parameter (e.g., voltage on a bus) at a certain time step. This approach aggregates all contingency analysis results across time steps in a single 3-D visualization.
  - b. Timeline view: the contingency visualization deck is split into three main parts:
    - i. The current state: this is the system state corresponding to the node that is clicked in the temporal scenario tree. It is visualized at the middle of the deck with a frame that highlights its centrality to the user or analyst.
    - ii. The parent states: this is the set of system states that lead to the current state (i.e. the set of states corresponding to the nodes on the path from the root of the temporal scenario tree, to the current state's node).
    - iii. The descendant states: this is the set of system states resulting from the current states (i.e. the states of all the nodes in the temporal scenario tree that are reachable from the current state's node).

In this approach, the user can focus on the main state he is interested in analyzing, while also being able to navigate through the states that have led to it at previous time steps, or that result from it at later time steps. Drop-down menus can be used to filter the candidate parent/descendant states into a smaller subset that satisfies certain conditions specified by the user, e.g., filtering on descendant states of our current state that have very low voltages or very high overloads.

In any choice of approach for the visualization deck, this system will be reading the results of the contingency analysis from our database, described earlier in this document. The database has been designed and implemented with the temporal data extraction use case requirement specification, and hence queries necessary for visual rendering are expected to be fast and efficient.

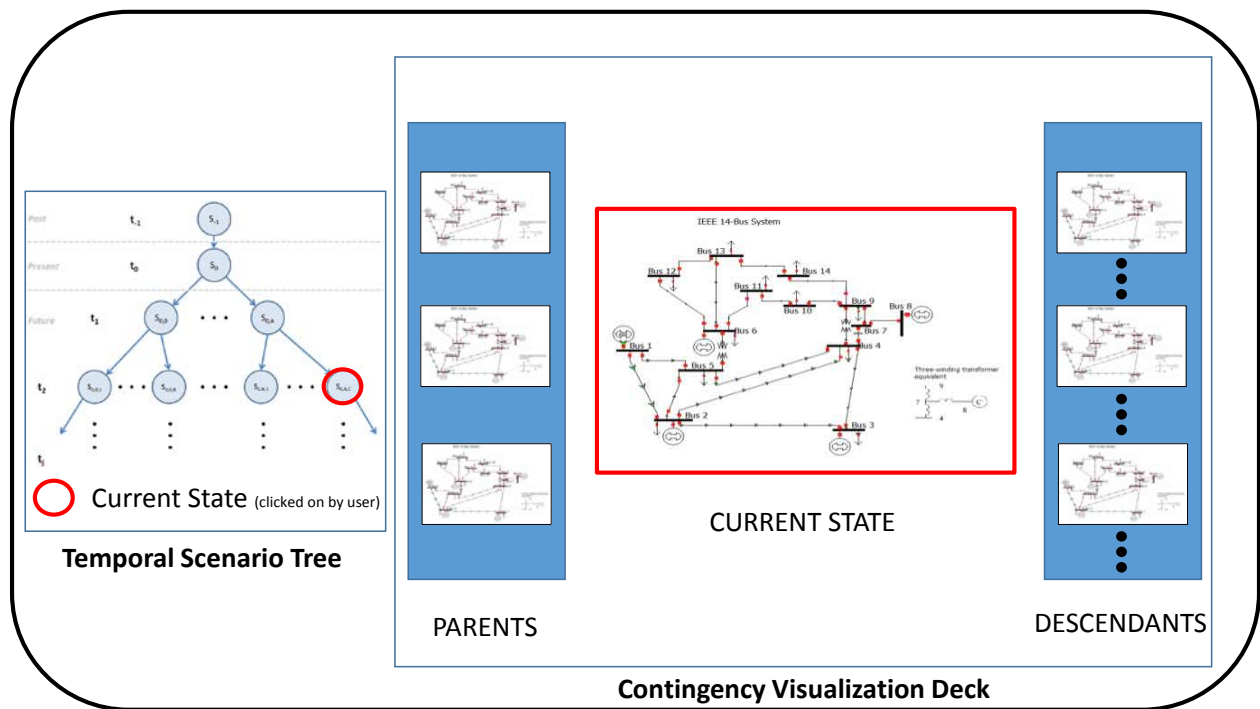


Figure 7.1: Schema of the “diagram view” approach to visualizing contingency analysis results

## 8 References

- [1] S. Grijalva, "Integrating Real-Time Operations and Planning using Same-Format Power System Models," in *Power Engineering Society General Meeting, 2007. IEEE*, 2007, pp. 1-6.
- [2] S. Grijalva and A. Roy, "Automated Handling of Arbitrary Switching Device Topologies in Planning Contingency Analysis: Towards Temporal Interoperability in Network Security Assessment," *Power Systems, IEEE Transactions on*, vol. PP, pp. 1-1, 2012.
- [3] S. Grijalva, "Direct utilization of planning applications in the real-time environment," in *Transmission and Distribution Conference and Exposition, 2008. T&D. IEEE/PES*, 2008, pp. 1-6.
- [4] R. Billinton and Y. Hua, "Incorporating maintenance outage effects in substation and switching station reliability studies," in *Electrical and Computer Engineering, 2005. Canadian Conference on*, 2005, pp. 599-602.
- [5] S. Su, K. K. Li, W. L. Chan, X. Zeng, and X. Li, "Using substation automation information for electric power load modeling and predictive maintenance of circuit breaker," in *Industrial Informatics, 2005. INDIN '05. 2005 3rd IEEE International Conference on*, 2005, pp. 546-551.
- [6] J. J. Meeuwsen and W. L. Kling, "Substation reliability evaluation including switching actions with redundant components," *Power Delivery, IEEE Transactions on*, vol. 12, pp. 1472-1479, 1997.
- [7] X. Xu, B. P. Lam, R. R. Austria, Z. Ma, Z. Zhu, R. Zhu, and J. Hu, "Assessing the impact of substation-related outages on the network reliability," in *Power System Technology, 2002. Proceedings. PowerCon 2002. International Conference on*, 2002, pp. 844-848 vol.2.
- [8] A. K. Kazerooni and J. Mutale, "Flexible transmission network planning with post-contingency network switching," in *Universities Power Engineering Conference (UPEC), 2009 Proceedings of the 44th International*, 2009, pp. 1-5.
- [9] P. R. Ribeiro Jr., "Power flow at networks modeled at substation level," Master's thesis, E.E. Graduate Program, Univ. Federal do Parana, Curitiba, Parana, Brazil, 2005.
- [10] E. M. Lourenco, A. S. Costa, and R. Ribeiro P, "Steady-State Solution for Power Networks Modeled at Bus Section Level," *Power Systems, IEEE Transactions on*, vol. 25, pp. 10-20, 2010.
- [11] M. P. Selvan and K. S. Swarup. (Jan-Feb 2005) Object methodology: method and design for topological processing. *IEEE Power and Energy Magazine*. 18-29.
- [12] A. Abur and A. Gomez Exposito, "Power system state estimation," in *Marcel Dekker*, ed, 2004.
- [13] L. Jinsong, L. Xiaolu, L. Dong, L. Hesun, and M. Peng, "Study on Data Management of Fundamental Model in Control Center for Smart Grid Operation," *Smart Grid, IEEE Transactions on*, vol. 2, pp. 573-579, 2011.
- [14] D. J. Tylavsky, P. E. Crouch, L. F. Jarriel, J. Singh, and R. Adapa, "The effects of precision and small impedance branches on power flow robustness," *Power Systems, IEEE Transactions on*, vol. 9, pp. 6-14, 1994.

- [15] A. Monticelli, "The impact of modeling short circuit branches in state estimation," *Power Systems, IEEE Transactions on*, vol. 8, pp. 364-370, 1993.
- [16] A. Monticelli and A. Garcia, "Modeling zero impedance branches in power system state estimation," *Power Systems, IEEE Transactions on*, vol. 6, pp. 1561-1570, 1991.
- [17] O. Alsac, N. Vempati, B. Stott, and A. Monticelli, "Generalized state estimation," *Power Systems, IEEE Transactions on*, vol. 13, pp. 1069-1075, 1998.
- [18] A. Gomez-Exposito and A. de la Villa Jaen, "Reduced substation models for generalized state estimation," *Power Systems, IEEE Transactions on*, vol. 16, pp. 839-846, 2001.
- [19] W. F. Tinney and C. E. Hart, "Power Flow Solution by Newton's Method," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-86, pp. 1449-1460, 1967.
- [20] B. Stott and O. Alsac, "Fast Decoupled Load Flow," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-93, pp. 859-869, 1974.
- [21] A. J. Wood and B. F. Wollenberg, *Power generation, operation, and control*, 2 ed.: Wiley-Interscience, 1996.
- [22] V. Brandwajn, "Efficient bounding method for linear contingency analysis," *Power Systems, IEEE Transactions on*, vol. 3, pp. 38-43, 1988.
- [23] P. A. Ruiz and P. W. Sauer, "Voltage and Reactive Power Estimation for Contingency Analysis Using Sensitivities," *Power Systems, IEEE Transactions on*, vol. 22, pp. 639-647, 2007.
- [24] R. C. Green, L. Wang, and M. Alam, "Applications and Trends of High Performance Computing for Electric Power Systems: Focusing on Smart Grid," *Smart Grid, IEEE Transactions on*, vol. 4, pp. 922-931, 2013.
- [25] C. Hsiao-Dong, C. S. Wang, and A. J. Flueck, "Look-ahead voltage and load margin contingency selection functions for large-scale power systems," *Power Systems, IEEE Transactions on*, vol. 12, pp. 173-180, 1997.
- [26] G. C. Ejebe and B. F. Wollenberg, "Automatic Contingency Selection," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-98, pp. 97-109, 1979.
- [27] E. G. Preston, M. L. Baughman, and W. M. Grady, "A new model for outaging transmission lines in large electric networks," *Power Systems, IEEE Transactions on*, vol. 14, pp. 412-418, 1999.
- [28] G. Jiachun, F. Yong, L. Zuyi, and M. Shahidehpour, "Direct Calculation of Line Outage Distribution Factors," *Power Systems, IEEE Transactions on*, vol. 24, pp. 1633-1634, 2009.
- [29] T. Guler, G. Gross, and L. Minghai, "Generalized Line Outage Distribution Factors," *Power Systems, IEEE Transactions on*, vol. 22, pp. 879-881, 2007.
- [30] PowerWorld, "PowerWorld Simulator Version 16," ed, 2012.
- [31] C. Grigg, P. Wong, P. Albrecht, R. Allan, M. Bhavaraju, R. Billinton, Q. Chen, C. Fong, S. Haddad, S. Kuruganty, W. Li, R. Mukerji, D. Patton, N. Rau, D. Reppen, A. Schneider, M. Shahidehpour, and C. Singh, "The IEEE Reliability Test System-1996. A report prepared by the Reliability Test System Task Force of the Application of Probability Methods Subcommittee," *Power Systems, IEEE Transactions on*, vol. 14, pp. 1010-1020, 1999.

- [32] F. L. Alvarado, "Computational complexity in power systems," *Power Apparatus and Systems, IEEE Transactions on*, vol. 95, pp. 1028-1037, 1976.
- [33] T. J. Overbye, D. A. Wiegmann, and R. J. Thomas, "Visualization of Power Systems," PSERC Publication 02-36 November 2002.
- [34] T. J. Overbye, "Visualization enhancements for power system situational assessment," in *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*, 2008, pp. 1-4.
- [35] J. D. Weber and T. J. Overbye, "Voltage contours for power system visualization," *Power Systems, IEEE Transactions on*, vol. 15, pp. 404-409, 2000.
- [36] Y. Sun and T. J. Overbye, "Visualizations for power system contingency analysis data," *Power Systems, IEEE Transactions on*, vol. 19, pp. 1859-1866, 2004.
- [37] T. J. Overbye and J. D. Weber, "Visualization of power system data," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, 2000, p. 7 pp.
- [38] M. Farrugia and A. Quigley, "Effective temporal graph layout: a comparative study of animation versus static display methods," *Information Visualization*, vol. 10, pp. 47-64, 2011.
- [39] S. Rufiange and M. J. McGuffin, "DiffAni: Visualizing Dynamic Graphs with a Hybrid of Difference Maps and Animation," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 19, pp. 2556-2565, 2013.
- [40] B. Bach, E. Pietriga, and J. D. Fekete, "GraphDiaries: Animated Transitions and Temporal Navigation for Dynamic Networks," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 20, pp. 740-754, 2014.
- [41] M. Itoh, M. Toyoda, and M. Kitsuregawa, "An Interactive Visualization Framework for Time-Series of Web Graphs in a 3D Environment," in *Information Visualisation (IV), 2010 14th International Conference*, 2010, pp. 54-60.
- [42] C. Tominski, H. Schumann, G. Andrienko, and N. Andrienko, "Stacking-Based Visualization of Trajectory Attribute Data," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, pp. 2565-2574, 2012.
- [43] P. Kansal and A. Bose, "Smart grid communication requirements for the high voltage power system," *IEEE PES General Meeting*, pp. 1-6, Jul. 2011.
- [44] A. Bose, "Smart transmission grid applications and their supporting infrastructure," *Smart Grid, IEEE Transactions on*, vol. 1, no. 1, pp. 11-19, 2010.
- [45] D. Tholomier, H. Kang, and B. Cvorovic, "Phasor measurement units: Functionality and applications," *IEEE PES Power Systems Conference and Exhibition*, pp. 1-12, Mar. 2009.
- [46] F. F. Wu, K. Moslehi, and A. Bose, "Power system control centers; past, present and future," *Proc. IEEE*, vol. 93, no. 11, pp. 1890-1908, Nov. 2005.

- [47] M. Chenine, K. Zun, and L. Nordstrom, "Survey on priorities and *communication requirements for pmu-based applications in the Nordic region*," *IEEE Power Tech*, pp. 1–8, Jul. 2009.
- [48] P. Kansal and A. Bose., "Bandwidth and latency requirements for smart transmission grid applications," *IEEE Trans. on Smart Grid*, vol. 3, no. 3, pp. 1344–1352, Sep. 2012.
- [49] "EEE standard for synchrophasors for power systems," *IEEE Std. C37.118-2005 (Revision of IEEE Std 1344-1995)*, pp. 1–57, 2006.
- [50] NASPI. (2009) Actual and potential phasor data applications. [Online] Available: <http://www.naspi.org/phasorappstable.pdf>
- [51] N. Data and N.M.T. Team. (2007) Phasor application classification. [Online]. Available:[http://www.naspi.org/resources/](http://www.naspi.org/resources/dnmtt/phasorapplicationclassification%2020080807.xls)  
[dnmtt/phasorapplicationclassification 20080807.xls](http://www.naspi.org/resources/dnmtt/phasorapplicationclassification%2020080807.xls)
- [52] H. Gjermundrod, D. Bakken, C. Hauser, and A. Bose, "Gridstat: A flexible qos-managed data dissemination framework for the power grid," *Power Delivery, IEEE Transactions on*, vol. 24, no. 1, pp. 136–143, 2009.
- [53] C. Hauser, D. Bakken, and A. Bose, "A failure to communicate: next generation communication requirements, technologies, and architecture for the electric power grid," *Power and Energy Magazine, IEEE*, vol. 3, no. 2, pp. 47–55, 2005.
- [54] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 5th ed. USA: Addison-Wesley Publishing Company, 2009.
- [55] R. W. Floyd, "Algorithm 97: Shortest path," *Commun. ACM*, vol. 5, no. 6, pp. 345–, Jun. 1962.
- [56] W. Jiang, V. Vittal, and G. Heydt, "A distributed state estimator utilizing synchronized phasor measurements," *Power Systems, IEEE Transactions on*, vol. 22, no. 2, pp. 563–571, 2007.
- [57] S. Koch, S. Chatzivasileiadis, M. Vrakopoulou, and G. Andersson, "Mitigation of cascading failures by real-time controlled islanding and graceful load shedding," in *Bulk Power System Dynamics and Control (iREP) - VIII (iREP), 2010 iREP Symposium*, 2010, pp. 1–19.
- [58] P. Anderson and A. Fouad, *Power system control and stability*, ser. Institute of Electrical and Electronics Engineers, IEEE Press power engineering series. IEEE Press, 2003.

- [59] Y. Zhang and A. Bose, "Design of wide-area damping controllers for interarea oscillations," *Power Systems, IEEE Transactions on*, vol. 23, no. 3, pp. 1136–1143, 2008.
- [60] Y. Zhang, "Design of wide-area damping control systems for power system low-frequency inter-area oscillations," PhD Thesis, Washington State University (WSU), Pullman, WA, USA, Dec. 2007.