

8-1-2009

## Distributed simulation of power systems using real time digital simulator

Krishnanjan Gubba Ravikumar

Follow this and additional works at: <https://scholarsjunction.msstate.edu/td>

---

### Recommended Citation

Gubba Ravikumar, Krishnanjan, "Distributed simulation of power systems using real time digital simulator" (2009). *Theses and Dissertations*. 1539.  
<https://scholarsjunction.msstate.edu/td/1539>

This Graduate Thesis - Open Access is brought to you for free and open access by the Theses and Dissertations at Scholars Junction. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Scholars Junction. For more information, please contact [scholcomm@msstate.libanswers.com](mailto:scholcomm@msstate.libanswers.com).

DISTRIBUTED SIMULATION OF POWER SYSTEMS USING  
REAL TIME DIGITAL SIMULATOR

By

Krishnanjan Gubba Ravikumar

A Thesis  
Submitted to the Faculty of  
Mississippi State University  
in Partial Fulfillment of the Requirements  
for the Degree of Master of Science  
in Electrical Engineering  
in the Department of Electrical and Computer Engineering

Mississippi State, Mississippi

August 2009

Copyright by

Krishnanjan Gubba Ravikumar

2009

DISTRIBUTED SIMULATION OF POWER SYSTEMS USING  
REAL TIME DIGITAL SIMULATOR

By

Krishnanjan Gubba Ravikumar

Approved:

---

Noel N. Schulz  
Professor of Electrical and Computer  
Engineering  
(Director of Thesis)

---

Anurag K. Srivastava  
Assistant Research Professor of Electrical  
and Computer Engineering.  
(Co Major Advisor)

---

Suresh C Srivastava  
Visiting Research Professor of Electrical  
and Computer Engineering  
(Committee Member)

---

James E. Fowler  
Professor of Electrical and Computer  
Engineering  
(Graduate Coordinator)

---

Sarah A. Rajala  
Dean of College of Engineering

Name: Krishnanjan Gubba Ravikumar

Date of Degree: August 8, 2009

Institution: Mississippi State University

Major Field: Electrical Engineering

Major Professor: Dr. Noel N. Schulz

Title of Study: DISTRIBUTED SIMULATION OF POWER SYSTEMS USING REAL  
TIME DIGITAL SIMULATOR

Pages in Study: 129

Candidate for Degree of Master of Science

The simulation of power system behavior, especially transient behavior, helps us in the analysis and planning of various power systems. However, power systems are highly complex and geographically distributed. Therefore system partitioning can be used to allow for sharing resources in simulation. In this work, distributed simulations of power system models have been developed using an electromagnetic transient simulator, namely Real Time Digital Simulator. The goal is to demonstrate and assess the feasibility of both non-real-time and real-time simulations using RTDS in a geographically distributed scenario. Different protocols and options used in the communication between power systems have been studied and analyzed. In this work, a test bed has been developed for data transfer between a power system simulated in RTDS at Mississippi State University and the power system simulated in RTDS at Texas A&M University. Finally, a locally distributed wide area control test bed was also developed and simulated.

## DEDICATION

This thesis work is dedicated to my parents, my best friend Indira and my advisor.

## ACKNOWLEDGMENTS

I would like to express my whole hearted thanks to all the people who were with me during all the times making this research possible. I would like to gratefully thank my advisor Dr. Noel Schulz for providing me this opportunity with lots of guidance, support, encouragement and assistance. Her positive attitude and willingness to help brought out the best in me. I would also like to thank Dr. Anurag K. Srivastava for his valuable guidance, interest and enthusiasm in making my research life smooth and rewarding. These were the people who really taught me how to be a professional and a good power engineer. They positively influenced my life here at Mississippi State University very much by providing me all the facilities and the opportunities to attend so many conferences and technical meetings. Without their constant guidance and support, I could have not gained the useful knowledge which I possess now and it would have been not possible to complete this thesis.

I would also like to thank Dr. Jimena L. Bastos for her guidance and belief in me. I am very grateful to Dr. Suresh C. Srivastava for his willingness to serve on my committee and his support to the whole power engineering group.

I would also thank the team of RTDS Inc. for helping me understand the complete operations and architecture of the Real Time Digital Simulator. I extend my special thanks to the National Instruments technical support. I greatly appreciate Mr. Greg

Zweigle of Schweitzer Engineering Laboratories Inc. for providing me the power system hardware and for donating the SEL 3378 (Synchrophasor Vector Processor) to the power group at MSU. I sincerely appreciate Mississippi State University, the Department of Defense and DURIP for their financial support during my thesis work.

Finally, last but not the least, special thanks to my colleagues here at Mississippi State University, Ms. Barbara Matta, my parents and my best friend Indira for their continuous love, help and support during my studies and research.



## TABLE OF CONTENTS

|   | Page |
|---|------|
| DEDICATION .....  | ii   |
| ACKNOWLEDGMENTS .....   | iii  |
| LIST OF TABLES .....  | viii |
| LIST OF FIGURES .....   | ix   |
| LIST OF ABBREVIATIONS.....  | xiv  |
| <br>CHAPTER   |      |
| I. INTRODUCTION .....   | 1    |
| 1.1 Introduction.....   | 1    |
| 1.2 Overview of Distributed Simulation.....   | 3    |
| 1.2.1 Geographically distributed simulation.....  | 3    |
| 1.2.2 Local distributed simulation .....  | 4    |
| 1.2.3 Local and geographically distributed simulation.....                                | 4    |
| 1.3 Overview of Wide Area Monitoring and Control.....                                     | 4    |
| 1.4 Objectives .....  | 5    |
| 1.5 Contributions of this work .....  | 6    |
| 1.6 Thesis Outline .....  | 7    |
| II. BACKGROUND AND LITERATURE REVIEW .....  | 9    |
| 2.1 Introduction.....   | 9    |
| 2.2 Background.....   | 9    |
| 2.2.1 Department of Defense (DoD) Multiple University<br>Research Initiative (MURI) ..... | 10   |
| 2.2.2 MURI Implementation and Network Capability .....                                    | 12   |
| 2.3 Distributed Simulation .....  | 14   |
| 2.3.1 Parallel algorithm and different kinds of coupling<br>Patterns.....                 | 17   |

|   |    |
|---|----|
| 2.4 Coordinated Distributed Control.....  | 21 |
| 2.4.1 Wide Area Monitoring and Control.....   | 21 |
| 2.5 Tools .....   | 23 |
| 2.5.1 Real Time Digital Simulator.....  | 23 |
| 2.5.2 National Instruments-PXI Controller.....  | 26 |
| 2.5.3 SEL 421 Protection, Automation and Control System .....                           | 30 |
| 2.5.4 SEL 3378 Synchrophasor Vector Processor.....                                      | 31 |
| 2.5.5 RSCAD Software Suite.....   | 32 |
| 2.5.6 National Instruments LabVIEW .....  | 34 |
| 2.5.7 NI Measurement and Automation Explorer.....                                       | 34 |
| 2.5.8 National Instruments LabVIEW Real Time .....                                      | 35 |
| 2.5.9 Wireshark.....  | 38 |
| 2.6 Summary .....   | 38 |
| III. PROBLEM STATEMENT AND PROPOSED SOLUTION.....                                       | 40 |
| 3.1 Introduction.....   | 40 |
| 3.2 Problem Description .....   | 40 |
| 3.3 Proposed Work.....  | 41 |
| 3.4 Approaches to Problem.....  | 42 |
| 3.5 Work Plan .....   | 49 |
| 3.6 Summary .....   | 50 |
| IV. RESEARCH CHALLENGES AND ANALYSIS .....  | 51 |
| 4.1 Analysis of Available protocols in the RTDS.....                                    | 51 |
| 4.1.1 Analysis of sending the data out of RTDS.....                                     | 51 |
| 4.1.2 Analysis of getting the data into RTDS .....                                      | 56 |
| 4.1.2.1 Using GTNET Playback Firmware.....  | 56 |
| 4.1.2.2 Using National Instruments PXI .....  | 57 |
| 4.2 Interface Algorithm and Numerical Methods .....                                     | 58 |
| 4.2.1 Numerical Solutions.....  | 59 |
| 4.3 Challenges Involved.....  | 63 |
| 4.3.1 High output rate of GTNET SV and LabVIEW modeling .....                           | 64 |
| 4.3.2 Time stamping and synchronization of simulators .....                             | 64 |
| 4.3.3 Transmission line model .....   | 67 |
| 4.4 Summary .....   | 67 |
| V. DEVELOPMENT OF A GEOGRAPHICALLY DISTRIBUTED<br>TESTBED WITH SIMULATION RESULTS ..... | 68 |
| 5.1 Introduction.....   | 68 |
| 5.2 Schematics of Data Operations .....   | 68 |
| 5.3 Geographically Distributed Test Bed Schematic.....                                  | 71 |
| 5.4 LabVIEW modeling .....  | 74 |
| 5.5 Experiments and Data Rates .....  | 84 |

|   |     |
|---|-----|
| 5.5.1 Data transfer test using three-bus power system test case .....                         | 85  |
| 5.5.2 Remote Monitoring experiments using RTDS .....  | 87  |
| 5.5.3 Calculation of Data rates .....   | 93  |
| 5.6 Summary .....   | 100 |
| VI. DEVELOPMENT OF WIDE AREA MONITORING AND<br>CONTROL TEST BED WITH SIMULATION RESULTS ..... | 101 |
| 6.1 Introduction.....   | 101 |
| 6.2 Wide Area Monitoring and Control (WAMCS) .....  | 101 |
| 6.3 Transient Stability Enhancement. ....   | 102 |
| 6.4 Test System Modeling and Control Algorithm.....   | 102 |
| 6.5 Simulation Results for Wide Area Control Scheme .....                                     | 106 |
| 6.6 Analysis of Distributed Simulation Locally with Simulation<br>Results.....                | 112 |
| 6.7 Discussions .....   | 120 |
| 6.8 Summary .....   | 122 |
| VII. CONCLUSIONS AND FUTURE WORK .....  | 123 |
| 7.1 Conclusions.....  | 123 |
| 7.2 Future Work .....   | 125 |
| REFERENCES .....  | 126 |

## LIST OF TABLES

| TABLE  | Page |
|--|------|
| 2.1 List of hardwares used .....   | 39   |
| 2.2 List of softwares used .....   | 39   |
| 4.1 External 1 PPS signal for GTNET accuracy testing (from GPS [SEL-2407]).....        | 66   |
| 4.2 Internal 1 PPS signal for GTNET accuracy testing (generated within RTDS).....      | 66   |
| 5.1 Results of Data transfer Experiment 1 using C#.....                                | 94   |
| 5.2 Results of Data transfer Experiment 2 using C#.....                                | 96   |
| 5.3 Results of Data transfer Experiment 3 using C#.....                                | 97   |
| 5.4 Results of Data transfer Experiment for 118 bytes (When MSU RTDS was Client) ..... | 98   |
| 5.5 Results of Data transfer Experiment for 118 bytes (When MSU RTDS was Server) ..... | 98   |
| 5.6 Results of Data transfer Experiment for 36 bytes (When MSU RTDS was Client) .....  | 99   |
| 5.7 Results of Data transfer Experiment for 36 bytes (When MSU RTDS was Server) .....  | 99   |

## LIST OF FIGURES

| FIGURE   | Page |
|--|------|
| 1.1 Architecture of a wide area monitoring, control and protection system [3]..... | 5    |
| 2.1 Remote testing and Measurement (RTM) Setup between Universities.....           | 11   |
| 2.2 Structure of remote Hardware/Hardware Implementation [4].....                  | 12   |
| 2.3 Conceptual Topology of Distributed Simulation [8] .....                        | 15   |
| 2.4 A prototype of the power system before it is distributed [4] .....             | 16   |
| 2.5 A prototype of the power system after it is distributed [4] .....              | 16   |
| 2.6 Basic Coupling Pattern: V-type Coupling Scheme [14] .....                      | 19   |
| 2.7 Basic Coupling Pattern: I-type Coupling Scheme [14].....                       | 19   |
| 2.8 Basic Coupling Pattern: VI-type coupling scheme [14] .....                     | 20   |
| 2.9 Wide Area Control Framework [17].....  | 23   |
| 2.10 Real Time Digital Simulator at MSU .....                                      | 25   |
| 2.11 Giga Transceiver Network Communication Card [18].....                         | 26   |
| 2.12 NI PXI System with an 8 slot Chassis [20].....                                | 27   |
| 2.13 NI PXI 8196 Controller [21].....  | 28   |
| 2.14 NI PXI 6733 Analog Output Device [19] .....                                   | 29   |
| 2.15 NI Shielded Connector Block-68 [19] .....                                     | 29   |
| 2.16 SEL 421 Protection, Automation and Control System [22] .....                  | 30   |
| 2.17 Operation of SEL 3378 with the Power System [22] .....                        | 32   |

|      |  |    |
|------|--|----|
| 2.18 | SEL 3378 Synchrophasor Vector Processor [22] .....                         | 32 |
| 2.19 | RSCAD Architecture [18].....   | 33 |
| 2.20 | NI Measurement and Automation Explorer.....                                | 36 |
| 2.21 | Configuration of Measurement and Automation Explorer [21] .....            | 37 |
| 3.1  | Trapezoidal Rule [1] .....   | 43 |
| 3.2  | Representation of Resistor .....   | 44 |
| 3.3  | Representation of Inductor.....  | 44 |
| 3.4  | Norton Equivalent of an Inductor [1].....                                  | 46 |
| 3.5  | Representation of Capacitor.....   | 47 |
| 3.6  | Norton Equivalent of Capacitor [1] .....                                   | 48 |
| 3.7  | Reduction of RLC branch to a Norton Equivalent [1] .....                   | 49 |
| 4.1  | GTNET Sampled Values Component in RSCAD [18].....                          | 55 |
| 4.2  | GTNET Playback for getting the data into RTDS .....                        | 56 |
| 4.3  | NI PXI for getting the data into RTDS .....                                | 57 |
| 4.4  | Interface Algorithm for Distributed Simulation using Predictor Method..... | 59 |
| 5.1  | Sending the data out of RTDS into WAN using GTNET and LabVIEW .....        | 69 |
| 5.2  | Getting the data into RTDS from WAN using NI PXI and OADC.....             | 71 |
| 5.3  | Proposed Schematic for RTDS-RTDS communication.....                        | 72 |
| 5.4  | Three Phase Current Injection Component in RSCAD .....                     | 73 |
| 5.5  | CC type Three Phase Voltage Source in RSCAD .....                          | 74 |
| 5.6  | LabVIEW Server Model with data capture Algorithm.....                      | 76 |
| 5.7  | Block Diagram of LabVIEW Server Model .....                                | 77 |
| 5.8  | Deployment Status Error because of third party application .....           | 79 |
| 5.9  | Bad Imports of LabVIEW RT DLL Checker .....                                | 79 |
| 5.10 | Stubbed Imports of LabVIEW RT DLL Checker.....                             | 80 |

|      |   |     |
|------|---|-----|
| 5.11 | LabVIEW Client Model with data Parsing Algorithm .....  | 81  |
| 5.12 | Block Diagram of LabVIEW Client Model.....  | 82  |
| 5.13 | Remote monitoring experiments using LabVIEW programming platform.....   | 84  |
| 5.14 | Remote monitoring experiments using C# programming platform.....  | 84  |
| 5.15 | Three Bus power system test case in RTDS .....  | 85  |
| 5.16 | General Configuration for GTNET block in RSCAD .....  | 86  |
| 5.17 | Instantaneous Line Currents in Rack 1 during steady state condition.....  | 87  |
| 5.18 | RMS Node voltages in Rack 1 during steady state condition.....  | 88  |
| 5.19 | Line Currents and Node voltages in LabVIEW client during steady state<br>condition .....  | 89  |
| 5.20 | Instantaneous Line Currents in Rack 2 during steady state Condition .....   | 90  |
| 5.21 | RMS Node voltages in Rack 2 during steady state Condition.....  | 90  |
| 5.22 | Instantaneous Line Currents in Rack 1 during Fault Condition .....  | 91  |
| 5.23 | RMS node voltages in Rack 1 during Fault Condition .....  | 91  |
| 5.24 | Line Currents and Node voltages in LabVIEW client during fault<br>condition .....   | 92  |
| 5.25 | Instantaneous Line Currents in Rack 2 during Fault Condition .....  | 93  |
| 5.26 | RMS Node voltages in Rack 2 during Fault Condition .....  | 93  |
| 6.1  | Test Case used for Wide Area Protection Scheme .....  | 103 |
| 6.2  | Test Case Modeling in RSCAD.....  | 104 |
| 6.3  | Function Block Diagram Programming in SVP .....   | 106 |
| 6.4  | Angle Difference between Bus 1 and Bus 2 terminal node voltages in<br>RTDS during steady state .....                                  | 107 |
| 6.5  | Transmission Line 1 and Transmission Line 2 Real Power between Bus<br>1 and Bus 2, three phase Node voltages at Bus 1 and Bus 2 ..... | 107 |
| 6.6  | Angle Difference between Bus 1 and Bus 2 during Fault condition .....   | 108 |

|      |  |     |
|------|--|-----|
| 6.7  | Transmission Line 1 and Transmission Line 2 Real Power between Bus 1 and Bus 2 during Fault condition.....       | 109 |
| 6.8  | Three phase Node voltages at Bus 1 and Bus 2 during Fault condition .....  | 109 |
| 6.9  | Angle Difference between Bus 1 and Bus 2 after SVP action.....   | 110 |
| 6.10 | Control (trip) signal for load shedding after SVP action .....   | 110 |
| 6.11 | Transmission Line 1 and Transmission Line 2 Real Power between Bus 1 and Bus 2 after SVP action .....            | 111 |
| 6.12 | Three phase Node voltages at Bus 1 and Bus 2 after SVP action.....   | 111 |
| 6.13 | Angle Difference between Bus 1 and Bus 2 terminal node voltages in RTDS after SVP action for fault on Bus 4..... | 112 |
| 6.14 | Time Step Information for Subsystem 1 (non decoupled system).....  | 113 |
| 6.15 | Network Solution for Subsystem 1 (non decoupled system).....   | 113 |
| 6.16 | Subsystem 1 Modeling in RSCAD .....  | 114 |
| 6.17 | Subsystem 2 Modeling in RSCAD .....  | 115 |
| 6.18 | Time Step Information for Subsystem 1 (decoupled system).....  | 116 |
| 6.19 | Network Solution for Subsystem 1 (decoupled system).....   | 116 |
| 6.20 | Time Step Information for Subsystem 2.....   | 117 |
| 6.21 | Network Solution for Subsystem 2.....  | 117 |
| 6.22 | Global Time Step Information for the complete system.....  | 118 |
| 6.23 | Communication and computation intervals between different processors in subsystems .....                         | 119 |
| 6.24 | Power system and control system signal exchange between Subsystems .....   | 120 |



## LIST OF ABBREVIATIONS

|       |   |
|-------|---|
| WAN   | Wide Area Network                       |
| LAN   | Local Area Network                      |
| MURI  | Multiple University Research Initiative |
| PDC   | Phasor Data Concentrator                |
| SIPS  | System Integrity Protection Schemes     |
| PSRC  | Power System Relaying Committee         |
| WIF   | Workstation Interface Card              |
| GUI   | Graphical User Interface                |
| RTDS  | Real Time Digital Simulator             |
| GPC   | Giga Processor Card                     |
| 3PC   | Triple Processor Card                   |
| IRC   | Inter Rack Communication Card           |
| DOPTO | Digital Optical Isolation System        |
| HIL   | Hardware in the Loop                    |
| DA    | Digital to Analog                       |
| FTP   | File Transfer Protocol                  |
| EMTP  | Electro Magnetic Transients Program     |
| PXI   | PCI eXtensions for Instrumentation      |

|       |   |
|-------|---|
| PCI   | Peripheral Component Interconnect           |
| PERL  | Power and Energy Research Lab               |
| IED   | Intelligent Electronic Device               |
| PMU   | Phasor Measurement Unit                     |
| PMCU  | Phasor Measurement and Control Unit         |
| DFT   | Discrete Fourier Transform                  |
| UTC   | Universal Time Coordinated                  |
| GTNET | Giga-Transceiver Network Communication Card |
| GSE   | Generic Substation Events                   |
| SV    | Sampled Values                              |
| DNP   | Distributed network Protocol                |
| OADC  | Optical Analog to Digital Converter         |
| DAQ   | Data Acquisition Device                     |

# CHAPTER I

## INTRODUCTION

### 1.1 Introduction

Due to the large size of the existing electricity grid, the computational time required to simulate the power system behavior is very high. The simulation of power system behavior, especially the transient behavior, helps us in planning and analysis. To minimize possible damage caused by disturbances in the power system, dynamic simulation is essential in power system design. The term “electromagnetic transient” refers to transients that involve the interaction between the energy stored, such as magnetic fields of the inductances and electric field of the capacitances in the system [1]. Electromechanical transient response is not included in these kinds of simulations due to slow response caused by the interaction between mechanical and electrical systems. As the transient analysis and steady state analysis of a power system utilize non linear DAE (differential algebraic equations), solving these equations is computationally expensive. Power systems are ‘highly complex and distributed’. System partitioning has been used as one of the solution approaches to allow sharing of distributed resources for the simulation. Convergence errors are common during this type of simulations due to the

large size of the power system modeling data. The concept of Distributed Simulation has been introduced to deal with the above described problem. Distributed simulation may share resources over a network like the internet or an intranet. Modeling and simulation of large complex systems, remote monitoring and control, and non-destructive remote testing of devices are three examples of the many possible applications in which both distributed and real time distributed simulations are advantageous [8]. Distributed simulation can be viewed under decoupled simulation with added advantages. Decoupled simulation is defined as “the simulation which involves breaking of a large model into subsystems for easy circuit formulation and faster simulation” [41].

The three major factors influencing distributed simulation are partitioning of network, synchronization of the solutions and data exchange between the two partitioned systems. Distributed Simulation also allows

- 1) Sharing geographically distributed specific and unique facilities/resources,
- 2) Sharing confidential data without displaying details, and
- 3) Performing remote operation and control of power system hardware.

In this work, different possibilities of performing distributed simulations of the power system have been investigated using an electromagnetic transient based Real Time Digital Simulator (RTDS). The work analyzes the different options that are available in the RTDS for performing this type of distributed simulation specific to

- 1) Geographically distributed resources,
- 2) Locally distributed simulation, such as wide area monitoring and control test bed, and
- 3) Combination of local and geographically distributed simulation.

## **1.2 Overview of Distributed Simulation**

Distributed simulation mainly uses the concept of subsystems. This is the process of aggregating the power of several computing entities to collaboratively run a computational task in a coherent and transparent way so that the whole system appears as a single centralized system [42]. In power systems, to perform such kind of decoupled simulations, researchers mainly try to use two kinds of interfacing, viz. busbars and transmission lines. This work mainly focuses on the concept of decoupling the system using transmission lines. Distributed simulation reduces the computational time required for solving the whole network and increases the computational efficiency by utilizing remote resources. It also reduces the burden on the individual processors locally reducing convergence errors.

### **1.2.1 Geographically distributed simulation**

In this kind of simulation, the resources that are used for performing the simulation are geographically distributed at different locations. These locations can be connected and classified under the wide area network. The only added consideration in this kind of simulation with respect to the local distributed simulation is the time delay in the network that is used to connect the distant simulators. It will not be feasible to use any other medium of data transfer other than the internet for this purpose as the other options, such as fiber optics would be costly and impractical.

### **1.2.2 Local distributed simulation**

In this type of simulation, the simulators are located in the same area connected by the Local Area Network (LAN). The rest of the design aspects for this are similar to the geographical distributed simulation. Here, the delay caused by the LAN will be relatively small and negligible. The idea of fiber optic connection would be feasible in this case because of shorter distance and travel time.

### **1.2.3 Local and geographically distributed simulation**

In this kind of simulation, there will be more than two subsystems, which will be connected locally as well as geographically. For example, there can be a case where two subsystems are located nearby and connected with a subsystem geographically.

## **1.3 Overview of Wide Area Monitoring and Control**

Recent power system collapse events in different countries show the urgent need for further stabilizing the power systems beyond the available technologies. A wide area monitoring and control system (WAMCS) provides a reliable security prediction and optimized coordinated actions, which will be able to mitigate or prevent large area disturbances [2]. Phasor Measurement Unit (PMU) technology is being deployed all over the world by different utilities for wide area measurement and control actions. These units are mainly based on time synchronized measurements of voltages and currents. Even though the information provided by the EMS/SCADA systems pertain to a wide area, some of the information such as phase angles are based on state estimation and are not time synchronized which makes them less accurate as compared to the time

synchronized PMU measurements. Accurate decisions and faster control actions can be achieved using these wide area monitoring systems.

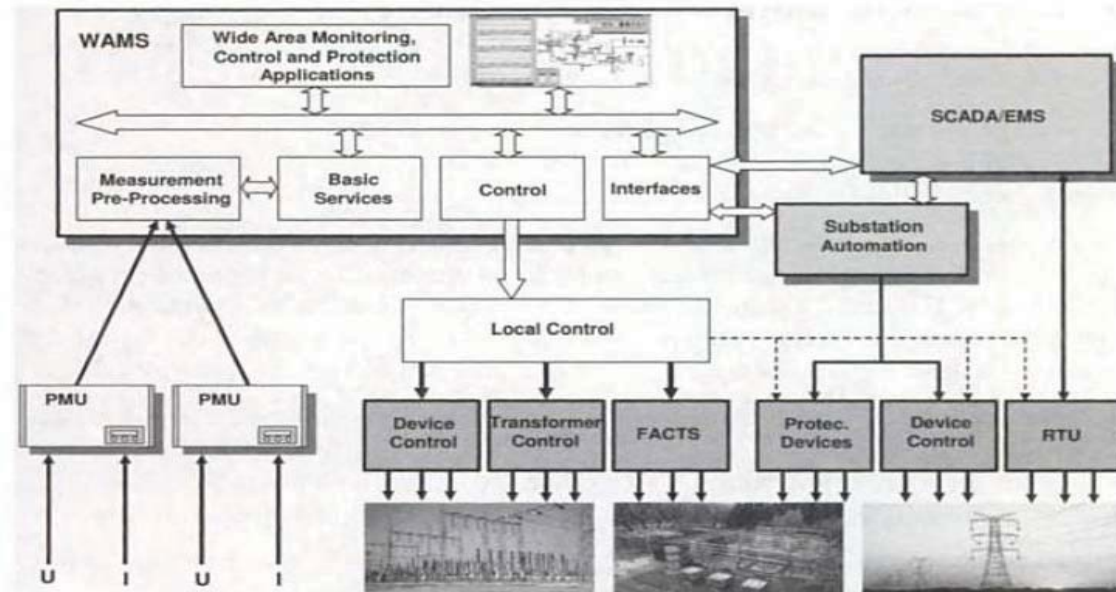


Figure 1.1

Architecture of a wide area monitoring, control and protection system [3]

Figure 1.1 shows the detailed architecture of a WAMS including the interfaces to SCADA/EMS and substation automation as well as the closed control loops back to the network controllers such as the Flexible AC Transmission Systems (FACTS) devices [3].

#### 1.4 Objectives

The objective of this thesis is to demonstrate and assess the feasibility of both non-real-time and real-time simulations using RTDS in local and geographically distributed scenarios, respectively. First, a schematic has to be proposed for such kind of

simulation and different options of getting the power system data in and out of the RTDS needs to be studied. Second, due to unavailability of predefined TCP/IP communication between the RTDS racks, there is a need to build a platform using LABVIEW, where the data between both the racks at two different locations will be exchanged through TCP/IP. Different experiments have to be performed to understand the network behavior and to determine the efficient method of transferring data within the available options. Third, the possibilities of an interface algorithm need to be studied which deals with the network delay and data loss during communication between the racks, which are distributed geographically. Fourth, build a wide area monitoring and control test bed is to be built to perform coordinated control between two subsystems in the RTDS.

Finally, the resource usage and time taken to solve the power system network need to be assessed with decoupled and non decoupled simulations using the RTDS between two racks. This work is part of a larger effort known as Multiple University Research Initiative (MURI).

## **1.5 Contributions of this work**

The major contributions of this work are listed below.

- Different options for performing distributed simulation locally and geographically using the RTDS have been analyzed and discussed.
- A test bed for performing distributed simulation between two simulators has been developed and validated.
- Different data transfer experiments have been performed between Mississippi State University and Texas A&M University for remote monitoring.



- A wide area monitoring and control test bed has been developed and validated.
- Analysis of decoupled and non decoupled simulations has been carried out locally.

## **1.6 Thesis Outline**

This thesis is organized into seven chapters.

Chapter 2 briefly introduces the research needs in the distributed simulation and proposes initiatives to meet them. These are followed by describing the implementation steps under the Multiple University Research Initiative (MURI). The concept of distributed simulation is introduced with different kinds of coupling patterns. It also presents the different research tools utilized in this work and an introduction to the coordinated distributed control.

Chapter 3 identifies the different problems and limitations in the existing methods for distributed simulation. It discusses the proposed solution briefly with the possible modeling environment. Finally, the different steps involved in the present research work are outlined.

Chapter 4 summarizes the investigation analysis for distributed simulation with some of the challenges involved. The investigation analysis mainly deals with the required hardware and software for local and geographically distributed simulation. It also discusses the different methods for development of the algorithm to deal with data loss.

Chapter 5 describes the different ways of getting the power system data in and out of the RTDS. It also discusses the proposed communication test schematic with

LabVIEW server and client modeling. It presents and analyzes the remote monitoring experiments with TAMU, using the network bandwidth and latency calculations.

Chapter 6 presents the development of wide area monitoring and control test bed using RTDS and SVP. It also analyzes the results of local distributed simulation using two racks in the same RTDS. The time step information and network solution details are presented for all the subsystems including the global time step information. Finally the chapter concludes with some discussions pertaining to geographically distributed simulation.

Chapter 7 presents the main conclusions of the research work carried out in this thesis and identifies few areas of the future work.

## CHAPTER II

### BACKGROUND AND LITERATURE REVIEW

#### 2.1 Introduction

In this chapter, initially the research needs and the initiatives taken to meet them are presented. Then, the MURI setup is introduced with the different kinds of planned implementations. Next, the mathematical concept of distributed simulation has been introduced discussing different kinds of coupling patterns. Finally, the research tools and a brief introduction to the coordinated distributed control are presented including a wide area control framework.

#### 2.2 Background

Network distributed simulations have remained a hot research topic because of their high potential which can contribute to the computational performance and improvement of time step based transient simulators. Study of power system behavior under stressed environment provides a better understanding for proper operation and protection of the electric grids. With the increasing deployment of distributed intelligence throughout the power systems, the possibilities for distributed control and operation

schemes are becoming progressively more attractive and feasible [32]. As the time domain simulators, such as the RTDS, simulate the real time performance of the power system, distributed simulations using this type of simulator enhance the flexibility and accuracy for performing transient simulations.

Distributed simulation, as described in [31] involves separately and concurrently computing the dynamics of several parts of a system that is subject to natural conservation laws. Then it couples the response fragments into one comprehensive solution at each integration time step.

### **2.2.1 Department of Defense (DoD) Multiple University Research Initiative (MURI)**

The MURI project consists of a research team formed by five universities, whose goal is to develop Remote Testing and Measurement (RTM) devices to enable the study of normally operating and stressed interconnected power systems utilizing laboratories located remotely from each other. The participating universities are Drexel University (DU), Northeastern University (NEU), Iowa State University (ISU), Texas A&M University (TAMU) and Mississippi State University (MSU). Three of these universities will allow open access to their hardware power laboratories and two universities will focus on monitoring and control functions as well as data model designs. There are five different kinds of experiments and remote interactions desired [4]. These are

- Implementation I: software power simulation tools to software power simulation tools
- Implementation II: software power simulation tools to hardware power systems
- Implementation III: hardware power systems to hardware power systems

- Implementation IV: software power simulation tools to multiple hardware systems
- Implementation V: multiple interconnected hardware power systems

This work-in-progress will have a vital impact on the new generation power engineers who understand measurement design and its relationships to hardware component analysis, mathematical modeling and analysis of large scale power systems [4]. The conceptual remote testing and measurement setup between participating universities is shown in Figure 2.1.

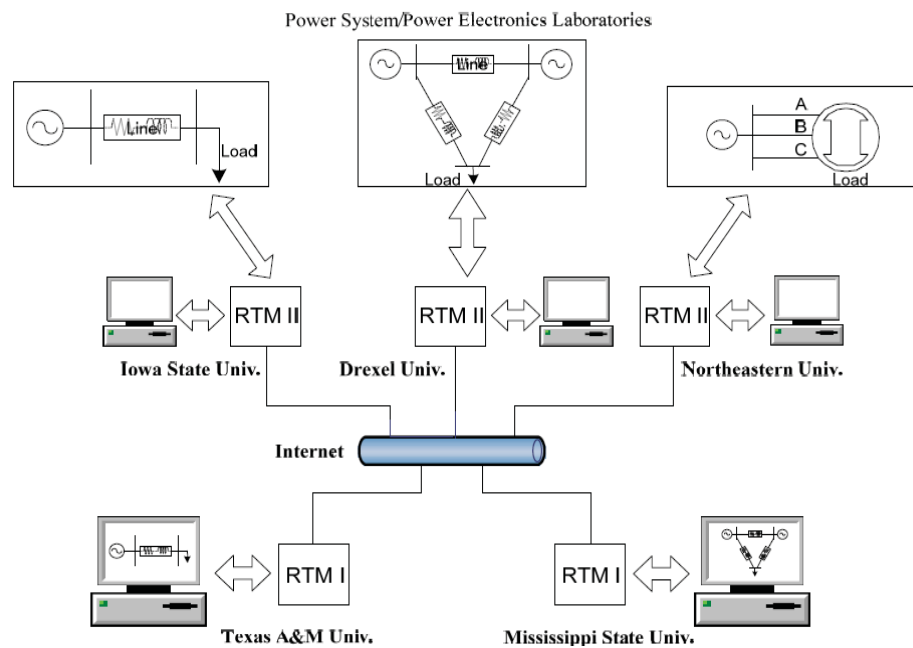


Figure 2.1

Remote testing and Measurement (RTM) Setup between Universities

## 2.2.2 MURI Implementation and Network Capability

The main objective of the MURI project is to develop a Simulation-Stimulation (Sim-Stim) Interface that will integrate the software to hardware and the hardware to

hardware interconnections so the remotely connected systems work together. In such simulation there will be two Sim-Stim interfaces, which are connected through internet. The Sim-Stim interface mainly consists of power sensors and converters. The converters include Digital to Analog and Analog to Digital type, depending upon the application. The Sim-Stim interface works as follows.

In these types of simulations, the power system will be divided into two parts, which consists of the source and load on either side. A virtual source and load will also be present on either side representing the missing portion. Initially the source profile will be sensed by the power sensors and it will be converted into digital data by the A/D converter before sending it through the internet. After receiving the source profile, the virtual component of the source on the other side gets activated depending upon the received status information. Then the load appears to develop a current as if it is really connected.

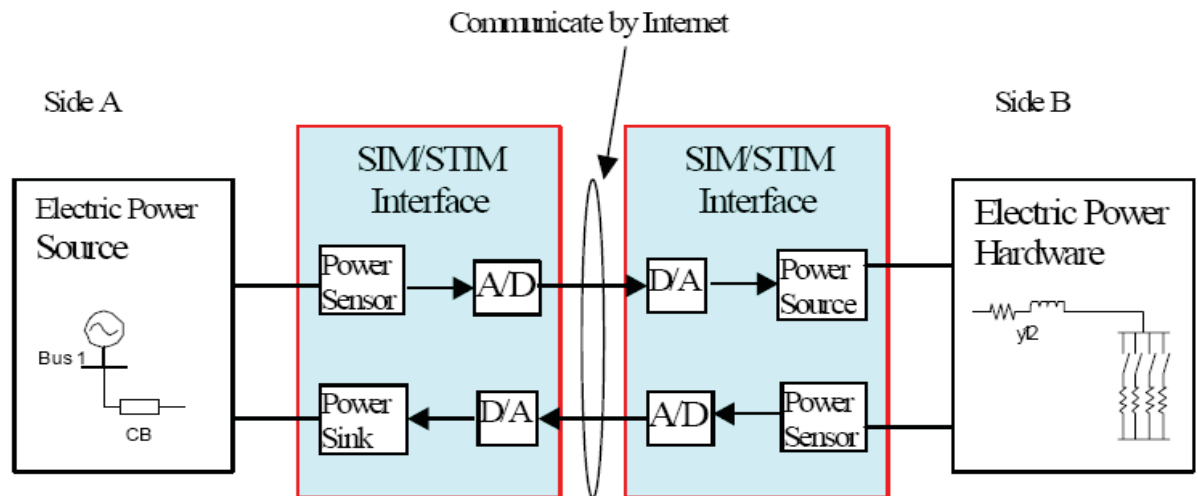


Figure 2.2

Structure of remote Hardware/Hardware Implementation [4]

Similarly the load profile will be sent back to the other side and a virtual load will be connected to the real source depending upon the received value of the current absorbed by the load. Generally, the voltage level information will be transferred from the source side to the load side and current level information will be transferred from the load side to the source side. Both the source and load will respond for any change in either part of the system. This type of interface must be consistent in terms of real time performance during various operating conditions such as steady state and dynamic operations. The interface should either generate or absorb power exchanging real power between both ends. The type of power match that the Sim-Stim interface must achieve, depends on the specific power system (e.g. AC, DC or AC/DC), its architecture, the operating conditions of interest (steady state or dynamic operation, normal/emergency) and type of phenomena of interest (power quality, stability etc) [4]. Authors in [5] provide the guidelines and design issues for developing a Sim-Stim interface that integrates power hardware with software for testing and development purposes.

While doing these types of simulation, the communication network plays a vital role affecting the accuracy of the simulation as the simulation is time dependent. Simulations mainly run into mathematical stability problems when dealing with delays caused by data acquisition devices, A/D converters and the network. Hence, researchers should try to use the Ethernet based measurement systems in the LAN as it has been proven as an effective and economical networking solution [6]. Increasingly high bandwidth of Ethernet and fast Ethernet along with the use of intelligent switches have proven to deliver predictable, even deterministic, I/O over Ethernet [6].

For connecting labs across campuses, Internet2 [30] can be used, which provides high speed network connections between universities facilitating leading edge network capabilities such as higher capacity and QoS (Quality of service). At the physical layer, Internet2 has established an advanced network infrastructure with backbones operating at 2.4 Gbps (OC48) to 10 Gbps (OC192) capacity, and at network layer, Diffserv is applied which is a new QoS that ensures data with high priority to be delivered in time[4]. So usage of such services could provide a better network infrastructure for such experiments with reliable data transfer and controlled time delays.

### **2.3 Distributed Simulation**

Distributed simulation is an important concept during different stages of system design in which modeling and simulation, remote monitoring and control and non destructive remote testing are some of the many important applications. Distributed simulations can reduce the burden and computational load that is required while simulating larger power systems such as shipboard systems, which involves a complex design and testing process. The modeling of power systems for advanced electric ships requires full scale prototypes and pre-simulation before the installation of the actual system. So distributed simulation can be used to partition the large shipboard power system into smaller manageable systems and solve those separately using distributed resources. Then, the results from different smaller subsystems are combined to arrive at an overall system solution. The dynamics of the several subsystems are computed separately and concurrently provide a path to share the computational load by multiple



computers thus, effectively reducing the simulation time [8]. A conceptual topology of the distributed simulation is shown in Figure 2.3.

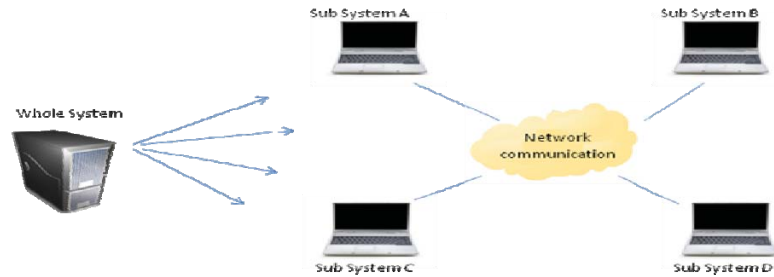


Figure 2.3

#### Conceptual Topology of Distributed Simulation [8]

Authors in [9] proposed a generalized coupling method for natural coupling, which imposes physical conservation laws at the decoupling point. It describes the simulation models developed in Virtual Test Bed (VTB), which allows natural and signal coupling distributed simulation. The transmission line modeling technique has been used in this work as the decoupling method because of its increased efficiency and accuracy.

Figure 2.4 shows the decoupling pattern proposed by Mississippi State University where this scheme was implemented before on Virtual Test Bed (VTB) software [9]. This scheme was tested on different terrestrial power systems and also on a shipboard power system [9].

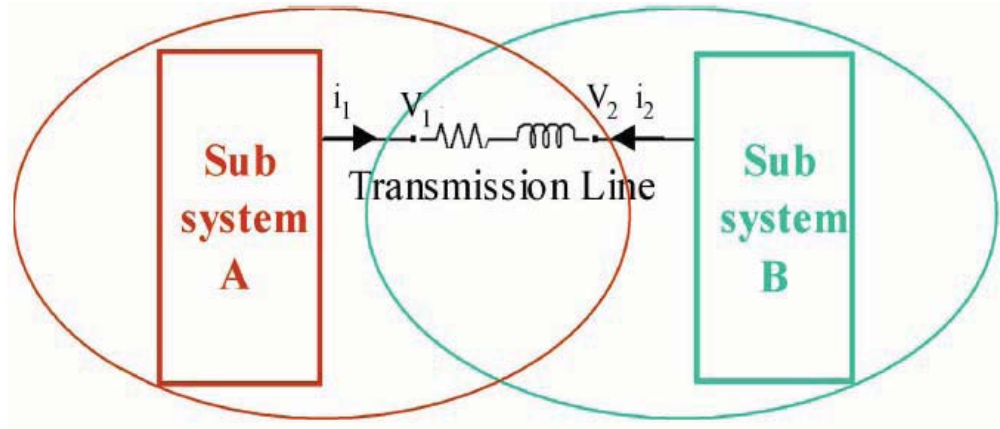


Figure 2.4

A prototype of the power system before it is distributed [4]

Figure 2.5 shows the partitioned power system with a transmission line interface. It shows the signals that are exchanged between both the systems along with the other parameters. The solution of both the subsystems combined together, gives the overall system solution. Initially, this system was implemented using MATLAB before migrating it to VTB.

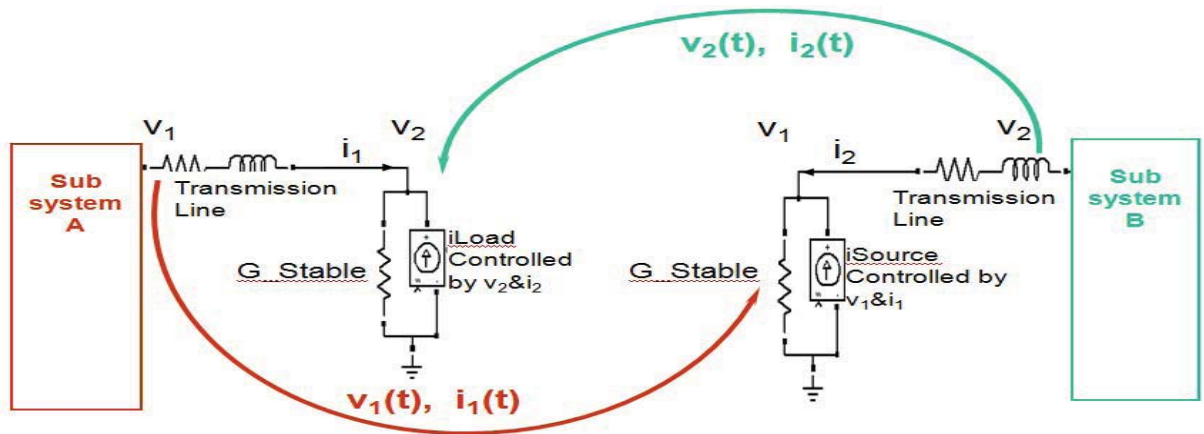


Figure 2.5

A prototype of the power system after it is distributed [4]

For natural coupling and signal coupling levels, the four kinds of agents, that were developed for distributed simulation are monitor agent, acquisition agent, source agent and load agent. These were used for sending and receiving the measurements and signal commands. The signal exchange in Figure 2.5 is explained below.

Initially,  $V_2$  and  $I_2$  in subsystem A are initialized to zero. Then, an equivalent circuit for subsystem B is constructed by setting the  $G\_Stable$  to a user defined value followed by the calculation of  $iLoad$ . Now the subsystem A is solved and the obtained current and voltage values are sent to subsystem B. In a similar way, subsystem B also sends the voltage and current values. With the obtained values, either side creates a new equivalent circuit again for the missing subsystem and exchanges the newly obtained values. Finally, the convergence is checked for the current value on both the sides and the simulation moves to next time step if the convergence matches. Load models were developed in VTB and were validated by comparing them with different other simulation packages such as MATLAB/Simulink and Power World. Finally distributed simulations were performed using VTB based on the explained decoupling concept and developed load models.

### **2.3.1 Parallel algorithm and different kinds of coupling patterns**

Parallel algorithms have been researched mostly to address many types of power system problems. These are mainly based on different decomposition and aggregation techniques. Even though the actual testing on multiprocessor architectures is more at the beginning stages, most of the previous works have been based on developing different types of parallel algorithms. Most of the parallel algorithms utilized the features of

triangular factorization and forward/backward substitution [10-13]. Parallel simulations have always been a lure to the power industry because of their efficiency in calculations and interactive technology.

A coupling pattern plays a vital role in the convergence of decomposed circuits. There are different kinds of basic and generalized coupling patterns. These patterns can be applied to large sized systems, which use the decoupled circuit method based on graphs [14]. The relaxation techniques in circuit simulators represent a large circuit by the algebraic or ordinary differential equations. In large circuit cases, the system is decomposed into smaller parts and repeated calculations are carried out by relaxing the behavior of the surrounding subsystems. Most of the time relaxation techniques have limited application because of their poor convergence as the decomposed circuits are not able to properly represent the original circuits. This problem was overcome by the generalized coupling patterns. There are mainly two types of basic coupling patterns and one type of generalized coupling pattern. The two basic types of coupling patterns are V-type and I-type as shown in Figures 2.6 and 2.7. V-type coupling is mainly used in relaxation based circuit simulators. It contains two voltage sources, which are controlled by the node voltages and conductive elements. The main reason for fast convergence in this pattern is because of the positive local feedback between the node voltages. However, the iterative solution converges slowly if the product of  $Y_a$  and  $Y_b$  is much less than  $Y_{ab}$ .

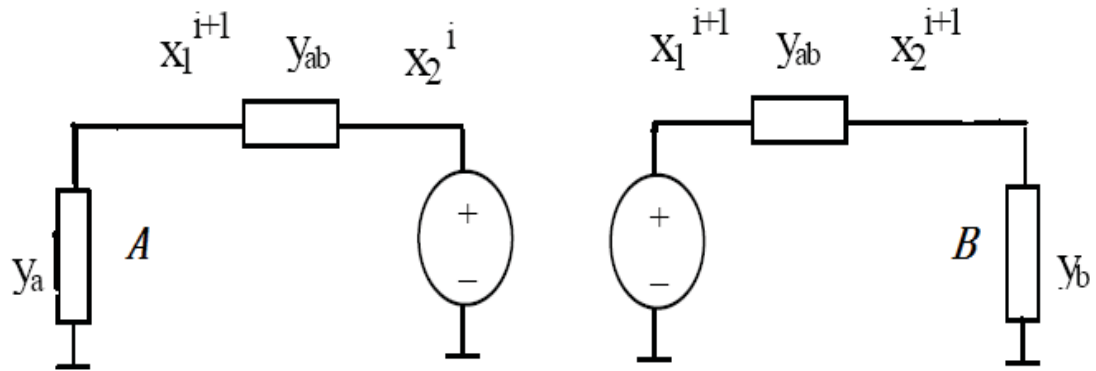


Figure 2.6

Basic Coupling Pattern: V-type Coupling Scheme [14]

Here,  $Y_a$  and  $Y_b$  are input conductances of the subsystems A and B respectively.  $X_1$  and  $X_2$  are the node voltages of A and B, where both the subsystems are connected by a common conductance  $Y_{ab}$ .

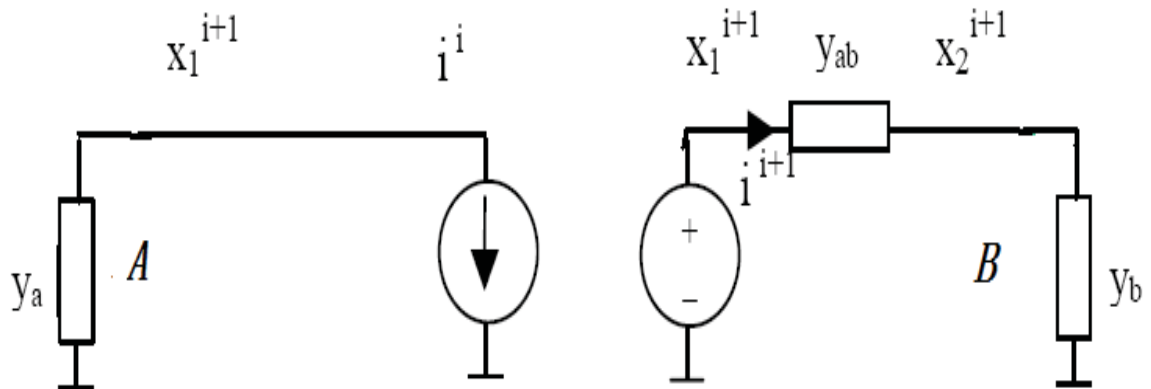


Figure 2.7

Basic Coupling Pattern: I-type Coupling Scheme [14]

The I-type of coupling was mainly used to brace the parts connected by pass transistors. This kind of coupling does not secure the convergence as it always results in negative feedback. Hence this is not very popular in relaxation based solvers. Even though the iteration converges rapidly sometimes, the resulting error changes its sign every iteration. So this kind of coupling is not as applicable as the V-type coupling pattern. As we have seen above, both the patterns have their own advantages and disadvantages. So to improve the speed of convergence in a positive direction, it would be better to have a technique which can adopt the advantages of both the patterns. This type of coupling which combines the properties of both the couplings is called as VI-type coupling [14]. It is shown in Figure 2.8.

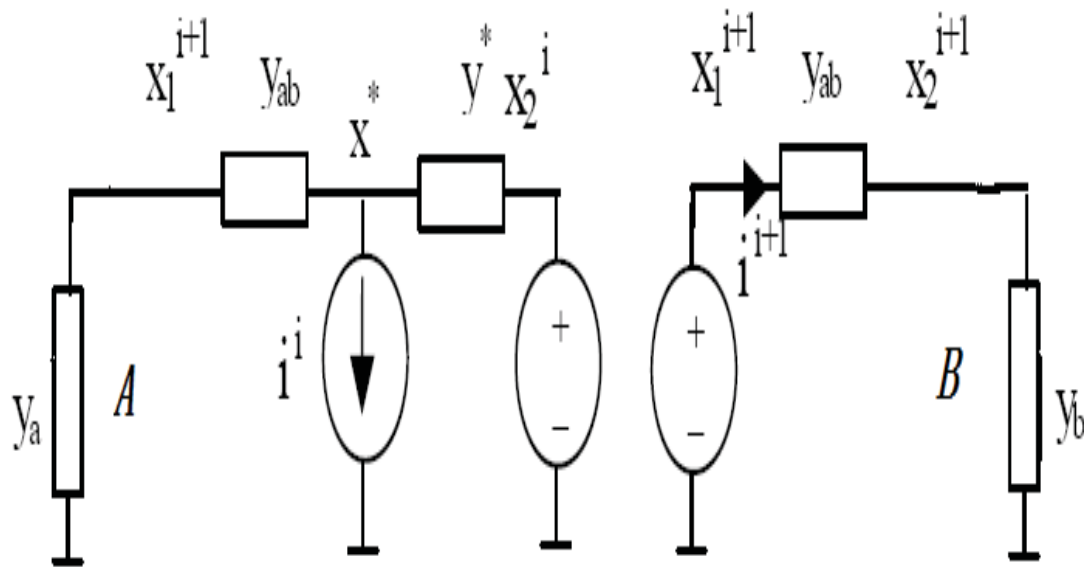


Figure 2.8

Basic Coupling Pattern: VI-type coupling scheme [14]

Paper [14] introduces these basic patterns and applies them in linear and non linear circuit simulations with dynamic analysis and some experimental results.

## **2.4 Coordinated Distributed Control**

Coordinated distributed control mainly deals with the coordination of communications and controls of the geographically dispersed power system. Here, a wide area power system is monitored and controlled in a distributed scenario.

### **2.4.1 Wide Area Monitoring and Control**

Recent power system blackouts have shown the inclination for more concern towards wide area monitoring and control to achieve reliability. Presently, wide area monitoring is mainly performed using the PMUs (Phasor Measurement Units) by collecting the time referenced power system data. Finding the phasor estimate of an arbitrary sinusoidal signal is not as easy as finding it for a defined sinusoidal waveform. For determining the parameters of phasor estimate of an arbitrary signal, a series of sample points of the waveform should be considered at appropriate intervals over a sufficient length of time. The quality of the phasor measurement will be dependent on factors such as sampling rate, algorithm used for estimating the value, measurement interval as well as the signal content of the phasor [15]. Most of the PMUs use the DFT (Discrete Fourier Transform) technique for estimation with a sufficient sample rate and accurate synchronization using UTC to produce an accurate and very stable phasor value for most system conditions [16].

Power/frequency disturbances tend to produce system wide clues and often opportunities to observe the propagation of the disturbance across the entire network [17]. Wide area monitoring using PMUs is accurate and globally synchronized. The PMUs offer very fast data rates such as 30 to 60 samples per second. This data mainly consists of time tagged bus voltage magnitudes with angles and similarly current magnitudes with angles. In a real time wide area phasor measurement system, the PMU data will be collected into a PDC (Phasor Data Concentrator) and will be used for different phasor data applications. The PDC performs some of the important operations such as protocol conversion, error checking and time correlation for the incoming data from the PMUs. It also performs sample selection and delay control for the outgoing data while monitoring the system for PMU errors. Such kind of wide area measurements from PMUs can be used for power system protection and control applications, preserving the system integrity. There are many System Integrity Protection Schemes (SIPS) actions classified by IEEE Power System Relaying Committee (PSRC) [43], such as load rejection, generator rejection, out of step tripping, under voltage load shedding, under frequency load shedding, adaptive load mitigation, voltage instability advance warning scheme, and angular stability advance warning scheme etc. All the SIPS actions can be performed faster and more accurately by using PMU data in a wide area network rather than state estimated data. Wide area control addresses automatic healing capabilities by proposing decisive smart topology changes and control actions with the goal of maintaining the integrity of the grid under imminent, adverse conditions [17].



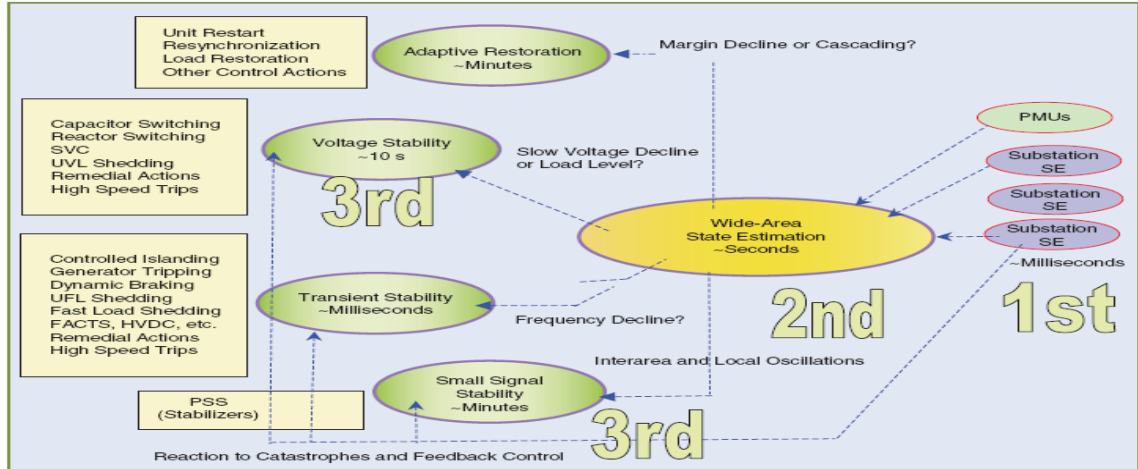


Figure 2.9

### Wide Area Control Framework [17]

## 2.5 Tools

This section briefly describes about the different hardware and software modules used in this research work. In the later chapters, the integration of these tools to achieve the desired objectives will be presented.

### 2.5.1 Real Time Digital Simulator

RTDS (Real-Time Digital Simulator) is a real-time power system simulator, which performs digital simulation of electromagnetic transient programs. It was developed by RTDS Technologies Inc [www.rtds.com]. RTDS is a fully digital simulator, which can perform simulations with a time step of 2 microseconds. The real-time operation of the RTDS allows it to perform development and testing of protection and control techniques for a power system. RTDS uses a Graphical User Interface (GUI), known as RSCAD for the simulator operation. The main advantage of RTDS is complete

parallel processing where the entire hardware is assembled in modular units called racks. Each rack contains different digital signal processors, such as Giga Processor Card (GPC) and Triple processor cards (TPC). RTDS is comprised of many hardware units and some of them are Workstation Interface Card (WIF), Inter-Rack communication Card (IRC), Digital Input and Output card (DOPTO) and so on. All the communications between the RTDS and the workstation are carried by WIF card. The IRC takes care of the communication between the racks. The RTDS at Mississippi State University consists of a cubicle with two processor racks (as shown in Figure 2.10). These two racks consist of eight Triple Processor Cards and two Giga Processor Cards. Each 3PC card consists of three analog digital signal processors known as ADSP21062 (SHARC) and in addition, this card also includes dedicated analog output channels, digital input and output ports and an optional optical input/output port [18]. The analog outputs can be used for connection of external equipment for hardware in the loop studies. The GPC card contains two IBM PPC750GX PowerPC processors (RISC) with powerful computational ability of solving the entire network solution containing up to 54 nodes and 56 breakers [18]. These GPC processors operate with a core frequency of 1GHZ and floating point computation rate of 1GFLOP [18].

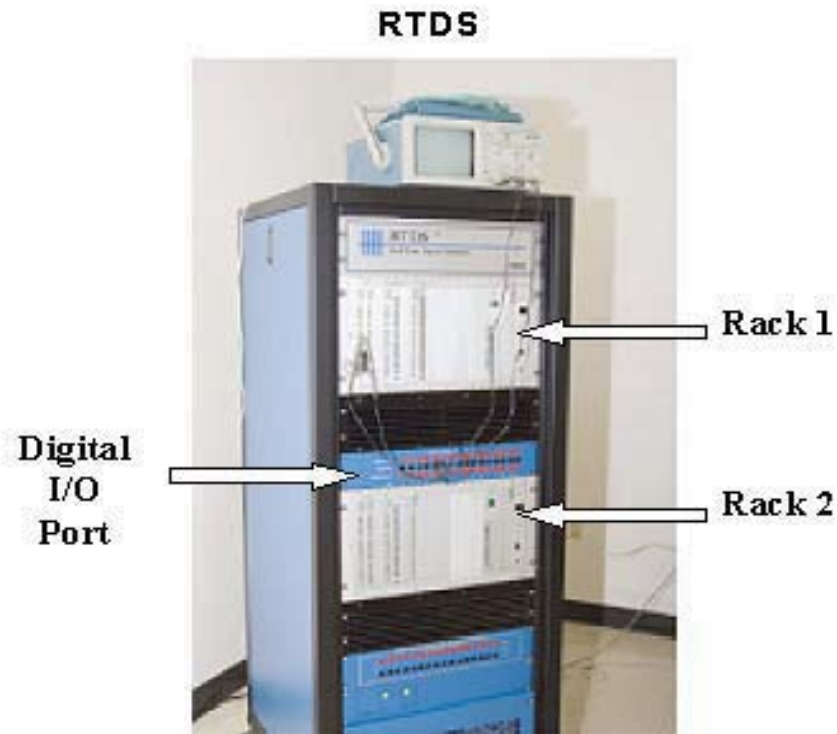


Figure 2.10

Real Time Digital Simulator at MSU

RTDS can be used to perform closed loop experiments, such as Hardware in the loop (HIL) testing of protective devices using real-time relays. RTDS is a high end simulator, which has increased speed and accuracy compared to the older analog simulators.

The Giga Transceiver network card (GTNET) provides real-time communications for the simulator through Ethernet. It has different firmware options such as GSE, SV (Sampled Values), PLAYBACK and DNP (Distributed Network Protocol). Each firmware performs a unique role in power system communications. The GTNET\_SV firmware represents power system voltages and currents in the format of IEC 61850-9-2 sampled value messages. So these messages can be detected and received by any device

that understands the IEC 61850 protocol, which is the communication standard for the design of substation automation. This standard is used for communication networks and systems in substations. Examples of such devices are relays and IEDs (Intelligent Electronic Devices). The GTNET is shown in Figure 2.11.



Figure 2.11

Giga Transceiver Network Communication Card [18]

### 2.5.2 National Instruments-PXI Controller

The NI PXI system is a high performance, rugged PC based platform for measurement and automation systems. PXI combines the Peripheral Component Interconnect (PCI) electrical bus with the rugged, modular Eurocard mechanical packaging of CompactPCI and adds specialized synchronization buses and key software features [20]. NI PXI is highly applicable in data acquisition, measurement and testing of power systems.

MSU Power and Energy Research Lab (PERL) is equipped with the NI-PXI system, shown in Figure 2.12. The LabVIEW graphical programming language is used to

interact with NI-PXI. Device such as NI-PXI 8196 controller, NI-PXI 6251, NI-PXI 6608, and NI-PXI 5412, occupy these NI-PXI system chassis slots [21].

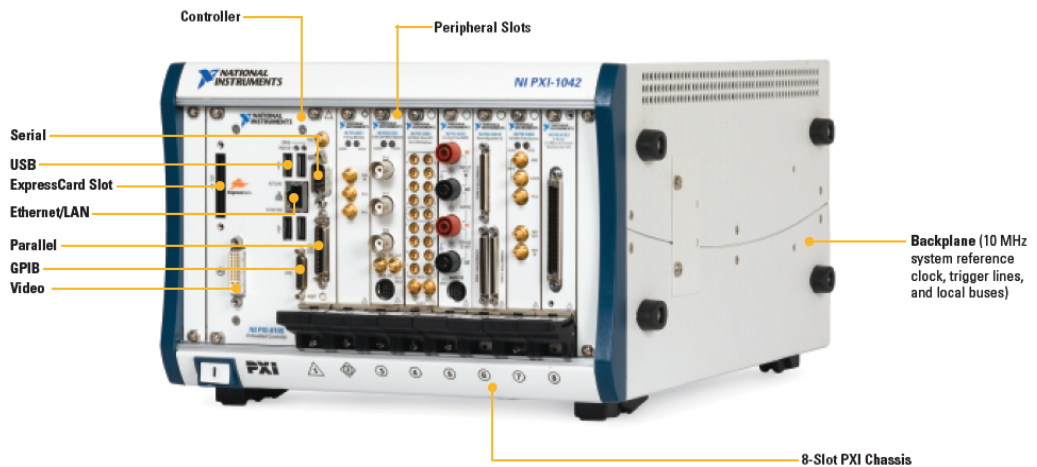


Figure 2.12

NI PXI System with an 8 slot Chassis [20]

The NI PXI 8196 controller is shown in Figure 2.13. The 8196 Real Time Embedded Controller at Mississippi State University consists of 2GHz Pentium M 760, 1GB RAM, 4 USB 2.0 Ports, an express card with 34 slots, an integrated hard drive, GPIB, serial and parallel communication ports. It can be used for data acquisition, real time control and data transfer over the Ethernet.



Figure 2.13

NI PXI 8196 Controller [21]

The NI PXI 6733 is a high speed 16 bit analog output device with an 8 channel output at an updating rate of 1MS/s. It also provides eight high-speed digital I/O lines, two 24-bit counters, digital triggering and external clocking. It supports LabVIEW, LabVIEW Real Time, CVI (VIs which use ANSI C development environment) and measurement studio for Visual Basic. This device is capable of generating the waveforms up to 500 kHz replacing several kinds of instruments, such as low speed arbitrary waveform generator and function generator. The NI PXI 6733 is shown in Figure 2.14.

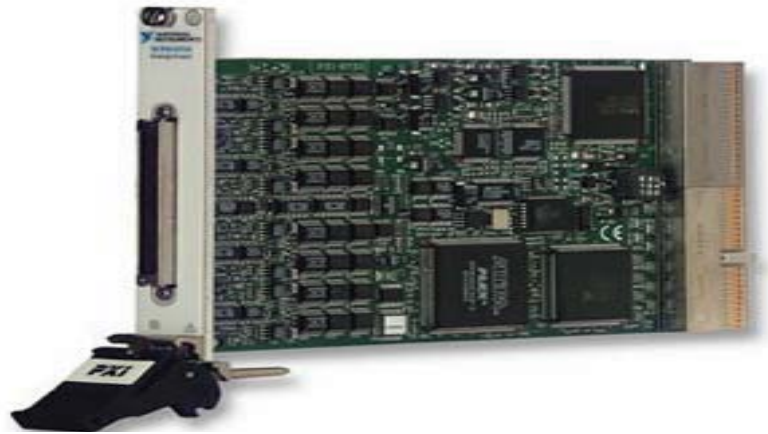


Figure 2.14

NI PXI 6733 Analog Output Device [19]

The SCB-68 is a rugged, very low noise signal termination I/O connector. The connector is shielded with 68 pin connectors and it has two general purpose breadboard areas. It is mainly used for interfacing I/O signals to plug in DAQ devices. It is compatible with so many series of DAQ devices. It is shown in Figure 2.15.



Figure 2.15

NI Shielded Connector Block-68 [19]

### 2.5.3 SEL 421 Protection, Automation and Control System

The SEL 421 was developed by the Schweitzer Engineering Laboratories, Inc. (SEL) mainly for protecting transmission lines with distance and directional functions. Some of the key features of these systems are Digital Fault Recording, Complete Sub Cycle Line protection, High Accuracy Time Stamping and Synchrophasor Measurements. The SEL 421 is IEC 61850 certified and it supports other protocols such as DNP, FTP and Telnet, where the last two are mainly used for downloading event reports in seconds. In order to implement a Wide Area Monitoring and Control Test Bed, the SEL 421 system is used as a Phasor Measurement Unit, which gives out synchrophasor messages in compliance with IEEE C37.118 synchrophasors standard [44]. The SEL 421 system is shown in Figure 2.16.



Figure 2.16

SEL 421 Protection, Automation and Control System [22]



#### 2.5.4 SEL 3378 Synchrophasor Vector Processor

SEL 3378 is known as the Synchrophasor Vector Processor because it takes the synchrophasor messages to perform wide area protection and control tasks. It is a real time synchrophasor programmable controller, which can time align incoming messages, process them with an internal logic, and send control commands to external devices to perform user defined actions [22]. It can measure system quantities such as phase angles, voltages, currents, real/reactive power and use them to improve the system efficiency and performance by implementing actions such as minimizing loop flow, balancing loads, controlling SVCs, generators and circuit breakers. It uses the flexible IEC 61131 programming language to layout the protection and control schemes. With the available preconfigured function blocks, the user can perform power calculation, phase angle difference, modal analysis and substation and state topology processing.

The closed loop operation of the synchrophasor vector processor with the power system is shown in Figure 2.17. It shows the basic scheme, where it gets the measurements from the terrestrial system into the function calculation block and gives out the control or protection commands signals back into the system. The function calculation block is defined by the typical user application. The synchrophasor vector processor is shown in Figure 2.18.

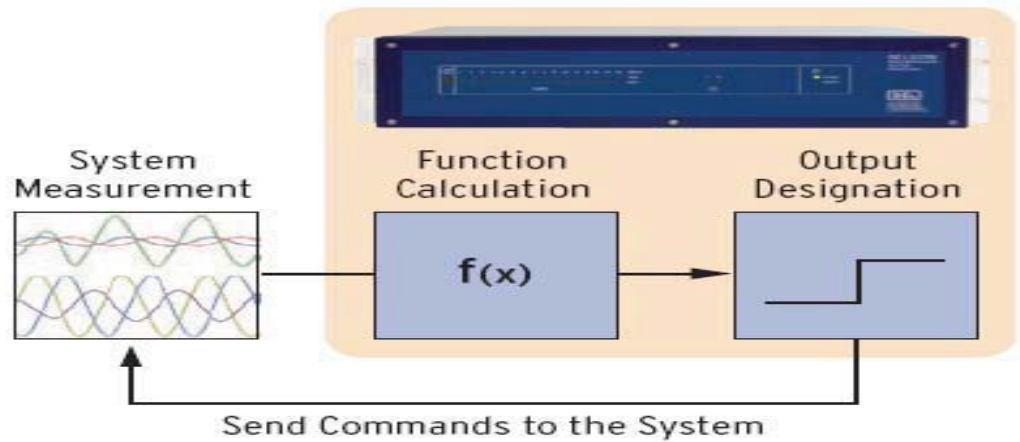


Figure 2.17

Operation of SEL 3378 with the Power System [22]



Figure 2.18

SEL 3378 Synchrophasor Vector Processor [22]

### 2.5.5 RSCAD Software Suite

On the software side, the RTDS consists of a software suite known as RSCAD. It mainly consists of GUI and a library of power system and control system components. The RSCAD is mainly used for creating, simulating, modifying and running the power system models. It also performs result analysis and data acquisition. The library gives access to the pre-defined power system and control system models, where the user can

even edit the existing models. The user has an option of creating their own models using the features such as User Defined Components (UDC) and C Builder. This adds more flexibility to the existing features of RTDS. Figure 2.19 shows the several modules of RTDS. The Draft is mainly used to layout a schematic diagram of the system to be simulated. The extensive power system and control system library is available inside the draft. After compiling the model in draft without any compilation errors, the system is run in real time using the RunTime option. In runtime, the user will be able to control and interact with the simulation by creating plots, switches, sliders, meters etc. It mainly serves as the graphical user display for monitoring and control actions. MultiPlot is mainly used for post processing applications of the results. The user can perform analysis, printing and calculations of the obtained data. The TLine and Cable files mainly provide the conversion of TLine/Cable data into a form which can be used by the draft [18].



Figure 2.19

RSCAD Architecture [18]

### **2.5.6 National Instruments LabVIEW**

LabVIEW (Laboratory Virtual Instrument Engineering Workbench), developed by National Instruments (NI), is a graphical programming language that is user friendly and is used to develop complex measurement and control applications very quickly and easily [21]. LabVIEW is mainly used for data acquisition, industrial control and instrument control on a variety of platforms such as Microsoft Windows, UNIX, LINUX and Mac OS. LabVIEW uses dataflow graphical programming language, where the execution is determined by the structure of a graphical block diagram. LabVIEW programs/sub - routines are called Virtual Instruments (VIs) because of their appearance and their imitation of actual instruments. A LabVIEW VI mainly consists of three parts. These are front panel, block diagram and a connector panel. Basically the front panel is user interactive interface which allows inputting or extracting data. The actual executable code can be built in the block diagram using the predefined and user defined blocks. A connector can be used to represent the VI, if it has to be used as a sub routine. Normally, a VI within another VI is known as sub VI.

### **2.5.7 NI Measurement and Automation Explorer**

MAX is basically an NI explorer, which allows user to view the NI systems that are associated with a particular computer. It displays the devices, interfaces and softwares in the computer including the remote systems. The remote systems can be accessed and viewed only when the connection between the computer and remote systems is active. It also helps in calibrating the devices where the user can create, edit and scale virtual instruments [21]. Figure 2.20 shows the Measurement and Automation Explorer running

on a computer. In this figure, the configuration is shown on the left side, status of the remote systems in the middle with the available details such as model number, serial number and MAC address. The network details, such as IP address, subnet mask, gateway and DNS server can also be seen.

### **2.5.8 National Instruments LabVIEW Real Time**

NI LabVIEW Real Time combines the power of LabVIEW graphical programming and the power of a real time operating system enabling researchers to build deterministic real time applications. Since windows is not a guaranteed real time operating system, the LabVIEW program runs on a separate real time operating system with the help of a real time module. LabVIEW will have direct control over all the tasks on the real time target. With such a real time operating system, the user can prioritize tasks in such a way that the most critical task always takes the control of the processor when needed [23]. At Mississippi State University the real time operating system is the NI PXI where LabVIEW 8.5 is installed on it.

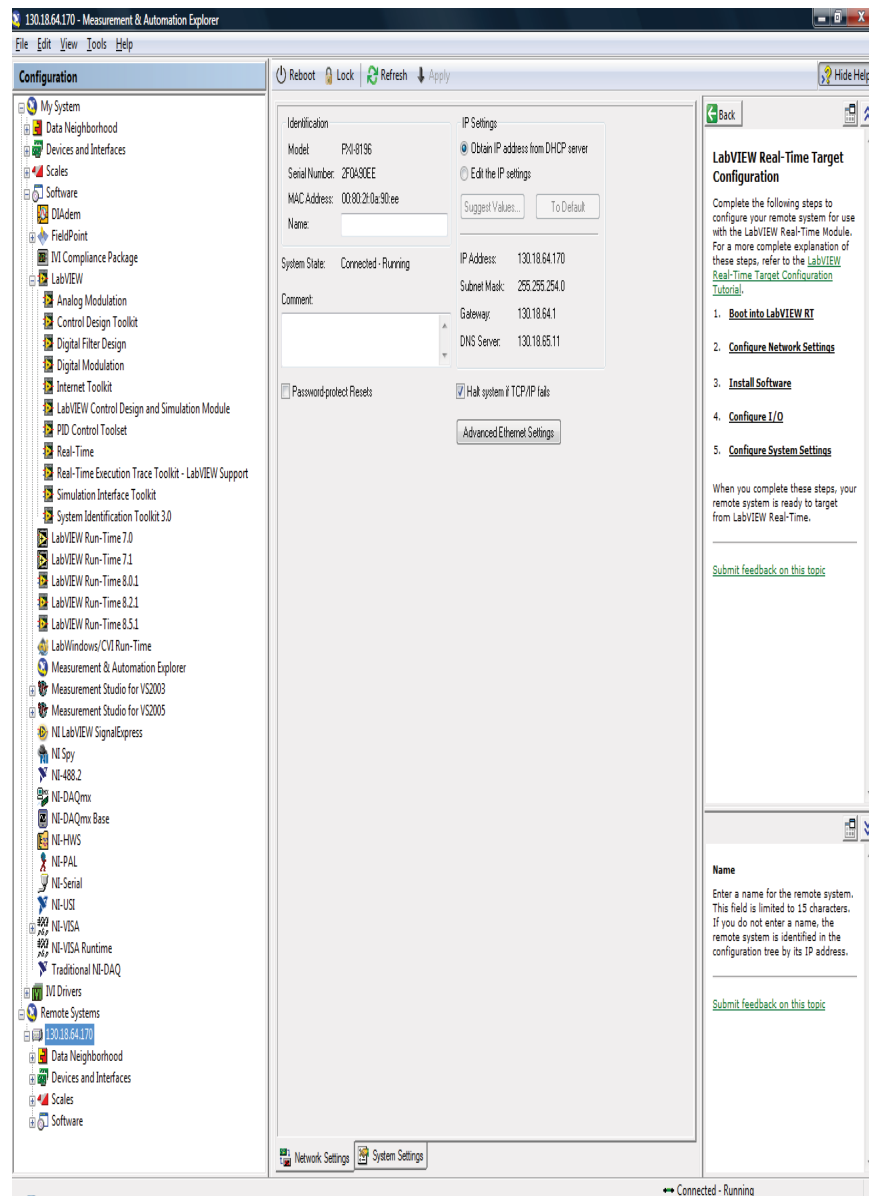


Figure 2.20

## NI Measurement and Automation Explorer

Figure 2.21 shows the configuration with a closer look on the software present at the host computer as well as the remote real time target.

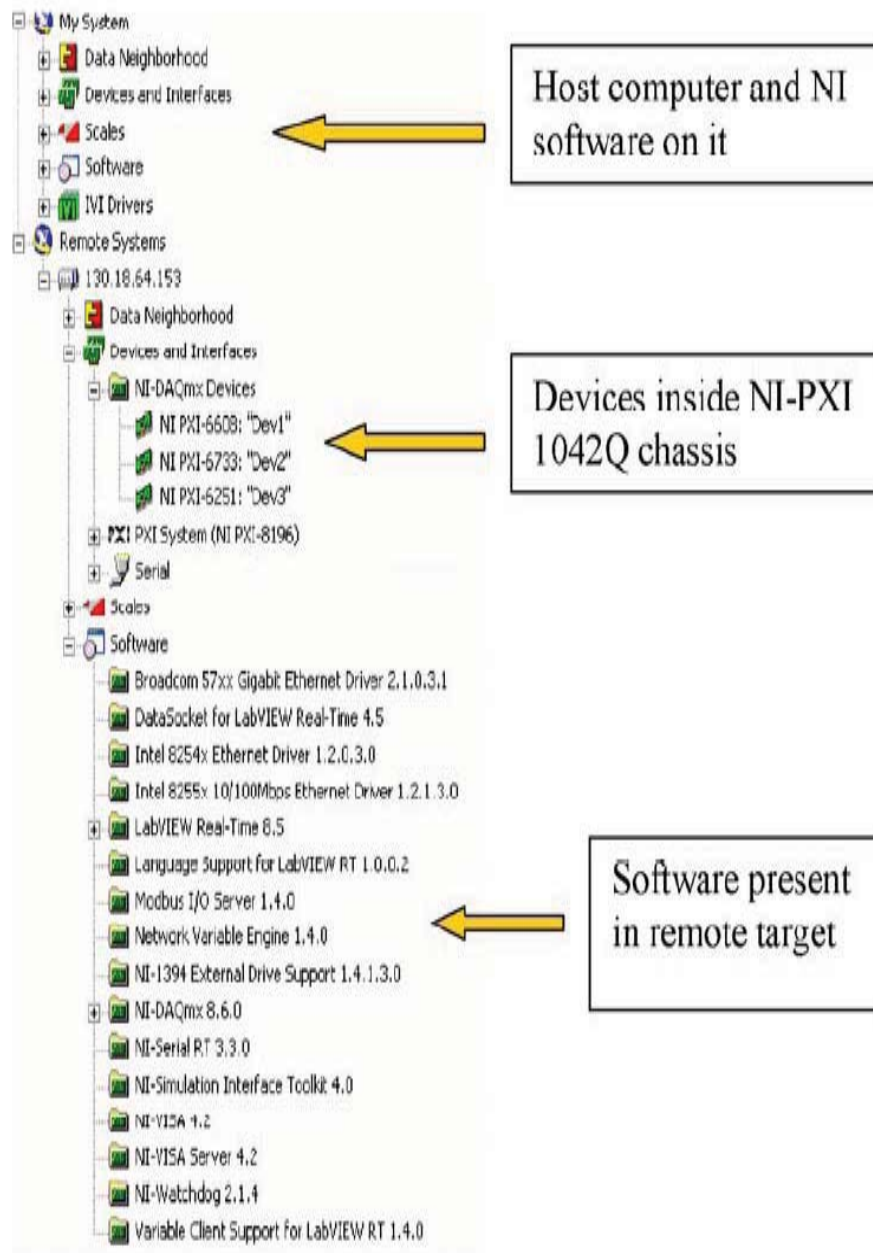


Figure 2.21

Configuration of Measurement and Automation Explorer [21]

### 2.5.9 Wireshark

Wireshark is a network protocol analyzer, which is used to analyze the incoming and outgoing network traffic of a PC. It is used as a standard across many industries and educational institutions for performing network monitoring actions. Some of its features include live capture, offline analysis, deep investigation of hundreds of protocols, rich VoIP analysis, capability to run on multi-platform, ability to read Live data from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI, and others [29].

## 2.6 Summary

To summarize, the background of the present work has been discussed with some literature review and the implementation details in this chapter. Previous work includes development of small distributed simulation models using Virtual Test Bed (VTB) and Matlab, which was performed in local area network without any considerable delays [4]. This work extends the distributed simulation concept and the power system models into wide area network using the RTDS. In this work, the network characteristics, such as bandwidth and latency, are studied and analyzed in wide area network. Distributed simulation techniques and different kinds of coupling patterns were discussed. Finally, the wide area monitoring and control scheme has been discussed briefly with some of its advantages. Further, the different hardware and software tools used in this work have been presented. The list of hardware and software used in this research are summarized below in Table 2.1 and 2.2.



Table 2.1

List of hardware used

|   | <b>Name</b>                                  |
|---|--|
| 1 | Real Time Digital Simulator (Includes GTNET) |
| 2 | NI PXI Controller (Includes DAQ card)        |
| 3 | SEL 421 Phasor Measurement and Control Unit  |
| 4 | SEL 3378 Synchrophasor Vector Processor      |
| 5 | SEL 2407 Satellite Synchronized Clock        |

Table 2.2

List of softwares used

|   | <b>Name</b>                                 |
|---|---|
| 1 | NI LabVIEW 8.5                              |
| 2 | LabVIEW Real Time 8.5                       |
| 3 | NI DAQmx 8.6.1f0                            |
| 4 | Measurement and Automation Explorer 4.4.1f0 |
| 5 | RSCAD 2.010.3                               |
| 6 | SVP Configurator 2.3.7.2                    |
| 7 | Wireshark 1.0.3                             |
| 8 | Accelerator Quickset Software 4.9.0.5       |

## CHAPTER III

### PROBLEM STATEMENT AND PROPOSED SOLUTION

#### 3.1 Introduction

This chapter identifies the problems and limitations with the existing methods. It also describes the proposed solution approach for dealing with different kinds of the problems. It briefly describes the modeling environment and the steps involved in this research work.

#### 3.2 Problem Description

The existing methods for solving transient simulations are highly powerful but these are extensively time consuming. The average time taken to solve a system increases proportionally with the increase in the size of power system. Although there exist a rich literature for distributed simulations, these are not practical to perform on wide area network in real time because of the computer network limitations. Most of the time, this type of distributed simulations were performed locally with different limitations imposed. Even though parallel algorithms provide rapid calculations, they are mainly based on

matrix operations and use “divide and conquer” methods to solve the system. Parallel algorithms and different kinds of interfacing techniques were introduced in Chapter 2 along with their features.

Interfacing distributed resources has always been a great challenge, which has different challenges, mainly in the area of data communication. The problem of exchanging data between utilities is still being researched because of confidentiality, interfacing and bandwidth limitations. Presently, there are not many practical systems which can model the data transfer concepts between different power system hardware and devices at remote locations.

### **3.3 Proposed Work**

The proposed work mainly consists of the investigation and analysis of different tools, methods and schemes for the distributed simulation. Initially, this work focused on identifying different tools and ways for performing distributed simulation followed by the development of a communication test bed for two real time simulators. Different data transfer experiments will be performed to calculate the data rates between the geographically distributed subsystems to understand the network behavior. Then, the different simulation times for performing distributed simulation locally are calculated and analyzed. This research also analyzes the different possibilities and limitations of performing distributing simulation in a wide area network. Next, a Wide Area Monitoring and Control test bed is developed to perform distributed, coordinated control. Here the power system is simulated in coupled and decoupled fashions and the simulation times are calculated. These models are developed in software called RSCAD and

implemented in time domain simulation hardware known as the Real Time Digital Simulator (RTDS). RSCAD is the software suite for the RTDS in real time.

### **3.4 Approaches to Problem**

In this work, the RTDS is used to model and analyze the power system. All the studies and simulations are performed on the RTDS in real time using the RSCAD software. The network solution of RTDS is based on the H.W. Dommel's digital computer solution algorithm for electromagnetic transients. The overall network solution in the RTDS is based on nodal analysis [18]. Dommel's EMTP method is mainly based on the method of characteristics and trapezoidal integrator. Trapezoidal integration is used for the differential equations because of its stability, simplicity and accuracy under most of the conditions. This type of EMTP method simulates the transients accurately in the network, which involves distributed and lumped parameters. Dommel had utilized trapezoidal integration rule to derive resistive companion network in each time step of the solution. Norton equivalents of each element consisting of a resistor and a current source in parallel were integrated into a nodal conductance matrix and injected current vectors to determine the complete system solution [1].

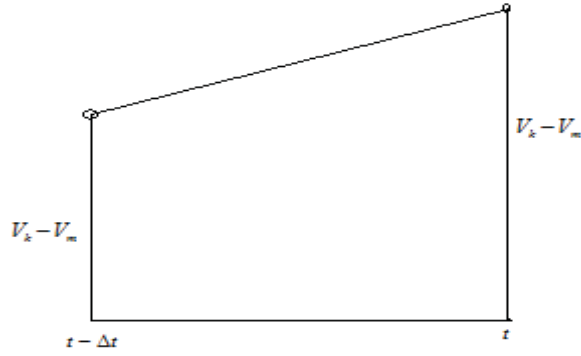


Figure 3.1

Trapezoidal Rule [1]

The Norton equivalent of resistance, capacitance and the inductance can be derived using the trapezoidal rule shown in figure 3.1. Trapezoidal rule is mainly derived from the Newton-Cotes formula that states that if one can approximate the integrand as  $n^{\text{th}}$  order polynomial then the integral of that function is approximated by the integral of that  $n^{\text{th}}$  order polynomial [24].

As an example, the equations and diagram below demonstrate how to represent a resistance with its Norton Equivalent form.

$$v_k(t) - v_m(t) = Ri_{k,m}(t) \quad (3.1)$$

(or)

$$i_{k,m}(t) = (1/R)(v_k(t) - v_m(t)) \quad (3.2)$$

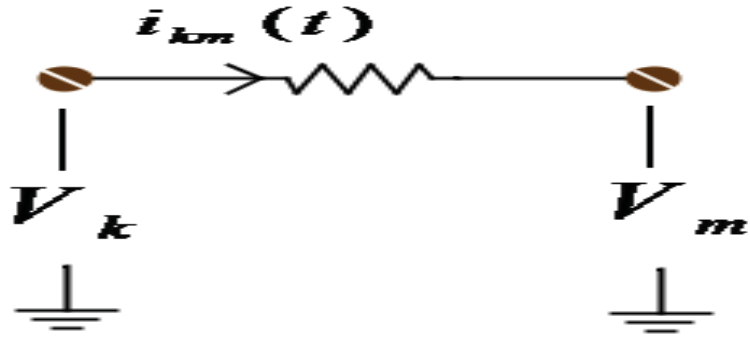


Figure 3.2

Representation of Resistor

The differential equation for the inductance (Figure 3.3) is given by

$$v_L = v_k - v_m = L \frac{di_{km}}{dt} \quad (3.3)$$

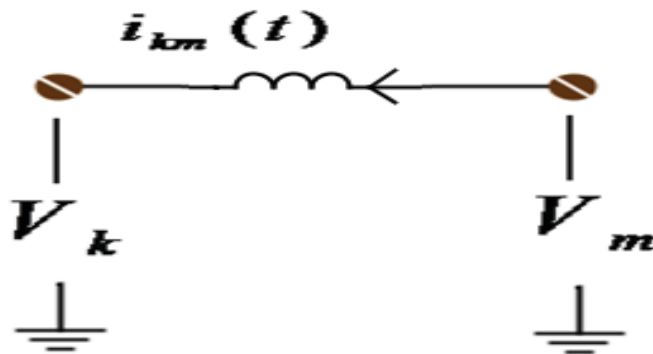


Figure 3.3

Representation of Inductor

The above equation can be integrated from a known state to the unknown state as

$$i_{km(t)} = i_{km(t-\Delta t)} + \frac{1}{L} \int_{t-\Delta t}^t (v_k - v_m) dt \quad (3.4)$$

Applying the trapezoidal rule for the above equation provides

$$i_{km(t)} = i_{km(t-\Delta t)} + \frac{\Delta t}{2L} (v_{k(t-\Delta t)} - v_{m(t-\Delta t)}) + \frac{\Delta t}{2L} (v_{k(t)} - v_{m(t)}) \quad (3.5)$$

From this equation, the Norton equivalent of the inductor can be drawn and is shown in Figure 3.4.

$$i_{km(t)} = I_{history(t-\Delta t)} + \frac{1}{R_{eff}} (v_{k(t)} - v_{m(t)}) \quad (3.6)$$

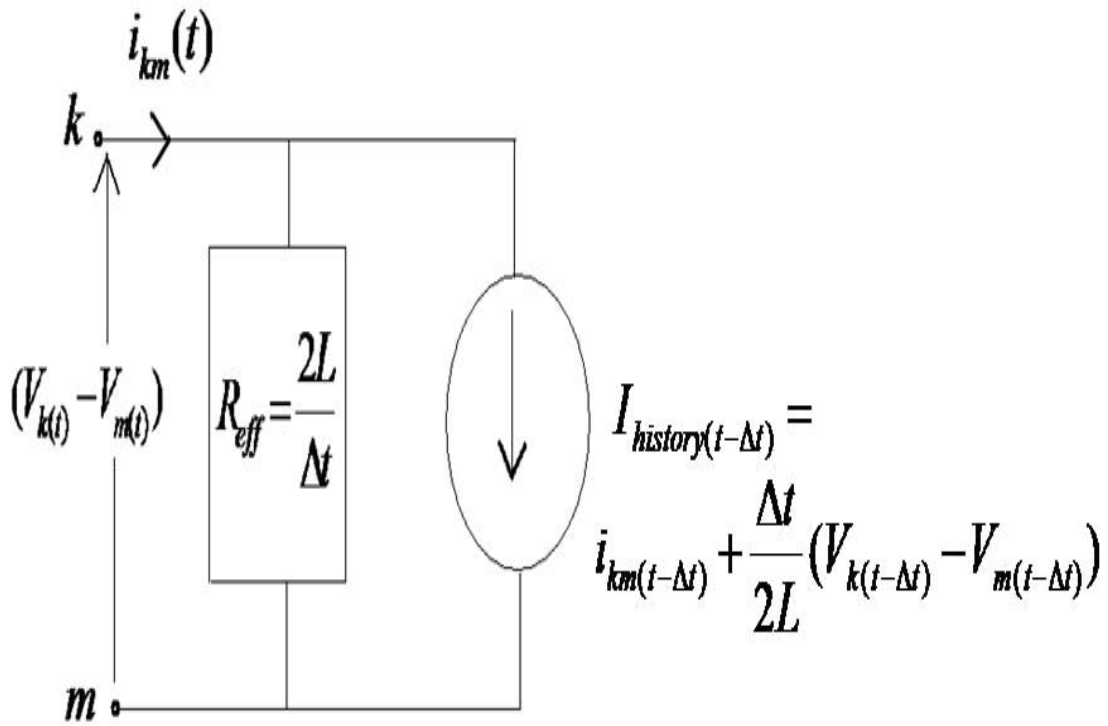


Figure 3.4

#### Norton Equivalent of an Inductor [1]

The  $1/R_{eff} = G_{eff}$  is the instantaneous term that relates the present time voltage to a present time current contribution, and involves a pure resistance [1]. The current source, shown in the Norton equivalent is a function of the quantities from the previous time step.

In a similar way, the equations and the Norton equivalent for the capacitor can be derived. A capacitor is shown in Figure 3.5.



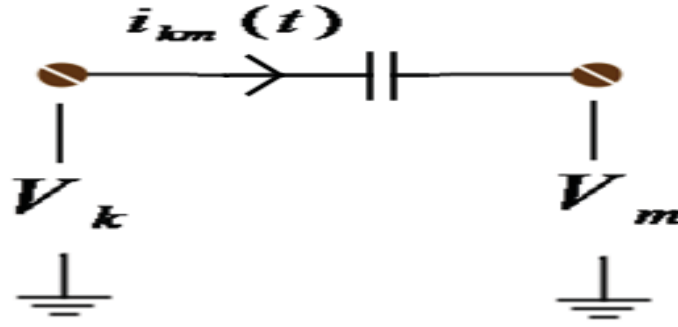


Figure 3.5

### Representation of Capacitor

The differential equation for the capacitor is given by

$$i_{km} = C \frac{dv_{km}}{dt} = C \frac{d(v_k - v_m)}{dt} \quad (3.7)$$

(or)

$$v_{km(t)} = (v_{k(t)} - v_{m(t)}) = (v_{k(t-\Delta t)} - v_{m(t-\Delta t)}) + \frac{1}{C} \int_{t-\Delta t}^t i_{km} dt \quad (3.8)$$

Applying the trapezoidal integration to the above equation provides

$$v_{km(t)} = (v_{k(t)} - v_{m(t)}) = (v_{k(t-\Delta t)} - v_{m(t-\Delta t)}) + \frac{\Delta t}{2C} (i_{km(t)} + i_{km(t-\Delta t)}) \quad (3.9)$$

The Norton equivalent of a capacitor is shown in Figure3.6. The capacitor current is given by,

$$\begin{aligned}
i_{km}(t) &= \frac{2C}{\Delta t} (v_{k(t)} - v_{m(t)}) - i_{km(t-\Delta t)} - \frac{2C}{\Delta t} (v_{k(t-\Delta t)} - v_{m(t-\Delta t)}) \\
&= \frac{1}{R_{eff}} (v_{k(t)} - v_{m(t)}) + I_{history}
\end{aligned} \tag{3.10}$$

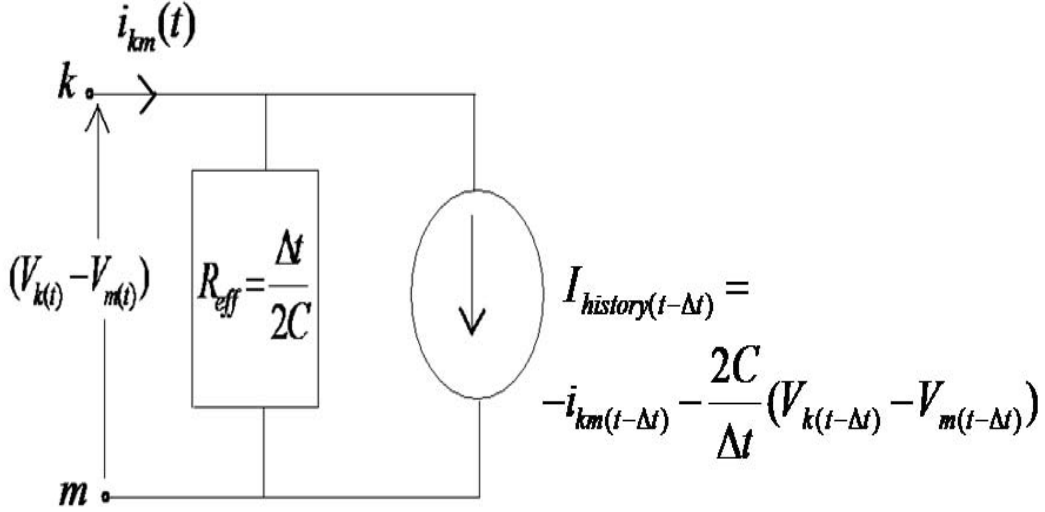


Figure 3.6

Norton Equivalent of Capacitor [1]

To reduce the computation at each time step, the Norton equivalents of all the components can be merged together, which automatically reduces the number of nodes. Figure 3.7 shows the reduction of a RLC branch into a single Norton equivalent. For obtaining the network solution of the power system model in RTDS, representing all the network components in their Norton equivalents, will result in conductance matrix from the nodal equations. Solving the resulting set of algebraic equations will result in the solution of power system in RTDS.

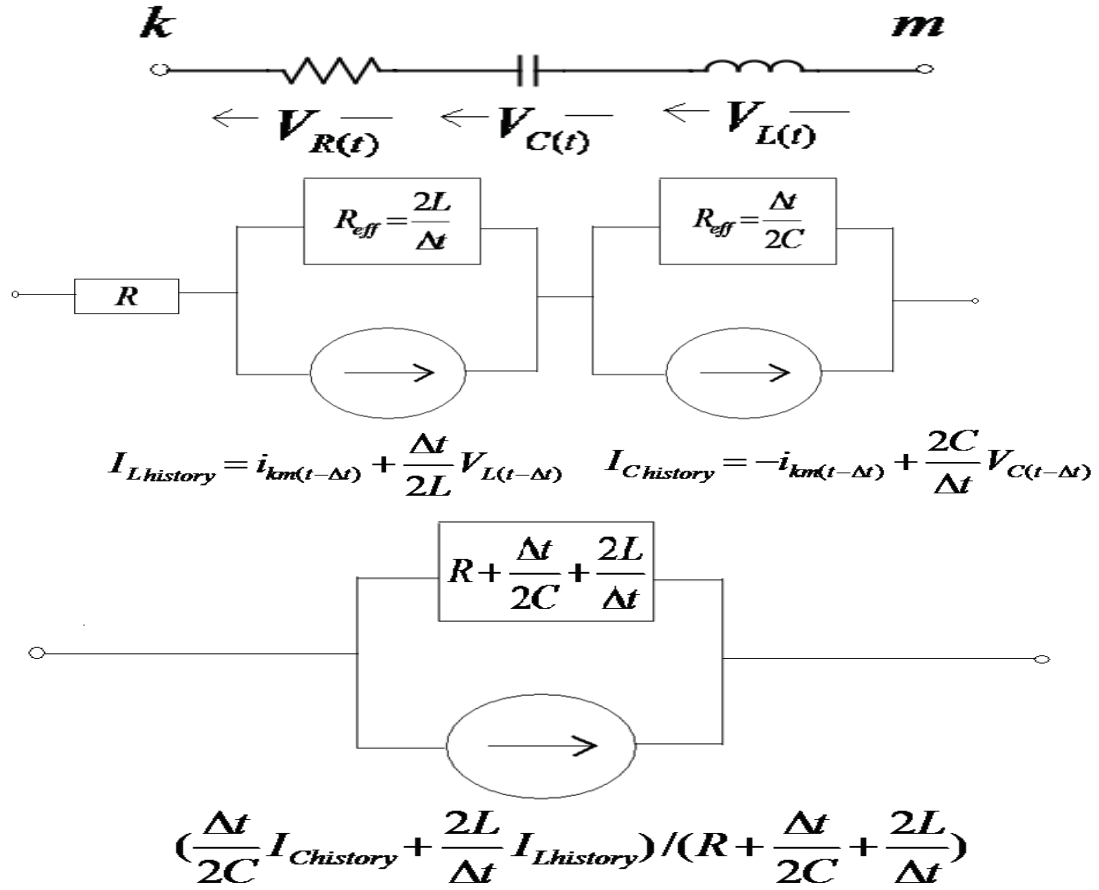


Figure 3.7

Reduction of RLC branch to a Norton Equivalent [1]

### 3.5 Work Plan

The following steps briefly outline the work flow of this research.

- Investigation of the available hardware and software capabilities for distributed simulation using RTDS and proposal of a schematic for communication between simulation platforms.

- Investigation of different methods for an extrapolation or predictor algorithm to deal with data loss and address the challenges involved in geographically distributed simulation.
- Development of a communication test bed between two real time simulation platforms and study of the network characteristics. The study involves the analysis and calculation of two main important characteristics of the network, which are bandwidth and latency.
- Validation of the communication test bed between the simulation platforms at MSU and TAMU by data transfer and remote monitoring experiments using LabVIEW and C# programs.
- Development of a Wide Area Monitoring and Control test bed for coordinated distributed control and analysis of locally distributed simulation.
- Analysis of decoupled and non decoupled simulations locally based upon their simulation times and resource usage using the Inter rack communication (IRC) card and Workstation Interface Card of RTDS.

### 3.6 Summary

In this chapter, the problem associated with the existing methods and the proposed work to address these has been described. The problem approach in this work mainly revolves around the network solution architecture of RTDS. Later the trapezoidal integrator was discussed followed by the Norton equivalent circuit modeling for the basic power system elements such as resistance, capacitance and inductance and their combination. The importance of these equivalent circuits comes into the picture when the

nodal analysis of the Real Time Digital Simulator network solution is considered. Finally, the chapter concludes with the detailed steps that are involved in the work flow of this research work.

## CHAPTER IV

### RESEARCH CHALLENGES AND ANALYSIS

This chapter presents the options for investigation by analyzing the available hardware and software capabilities at Mississippi State University Power and Energy Research Lab (PERL) for performing wide area distributed simulation. Different methods to deal with data loss are also discussed followed by some of the research challenges involved.

#### **4.1 Analysis of Available protocols in the RTDS**

One of the important aspects of distributed simulation is the communication between the subsystems. Since the RTDS is used in this work, different ways of sending and receiving the data needs to be explored.

##### **4.1.1 Analysis of sending the data out of RTDS**

Initially, different ways of sending the data out of the RTDS needed to be explored. In this work, the GTNET has been selected for sending the data out of the RTDS as it is the high transfer rate communications card which transfers large amounts of data accurately at a faster rate. The amount of data transferred and the rate, at which it

is transferred, depends on the installed firmware. GTNET has four protocols available which are GSE (Generic Substation events), PLAYBACK (Data playback from simulation results), DNP (Distributed Network Protocol) and SV (sampled values) [18]. The specific protocol can be used by installing its corresponding firmware on the GTNET. Only one firmware can run at a time on the GTNET hardware.

Basically GTNET\_GSE supports IEC 61850 GSSE and IEC 61850 GOOSE protocols. This firmware interfaces 32 binary inputs and binary outputs from up to five unique external devices such as relays. These inputs and outputs are mainly meant for the trip and status signals. The GTNET\_GSE component can be configured to send and receive GSSE or GOOSE messages containing the status of 32 binary input and output points from five unique external Intelligent Electronic Devices (IEDs). The major disadvantages of this protocol are the GSSE messages as multicast packets. These cannot be routed over WAN as both of them are layer 2 protocol and IEC61850 GOOSE/GSSE is designed specifically for very high-performance (4ms) in LAN for peer-to-peer messaging. In order to overcome this, it has to be mapped onto a higher layer protocol which has IP capability. Using Virtual LAN (VLAN) or VPN, it is possible to make this message go over WAN. But it is not suitable as it involves a lot of problems such as router constraints, security limits and managerial overhead. Also in these protocols/messages, the lost data cannot be tracked since there is no acknowledgement involved. The IEC GSSE message represents only the control signals and state of the system/component in the digital form. The main reason why one cannot use this protocol is due to the fact that these IEC GOOSE messages are sent from the RTDS only when

there is a change of state or with the use of a specified preset interval. Due to such nature of output, the user does not have any control over the output of GTNET GSE firmware.

The GTNET SV supports the IEC 61850-9-2 sampled value protocol. The sampled value protocol continuously sends the sampled values of analog signals (a maximum of 8) at two distinct rates, which are configurable at the rates shown below [18].

- 80 samples per nominal system frequency cycle
- 256 samples per nominal system frequency cycle

The sampled valued messages, which are sent out of GTNET, are multicast messages and these cannot be routed over the wide area network. Hence, there is a need to figure out a way of mapping that data to a higher layer protocol for sending it through wide area network. But these messages seem to be useful as these are being sent continuously irrespective of any changes in the system. The Playback firmware is mainly used to playback the captured waveform data stored in ASCII data files at an arbitrary sample rate in the same manner as a real time playback system [18].

DNP is an IP based protocol which is mainly used for communication between RTUs (Remote Terminal units) and IEDs (Intelligent Electronic Devices) to SCADA systems. The DNP protocol, available in RTDS may not be useful for distributed simulation until one has a master station for it. Basically the DNP protocol in RTDS is a slave and it can only communicate with a Master. The DNP firmware in RTDS is not feasible because it is not possible to communicate between two slaves (two RTDS's) unless a master station is introduced in the loop between the simulators. There is no



documentation which supports peer to peer communication between slaves and also that the RTDS can be configured as a DNP master.

Hence, GTNET SV was used for sending the data out of RTDS in the work. Currently, at MSU PERL, the GTNET SV in RTDS can send sampled values of 4 currents and 4 voltages at an IEC configured rate of 80 samples/cycle. This firmware will be helpful if the data is being sent through a virtual LAN on the wide area but one can also encapsulate it over a higher level of protocol for sending it through WAN. The data samples can be time stamped the data samples coming out of the GTNET SV with an external or internal 1PPS signal where the internal 1 PPS signal will be generated by the RTDS internal clock.

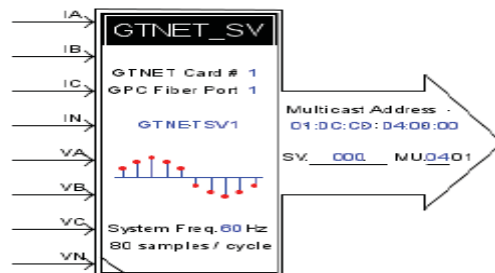


Figure 4.1

#### GTNET Sampled Values Component in RSCAD [18]

Figure 4.1 shows the GTNET SV block diagram, where it accepts four current and four voltage signals at a system frequency of 60Hz with an output rate of 80 samples/cycle. It also shows the multicast address of the device which receives the IEC 61850-9-2 sampled values.

#### 4.1.2 Analysis of getting the data into RTDS

After getting the data through the WAN on the other side, one needs to analyze the different possibilities of sending the data into RTDS. Once the data is in the RTDS, one can use it to perform simulations. But it is required to ensure that one gets the data into the RSCAD Draft but not the Runtime. Below are some of the ideas for sending the data into RTDS.

##### 4.1.2.1 Using GTNET Playback Firmware

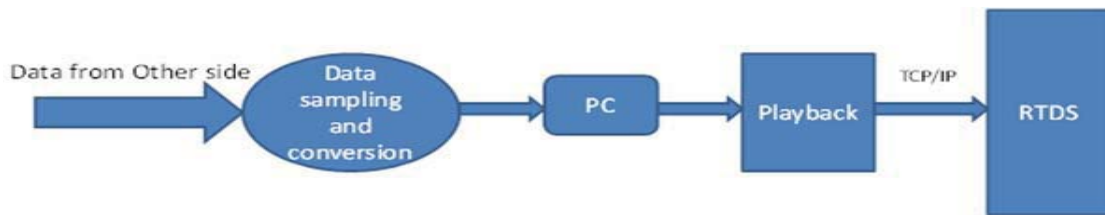


Figure 4.2

GTNET Playback for getting the data into RTDS

Figure 4.2 shows the option of getting the data into RTDS using the Playback firmware where the data sampling and data conversion have to be performed independently. In this work, the researchers plan to use the RTDS GTNET Playback firmware, which is mainly used to playback the simulation output data into the Runtime. Some of the advantages of this schematic are:

- Playback's capability of injecting voltages and currents in Runtime.
- It is based upon TCP/IP protocol.

But it also has some of the disadvantages for distributed simulation application, such as

- Data conversion needs to be done for the other subsystems data
- The received data should be written into a separate ASCII file with a specific format.
- Data needs to be buffered before playing back into the system.

#### 4.1.2.2 Using National Instruments PXI

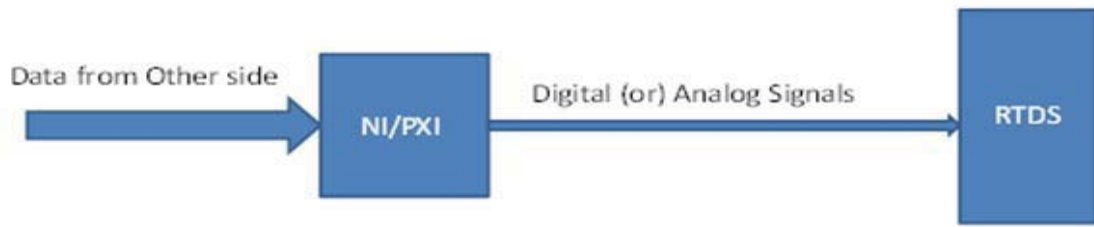


Figure 4.3

NI PXI for getting the data into RTDS

The NI PXI can be used to reduce much communication overhead as it works with LabVIEW, which supports many communication protocols. The NI PXI also performs many signal and data processing operations such as data conversion, data filtering and signal sampling.

There are also other options, such as scripting in RTDS to get the data into Runtime, but it has many limitations with respect to distributed simulation. Scripting is capable of performing data operations using TCP/IP and Modbus protocols. In distributed simulation using RTDS, one needs to get the power system data into the draft so that it affects the simulation directly instead of affecting the simulation results in Runtime. With

the limited available options, the NI PXI was chosen to serve the purpose of capturing and sending the power system data into the RTDS available from the other subsystem through the Wide Area Network.

Hence, the GTNET SV has been used to send the data out of RTDS and NI PXI to receive and send the data into the RTDS.

#### **4.2 Interface Algorithm and Numerical Methods**

The distributed simulation algorithm is presented here with the methods for extrapolating the data under the data loss situations during the data transfer between subsystems. An interface model is required to deal with the issues of delay and data transfer in the RTDS to RTDS communication. This interface model can be built using the RSCAD, which has a special feature for building user defined components known as C Builder. C Builder is very user friendly and it can be used to build user defined custom models on GPC cards. The GPC cards can be efficiently used as the GTNET is also connected to it. In C Builder, the component can be drawn in different modes such as single line and three phase representations and initialize the parameters with user defined inputs and outputs. C coding is used to define the new model and it is converted to an executable code by the special compiler present in C Builder. It provides a faster way of compilation. The main advantage of the C Builder is modification and enhancement of the component model for future developments until a final version is achieved.

The interface model, built using C Builder requires an algorithm for estimating the data if there is a data loss during the communication. The proposed algorithm for the distributed simulation is shown below in the form of a flow chart in Figure 4.4. It

represents the calculation and exchange of values between the two systems shown in Figure 2.5.

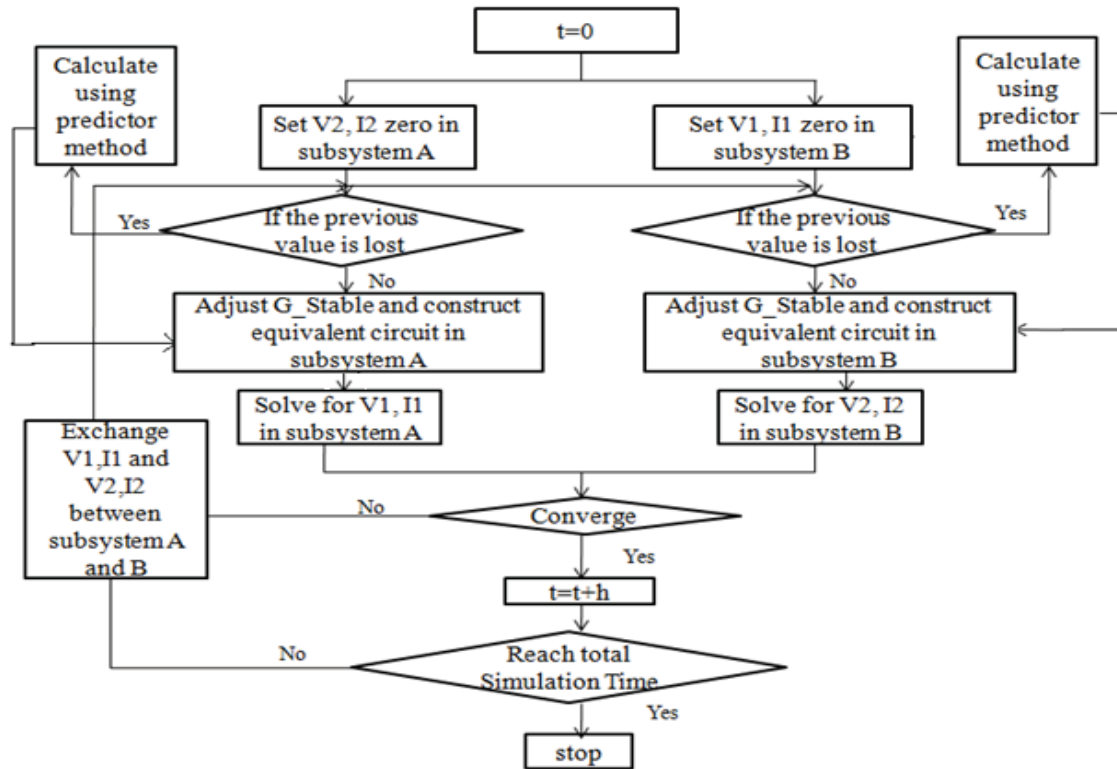


Figure 4.4

#### Interface Algorithm for Distributed Simulation using Predictor Method

##### 4.2.1 Numerical Solutions

One of the most important aspects, during data communication is the data loss. If data loss occurs during the communication, then the models cannot be run on either side as they are running in real-time. So to prevent the data loss one needs to find a way of providing estimated data for the time until the real data is available. The estimated values will be corrected after getting the updated data.

One of the methods, used in the literature is the extrapolation method. In this method, the basic approach is to solve the differential algebraic equations with an elementary technique using different step sizes and then use Richardson extrapolation to estimate the solution in limit as the step size approaches zero. Since extrapolation requires multiple solutions of the differential equations, computationally efficient techniques can be used for calculating the initial solutions. The modified midpoint method has been used for the initial estimates and then the Richardson extrapolation method for more accurate results. The main advantage of this method is that it requires less computational effort, when the solution is not needed at several points. This method is outlined below. Here the interval is given by  $(\alpha, \beta)$  and  $h$  represents the length of the interval.

$q^0$  represents the initial state and the number of steps is given by  $m$ . The interval is partitioned into  $m$  steps of equal length, given by  $h = (\beta - \alpha) / m$ . Euler's method for the first step gives:

$$q^1 = q^0 + hf(\alpha, q^0) \quad (4.1)$$

So with an estimate of  $q^1$ , midpoint method is used to obtain estimated solutions at the remaining points. An equation can be obtained using two past solutions as,

$$q^{k+1} = q^{k-1} + 2hf(\alpha + kh, q^k) \quad (4.2)$$

It is integral over an interval of length of  $2h$  using the slope of  $x(t)$  at the midpoint  $t = \alpha + kh$ . Finally, an end point correction can be performed given by  $x''$  using the last two points. It represents the truncation error of the estimate. This method requires  $m+1$  function evaluations to compute the truncation error. The truncation error

can be shown to be of the order  $o(h^2)$ . The truncation error can be represented by an even power series given by:

$$x(\beta) = x^m + \sum_{j=1}^{\infty} \gamma_j h^{2j} \quad (4.3)$$

Here  $\gamma$  is the safety factor considered to be conservative regarding the error and it doesn't depend upon  $h$ . The advantage of this technique is that it converges very rapidly missing the odd terms in the truncation error. The modified midpoint method calculations described above is incorporated with the extrapolation method. Midpoint method produces an estimate  $x^m = x(\beta)$ .

This estimate is based on  $m$  steps of length  $h$ .  $A_{k1}$  denotes the midpoint estimate of  $x(\beta)$  using  $m_k = 2^k$  steps. The truncation error estimate of  $A_{k1}$  is given by  $E_{k1}$ .

It is as following:

$$A_{k1} + E_{k1} = A_{k-1,1} + E_{k-1,1} \quad (4.4)$$

Using the form of truncation error:

$$E_{k1} \approx \gamma_1 h_k^2 \text{ where } h_k = h_{k-1} / 2 \quad (4.5)$$

The relationship between successive truncation errors is given by  $E_{k1} \approx E_{k-1,1} / 4$ .

Thus the truncation error:

$$E_{k,j-1} \approx \frac{A_{k1} - A_{k-1,1}}{3} \quad (4.6)$$

This can be used to construct the new extrapolated estimate given by

$$A_{k2} = \frac{4A_{k1} - A_{k-1,1}}{3}, 2 \leq k \leq p \quad (4.7)$$

The linear combination of  $A_{k1}$  and  $A_{k-1}$  increases the order of the new estimate to  $O(h^4)$ . The generalized formula for the relationship between successive truncation errors is given by

$$E_{k,j-1} \approx \left(\frac{1}{2}\right)^{2(j-1)} E_{k-1,j-1} \quad (4.8)$$

Here, k represents the number of iterations or time steps and p represents the level of extrapolation. This technique converges very rapidly by missing the odd terms in the truncation error estimate. There are different methods for extrapolating the value such as predictor method and trapezoidal rule based method. It requires less computational effort with respect to the other methods when the solution is not needed at different points. The algorithm for the extrapolation method is:

1. Pick  $\varepsilon > 0$  and  $r \geq 2$
2. Set  $k = 0$  and  $q^0 = a$ .
3. Do
  - {
  - (a) Set  $k = k+1$ ,  $m = 2^k$  and  $h = (\beta - \alpha) / m$ .
  - (b) Compute
    - {

$$q^1 = q^0 + hf(\alpha, q^0)$$

$$q^{j+1} = q^{j-1} + 2hf(\alpha + kh, q^j), 1 \leq j < m$$



$$x^m = \frac{q^m}{2} + \frac{q^{m-1} + hf(\beta, q^m)}{2}$$

}  
(c) If  $k > 1$  then compute

And

$$A_{kj} = \frac{4^{j-1} A_{k,j-1} - A_{k-1,j-1}}{4^{j-1} - 1}, 2 \leq j \leq k$$

$$E_{k,j-1} \approx \frac{A_{k,j-1} - A_{k-1,j-1}}{4^{j-1} - 1}$$

}}

4. While  $(k < 2)$  or  $[(\|E_{k,k-1}\| > \max\{\|A_{kk}\|, 1\}\varepsilon) \text{ and } (k < r)]$ .

Here,  $\varepsilon$  is the desired upper bound on the size of local truncation error and  $r$  is the maximum extrapolation level.

### 4.3 Challenges Involved

In this section, some of the challenges during the development of geographically distributed test bed are discussed. Along with the challenges involved in performing this distributed simulation using WAN. These are classified as given below.

#### 4.3.1 High output rate of GTNET SV and LabVIEW modeling

The output of GTNET SV component in the RTDS is 80 Samples/Cycle. It can be either 80 samples or 256 samples according to the IEC 61850 standard. IEC 61850 is the standard for substation automation design. The rate of 80 Samples/Cycle corresponds to a data rate of 4.53 Mbps. The calculations are shown below.

$$80 \text{ samples/cycle} = 4800 \text{ samples/second} \quad [1 \text{ cycle} = (1/60) \text{ second}]$$

$$4800 \text{ samples/sec} = 566400 \text{ bytes/second} \quad [1 \text{ sample} = 118 \text{ bytes}]$$

$$566400 \text{ bytes/second} = 4531200 \text{ bits per second} \quad [1 \text{ byte} = 8 \text{ bits}]$$

It is tough for any platform to capture data in the micro-second range without losing or dropping any data. Micro-second range data operations are mostly classified under Real Time and which is not possible for any program running on the Non - Real Time Windows platform of Microsoft. It can only be possible with extensive programming and highly compatible platform. Achieving this GTNET data capture using the LabVIEW program on the Windows platform had so many issues. These issues were resolved with proper modeling of data capture program in LabVIEW.

#### 4.3.2 Time stamping and synchronization of simulators

For performing distributed simulation over the WAN, one of the important issues that needs to be considered is synchronization of the simulators on either side since RTDS simulation is a time step based simulation. Distributed simulation requires either of the simulators to start at a synchronized time for proper operation and exchange of data between the subsystems. Currently there is no option available where the simulator time step can be synchronized with a satellite synchronized clock. So the only option left

is to synchronize the GTNET SV at either end with a GPS (Global Positioning System) clock, so that the data transfer between the subsystems is synchronized.

RTDS has an option to either generate a 1PPS input signal for the GTNET or an external 1PPS signal can be given to RTDS GTNET to synchronize RTDS (only output of the GTNET data) to that external source. MSU has performed a study experiment to differentiate the internal and external 1PPS signal. Details of that experiment are mentioned below. A study experiment was conducted on the GTNET using external and internal 1PPS signal. The data packets coming out of GTNET were studied in both cases. This analysis was carried out using Wireshark. With either external or internal 1 PPS signal, the numbers of packets were never ending at 4799 and were higher than 4800 in 1 second. This means that in either case, GTNET output was not exactly 4800 packets per second. But with an external 1 PPS signal from SEL – 2407, the error was less as compared to that with the internal 1 PPS signal. The number of packets sent out in 1 second was higher when an internal 1PPS signal was used. Experimental results are shown in Tables 4.1 and 4.2.

Table 4.1

External 1 PPS signal for GTNET accuracy testing (from GPS [SEL-2407])

| Experiment Number                       | Number of packets | Time (Seconds) |
|---|-------------------|----------------|
| 1                                       | 4807              | 0.999820       |
|   | 4808              | 1.000018       |
| 2                                       | 4802              | 0.999826       |
|   | 4803              | 1.000025       |
| 3                                       | 4806              | 0.999966       |
|   | 4807              | 1.000218       |
| 4                                       | 4809              | 0.999768       |
|   | 4810              | 1.000020       |
| 5                                       | 4811              | 0.999971       |
|   | 4812              | 1.000212       |
| Average (number of packets after 1 sec) | 4808              | 1.0000986      |

Table 4.2

Internal 1 PPS signal for GTNET accuracy testing (generated within RTDS)

| Experiment Number                       | Number of packets | Time (Seconds) |
|---|-------------------|----------------|
| 1                                       | 4819              | 0.999871       |
|   | 4820              | 1.000071       |
| 2                                       | 4812              | 0.999877       |
|   | 4813              | 1.000075       |
| 3                                       | 4821              | 0.999865       |
|   | 4822              | 1.000070       |
| 4                                       | 4815              | 0.999870       |
|   | 4816              | 1.000070       |
| 5                                       | 4814              | 0.999867       |
|   | 4815              | 1.000063       |
| Average (number of packets after 1 sec) | 4818              | 1.0000698      |

From the above tables, it can be inferred that, with an external 1 PPS signal (GPS Synchronized), approximately 4808 packets are being received out of GTNET within 1

second and, with an internal 1 PPS signal, approximately 4818 packets are being received out of GTNET within 1 second.

#### **4.3.3 Transmission line model**

The RTDS is designed in such a way that the travelling wave transmission line model is used to connect two subsystems between racks. The RTDS uses Bergeron's and frequency dependant transmission line models represent travelling wave models. Distributed Simulation over the WAN requires a transmission line model of length around 6000 km depending upon the time taken for data transfer between the locations of the two subsystems. This work involved performing distributed simulation between Mississippi State University and Texas A&M University. The time delay that exists between these universities is around 20 ms, which has been calculated using the ping command in MS Windows.

A travel time of 50 microseconds corresponds to 15 km of transmission line length at the speed of light [18].

#### **4.4 Summary**

To summarize, initially this chapter dealt with the analysis of different ways of getting the data in and out of RTDS. Different protocols and firmwares are discussed. Later, the interface algorithm has been presented with different numerical methods for dealing with data loss. Finally, the chapter concludes with the discussion on some of the challenges involved in performing local and geographically distributed simulation.

## CHAPTER V

### DEVELOPMENT OF A GEOGRAPHICALLY DISTRIBUTED TESTBED WITH SIMULATION RESULTS

#### 5.1 Introduction

In this chapter, initially the different schemes of getting the data in and out of RTDS are described. Then the proposed communication test using the RTDS is discussed. Later, LabVIEW modeling is explored with the design of client and server models for data operations with RTDS. Finally results from the experiments performed between MSU and TAMU are detailed. This chapter also discusses the results of wide area monitoring using GTNET of RTDS.

#### 5.2 Schematics of Data Operations

In this section, different schematics of getting the data in and out of the RTDS will be discussed using the RTDS GTNET, LabVIEW and NI PXI.

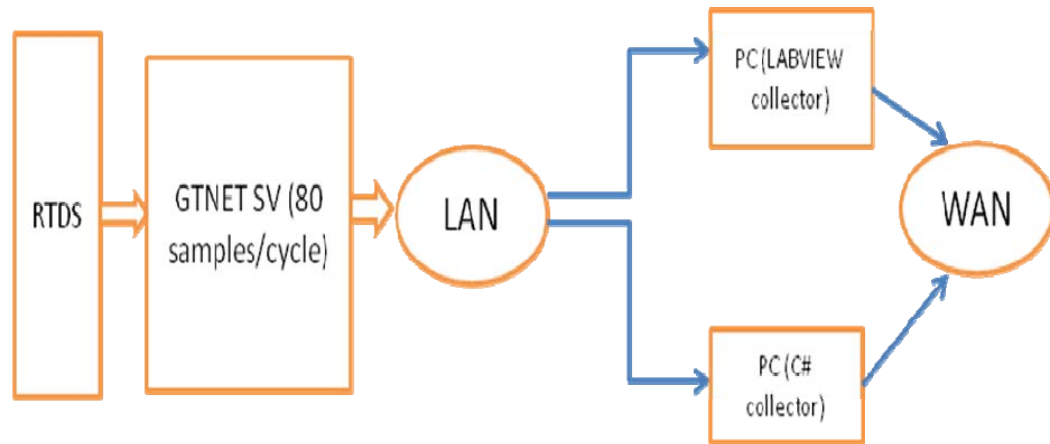


Figure 5.1

Sending the data out of RTDS into WAN using GTNET and LabVIEW

The RTDS sends out the node voltages and line currents through the GTNET SV at a rate of 80 samples/cycle. This data is sent out as Ethernet frames and it is captured in the Local Area Network. Since the data is in the IEC 61850-9-2 format, one needs a platform which can understand that format and decode the actual data before sending it into the RTDS on the other side. Hence, LabVIEW was chosen to be used for collecting and sending the data from one side to the other. The power system data from the GTNET SV was collected by a PC in which LabVIEW is installed. The detailed operation of the LabVIEW modeling is discussed later in this chapter. After collecting the data, it was encapsulated with the TCP/IP protocol to send it to the other side. The data from GTNET SV can also be captured using the C# platform which will be running on a PC. Basically, the LabVIEW program or C# program on the data collecting side will act as a server and will send the data to a client using TCP/IP protocol. In this work, predefined LabVIEW TCP/IP communication tools are used. Figure 5.1 shows the schematic diagram of getting

the data out of RTDS into the Wide Area Network. It shows how the data is collected in the LAN by a PC before encapsulating it with the TCP/IP protocol.

Once the data reaches the other side, the NI PXI is mainly used to get the data into the RTDS. This can be done in two ways. In the first method, one can capture the data by a PC which runs LabVIEW and decode the actual current and voltage signals before feeding it to the NI PXI. These digital signals then will be converted into analog signals by the NI PXI before being fed into the RTDS. In this method, the PC runs a LabVIEW client which corresponds to the LabVIEW server on the other side. Thus, it has the LabVIEW client and NI PXI running separately. In the second method, one can run the client LabVIEW file directly on the NI PXI in order to reduce the processing time of the captured data locally. In particular, the PXI 6733 analog output module has been used to generate the analog signals from the digital data of GTNET SV. The generated analog signals are fed as input to the high precision analog input card (OADC) of the RTDS, which has the feature of providing optical isolation to the input signals as shown in Figure 5.2. Generally, the OADC card is used to interface analog simulators to the RTDS. One interface technique uses a transmission line simulated on the RTDS where the line has one end connected to a bus represented within the RTDS and the other end connected to a bus represented in analog simulator [18]. In such cases, the OADC card converts the analog bus voltage signals of the other analog simulator into digital form so that these can be used in the simulation by RTDS.



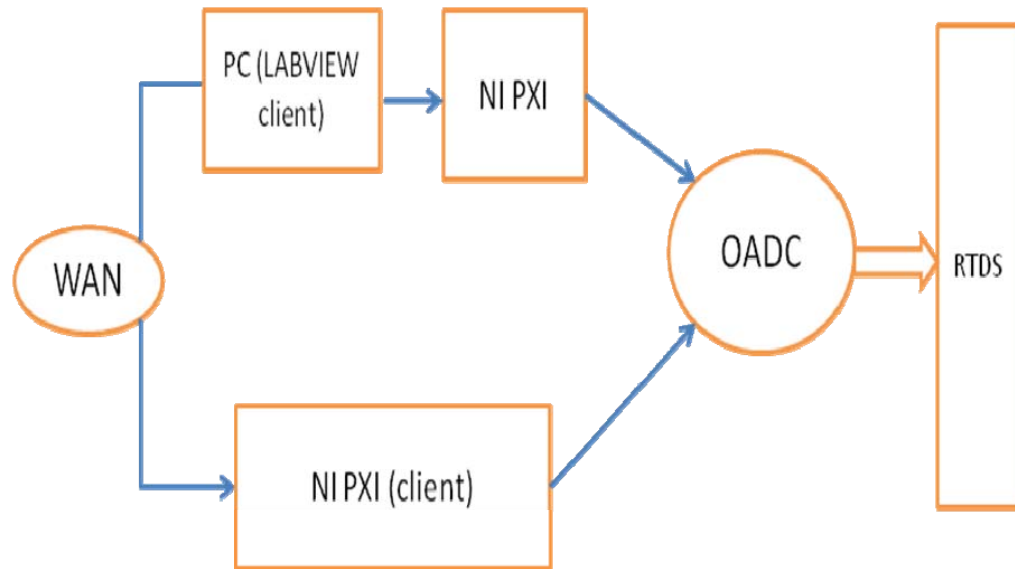


Figure 5.2

Getting the data into RTDS from WAN using NI PXI and OADC

### 5.3 Geographically Distributed Test Bed Schematic

The developed schematic of the communication test bed between the simulators is presented in this section. It shows the data flow between the two subsystems in different simulators, which are geographically separated.

As explained before, the power system data is initially captured by the PC1 using LabVIEW and transferred to the other side using TCP/IP. On the other side, the data is directly received by the LabVIEW client running on the NI PXI and is parsed and converted into analog signals before sending them into the RTDS.

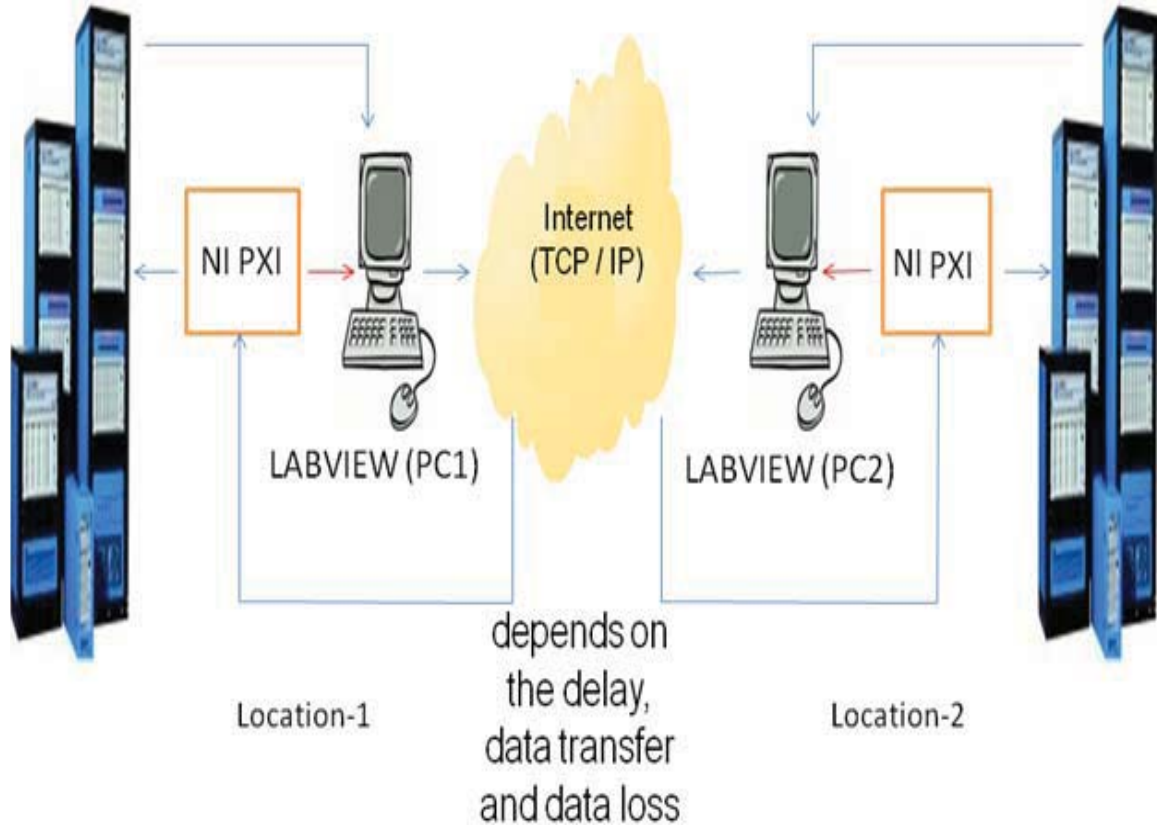


Figure 5.3

#### Proposed Schematic for RTDS-RTDS communication

Internet 2 is being used between the simulators and the time taken for data transfer depends on the network delay and data transfer speed. The LabVIEW computers, shown in Figure 5.3 act as GTNET data collectors and also as an interface to the NI PXI. A PC is required to see the functioning of LabVIEW client that runs inside the NI PXI in Real Time. After getting the power system data into the draft of the other RTDS, it can be used in the simulation using current injection and voltage read components. The current injection component is shown in Figure 5.4. The hardwired inputs are represented by wire labels in RSCAD and N13, N14 and N15 represent the current signals which are

read in through OADC card. IA, IB and IC represent the injected current values, which can be now used in the power system simulation.

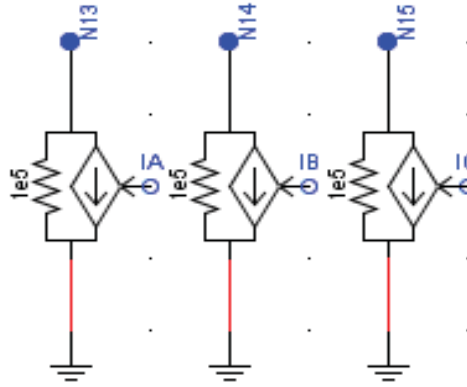


Figure 5.4

#### Three Phase Current Injection Component in RSCAD

In a similar way, the voltage signals which are read in from the OADC card can be fed as input to a voltage source. For example, if signals VA, VB and VC are read from the OADC, then these can be given as inputs, as shown in Figure 5.5. This CC (Controls Compiler) type of source is solved on the GPC card as well as 3PC. This model is a normal three phase source model with optional R/L or R-L zero sequence impedance whose input can be controlled by control signals. Basically these types of sources which are operated by controls compiler signals will permit modulation of both frequency and magnitude of the main source.

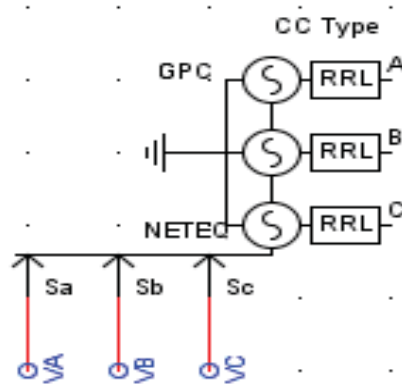


Figure 5.5

CC type Three Phase Voltage Source in RSCAD

#### 5.4 LabVIEW modeling

LabVIEW has played a vital role in the development of the communication test bed between the simulators. LabVIEW mainly serves following purposes.

1. Collecting the IEC 61850-9-2 data from GTNET SV in the LAN.
2. Encapsulating the captured data with TCP/IP protocol for sending it through WAN.
3. Receiving the data from WAN and decoding the signals by parsing out the required voltage and current signals.
4. Uses the LabVIEW Real Time for operating with DAQ mx card to produce analog outputs.

Details of all the steps mentioned above are as follows. This packet capturing model was developed by the National Instruments for capturing the basic Ethernet packets and this was extended in this work. This model was enhanced and extended to a large extent to capture the IEC 61850 data packets and parse the required power system

signals operating in Real Time with the NI PXI. The conversion of these digital signals was also included into analog data for input into the RTDS. Initially researchers evaluate the data capture in Local Area Network with the TCP/IP encapsulation of data. The front VI for the data capture LabVIEW file is shown in Figure 5.6. At the top of the VI, the list of adapters that are available on the PC will be displayed in a box. Depending on the application, one can choose the corresponding adapter to perform the data operations. Below the list of adapters, the raw power system data will be displayed in hexadecimal format, which will be a mixture of eight signals data corresponding to a total size of 64 bytes. The raw power system data will be continuously changing with respect to the data output by GTNET SV. The specifications of the data packet will also be displayed in the front VI, which will include all the packet information. It mainly includes the Ethernet and IP header details. Some of the Ethernet header details include destination MAC (Medium Access Control) address and source MAC address. The IP header details are the version, header length, flags, fragment offset, TTL (time to live), kind of protocol, header checksum with the source and destination IP address. One can also see the IP data and specify the port number at which the server waits to listen and accept a connection from the client. LabVIEW 8.5 version was used in this work. This VI mainly uses the Winpcap library to capture the network data. Winpcap is the standard tool for link layer network access in the windows environment. It is widely known as the tool for packet capture and network monitoring library for windows [28].

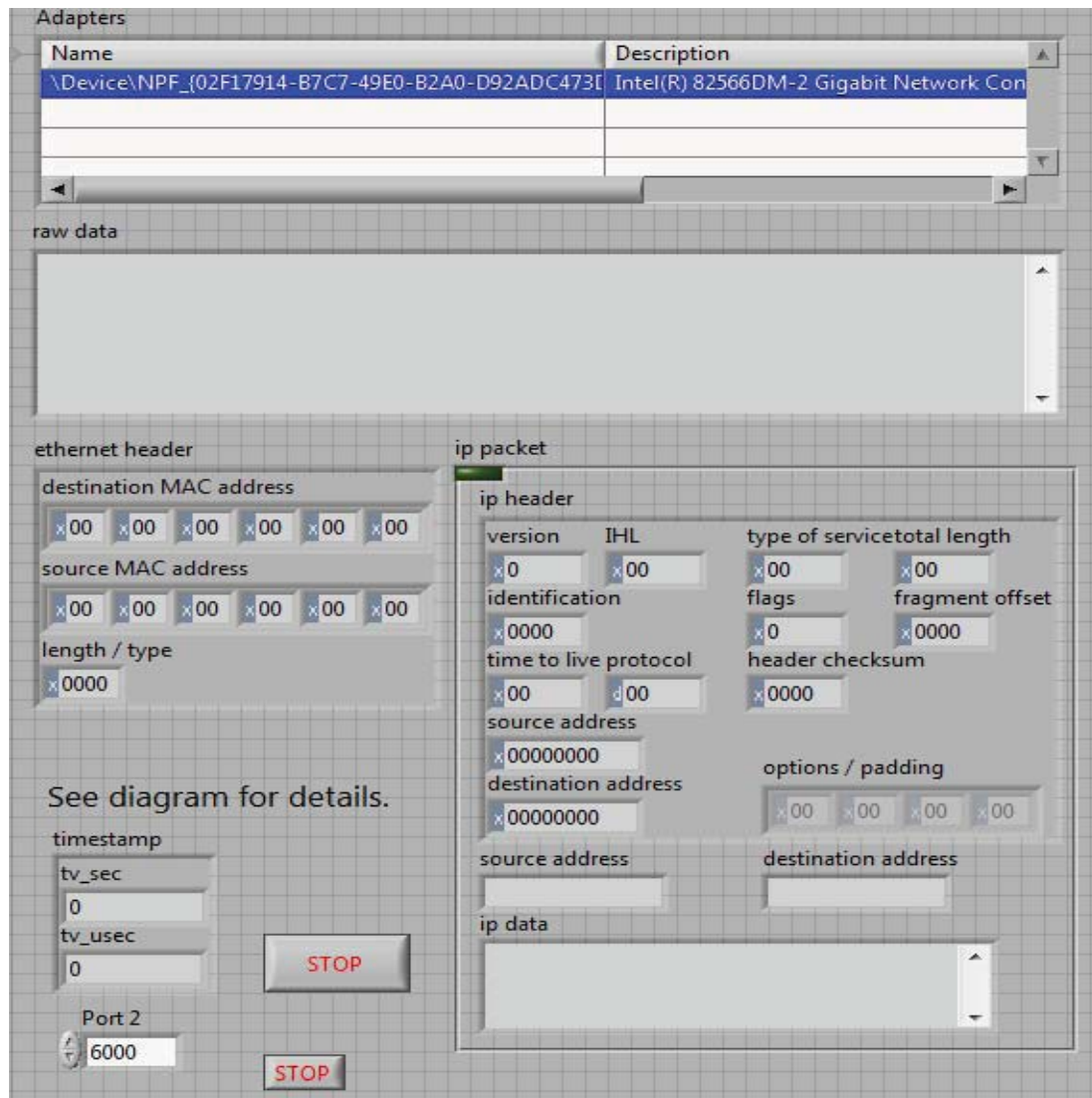


Figure 5.6

### LabVIEW Server Model with data capture Algorithm

The block diagram for the above VI is shown in Figure 5.7. It shows the wire connections and gives a brief insight to its operation.

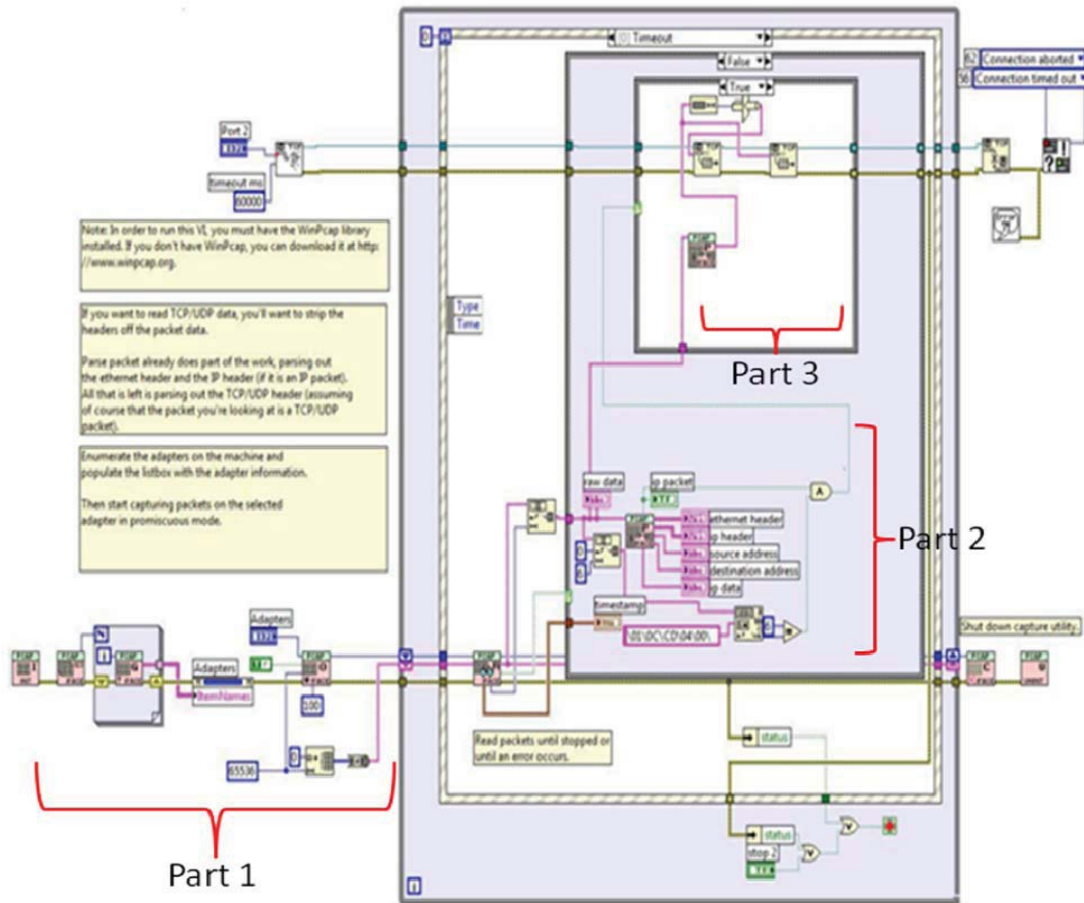


Figure 5.7

### Block Diagram of LabVIEW Server Model

Part 1 shown in Figure 5.7 is the initial step of data capture. The adapters are initialized and the interface is opened for data capture. Once the data capture begins, the captured data is transferred into Part 2 where the parse packet.vi separates the Ethernet header, IP header, source address, destination address and IP data. Here, the MAC address of GTNET is verified from each data packet. Only the packets containing GTNET MAC addresses are sent into the Part 3, where the data is being encapsulated by the TCP protocol to send it over the internet. The data encapsulation is being carried out

in a case structure, so that it can only operate when it gets a true signal from the AND operator after verifying that it is a IP packet and the MAC address of GTNET SV are true. In this way, the data enters the wide area network and reaches the other side where it will be captured by the LabVIEW client file, which will be running on the NI PXI in Real Time. For the client programming, LabVIEW Real Time is being used, but for the server programming Real Time is not being used. Server programming with LabVIEW Real Time will solve the problem of constant high output data rate from the GTNET SV, which is 80 samples/cycle, i.e. 4800 samples per second. To control this high output data rate, the data capturing (while sending itself) software was implemented in real time in NI PXI, but it attempted without any success. Hence, only NI PXI is used for data receiving purpose (on the client side) and not for sending purpose (on the server side). This is explained below.

The LabVIEW server file DLLs are not able to run on the LabVIEW Real Time. So, it cannot deploy the LabVIEW server model on the NI PXI system in real time. It shows an error as shown in Figure 5.8. The error is because of the third party application Winpcap which is compatible only with LABVIEW in non real time. The library in Winpcap is a shared library using DLLs. When the server VI is used in real time, the LABVIEW real time tries to access a Winpcap DLL which is a Windows linked application creating a problem. A utility has been tried, which was provided by NI in determining whether or not the function calls in the DLL are supported on the real-time operating system. The 8.5 Version of the utility was used that corresponds to the version of the LABVIEW Real-Time being used.



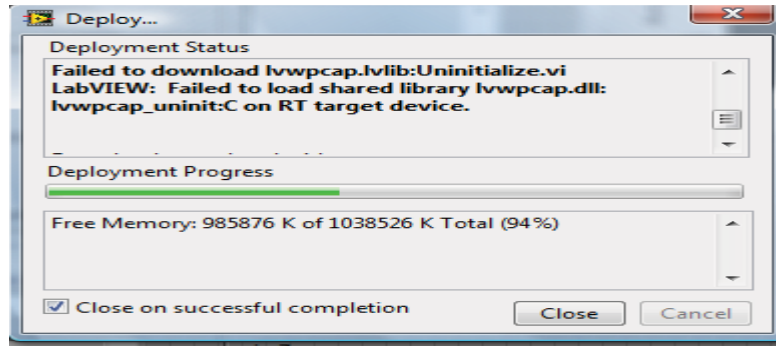


Figure 5.8

Deployment Status Error because of third party application

The utility showed that it needed additional dependent DLLs (Winpcap.dll) for running it and the additional DLL in the windows was not found. Figures 5.9 and 5.10 show the bad imports and stubbed imports of LabVIEW RT DLL checker, respectively. This concludes the investigation that server file cannot run in real time as the required DLLs are linked to the windows DLLs.

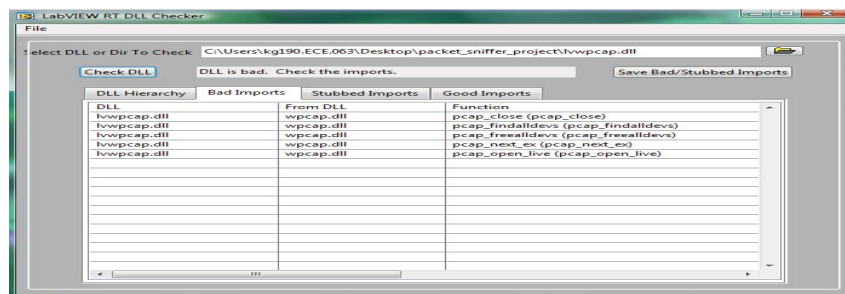


Figure 5.9

Bad Imports of LabVIEW RT DLL Checker

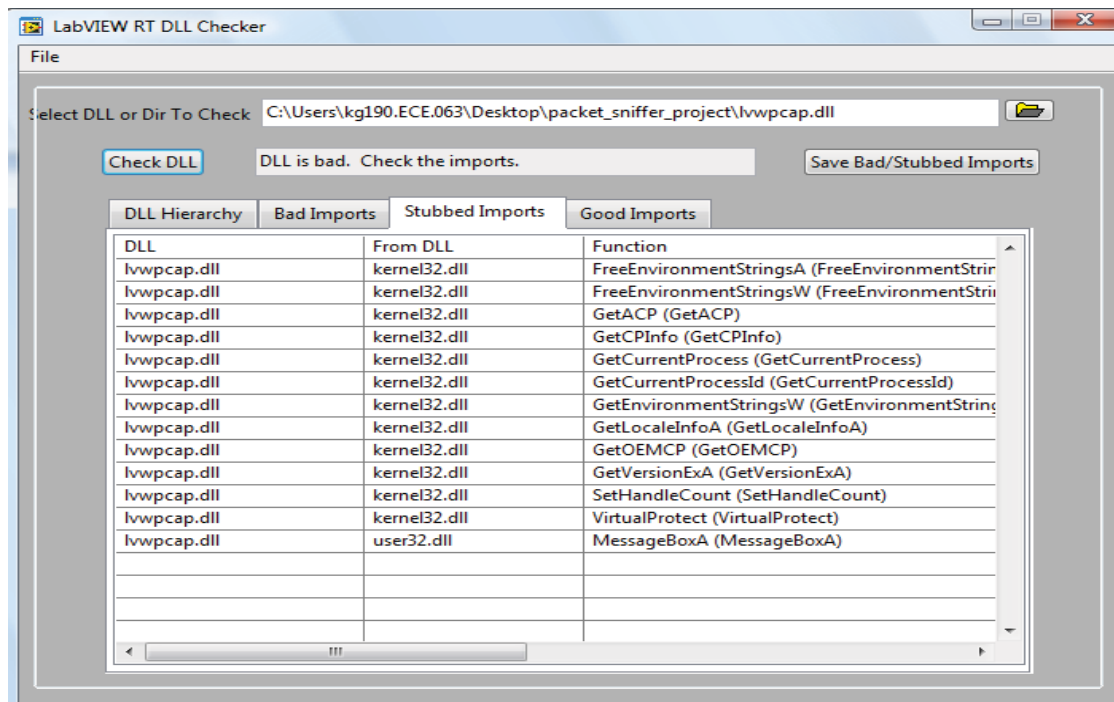


Figure 5.10

### Stubbed Imports of LabVIEW RT DLL Checker

The front VI for LabVIEW client file is shown in Figure 5.11. It shows the port number and IP address for its connection with the LabVIEW server on the other side. In this VI, the graphs and hexadecimal data of all the eight signals can be seen. The power system data that gets displayed during experiment is not time referenced. Hence, the x-axis of these graphs does not represent the time with respect to the signal. The VI runs on the NI PXI in real time converting the digital data into analog signals using the NI 6733 card. The block diagram of the client file is shown in Figure 5.12. It explains how the parsing of data is done for the eight signals by explaining the operation of FOR loop and the WHILE loop. It also explains the modeling of conversion of the digital TCP/IP data (in the form of a string) into analog data. The operational speed of this VI depends on the

processing capability of the hardware on which it is running. In the present setup, it depends on the processing capability of the NI PXI.

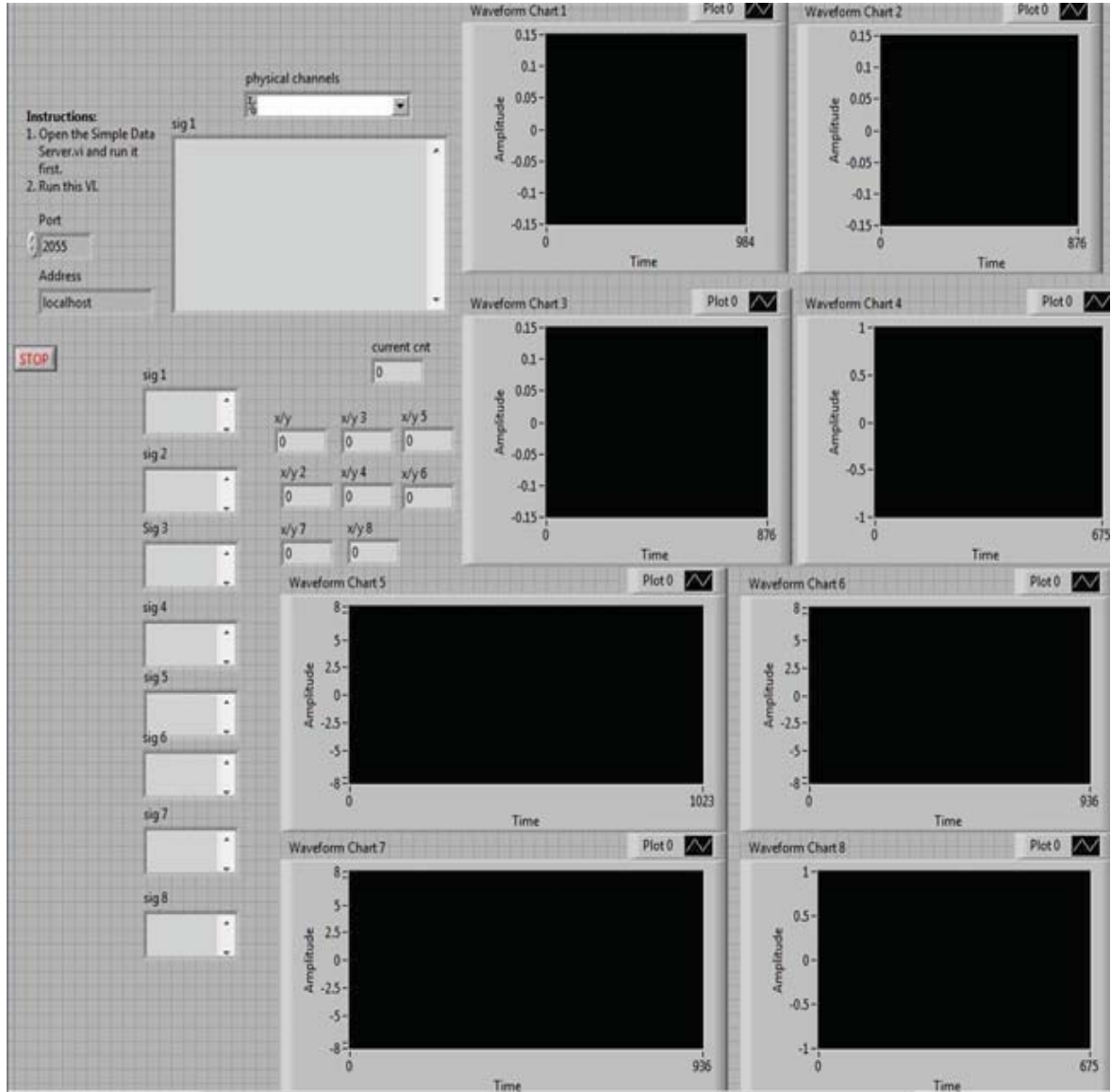


Figure 5.11

LabVIEW Client Model with data Parsing Algorithm

The same VI, when run on different compatible hardware gives different loop processing speeds. This VI does not utilize any third party libraries or applications such as WinPcap and hence it can be operated in real time. A major misapprehension about real time is that real time means quick. Real time operation mainly signifies that the calculations and processing will be surely completed within a given window of time.

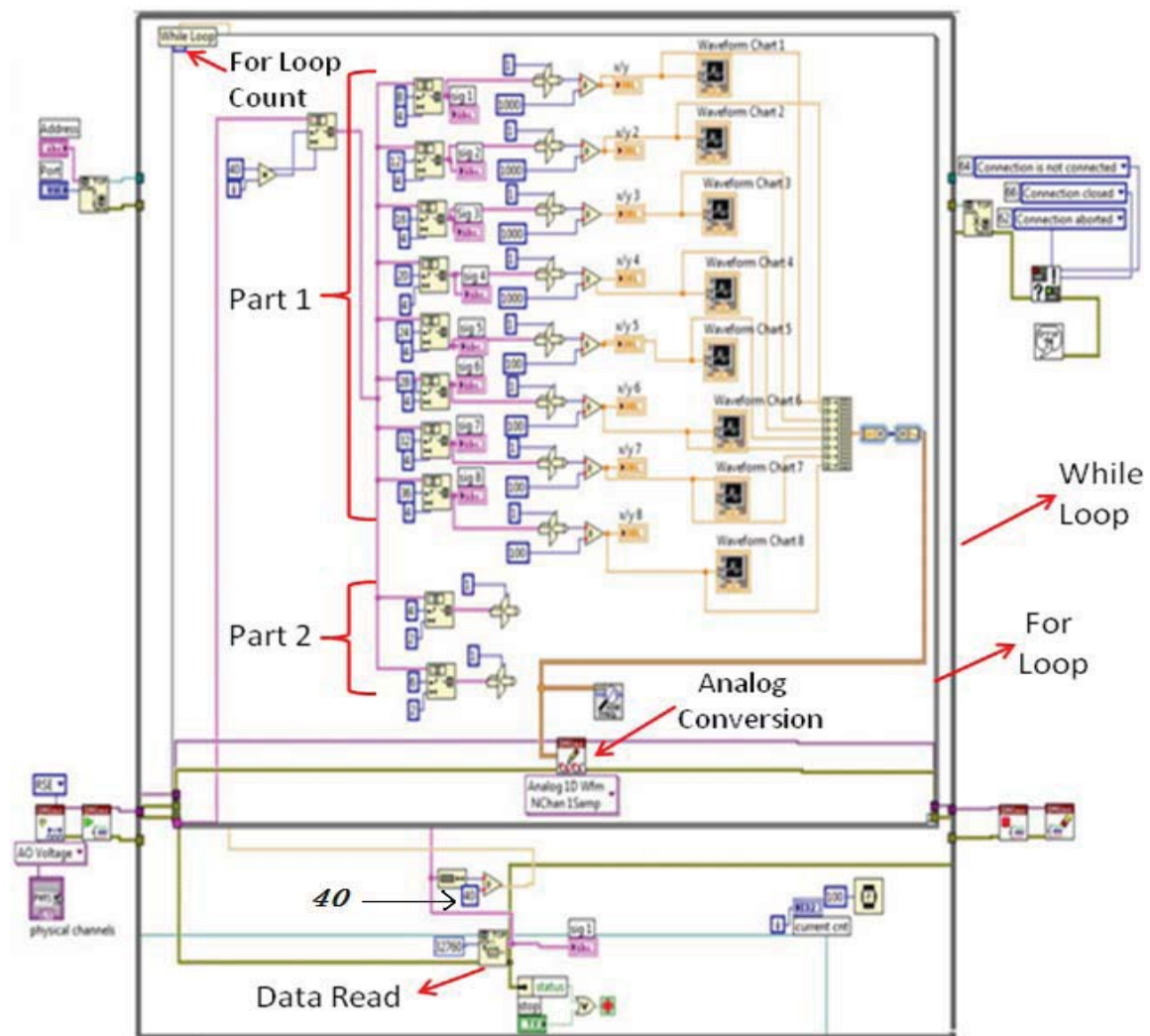


Figure 5.12

Block Diagram of LabVIEW Client Model

Initially, the designed Client LabVIEW VI is added to the new project created in the Real Time target. The supporting VIs for the Client file are also added for the proper operation in Real Time. Then, the project will be deployed on the Real Time target i.e. NI PXI through Ethernet and will be checked for any compatibility and missing link errors. After successfully deploying the VI, running the VI will make it run in Real Time. In Figure 5.12, Part 1 shows the parsing of eight signals from the TCP/IP data string. After parsing the required bytes out of the string, these are converted into double signals of 64 bits real. All these extracted double signals are input into an array and the built array is fed to “DAQmx Write” component. This component writes a waveform that contains a single sample to each channel in a task that contains one or more analog output channels [23]. Part 2 shows the four bytes that contain information about the data received such as sample count and the number of bytes that have arrived. The data read shown in the figure is nothing but the TCP read. In this component, one specifies the number of bytes to read from the TCP/IP string and it is 32760 in the present setup. This value can be randomly chosen around 30000 making sure that it is divisible by 40. Keeping it around 30000, will make the client file work properly for the available data transfer rates. After dividing the length of the string by 40, the output value is fed to the loop count of the FOR loop so that it executes that many times. The input 40 to the divide block in Figure 5.12 corresponds to the total of 36 bytes of data plus 4 bytes of data information.

Hence, the DAQ mx card is used with the LabVIEW Real Time Modeling of the Client VI to obtain the analog outputs from the SCB 68 board, which is connected with the NI PXI. These obtained signals will be sent into the RTDS for simulation using the

Optical Analog to Digital Converter (OADC) card. The development of the communication test bed was performed in collaboration with TAMU University.

## 5.5 Experiments and Data Rates

After the development of the test bed, different experiments were performed locally and remotely for verification of data transfer. The remote experiments were performed with Texas A&M University. For the GTNET data capture from the RTDS, two different platforms were tested, which included LabVIEW and C# programs. Using these two programs, different kinds of experiments were performed during different times of the day for data rate calculation and data transfer validation. These are shown in Figures 5.13 and 5.14.



Figure 5.13

Remote monitoring experiments using LabVIEW programming platform



Figure 5.14

Remote monitoring experiments using C# programming platform

Figure 5.13 shows the two different kinds of experiments that were performed. These experiments resulted in many conclusions and analysis. The results of these experiments have been tabulated and compared. The first kind of experiments involved LabVIEW server and LabVIEW client for data transfer between the two simulators. The second kind of experiments involved the usage of C# program designed by the RTDS. The RTDS only has the part of data capture in the LAN, but this program was extended to data capture and data transfer in the WAN.

### 5.5.1 Data transfer test using three-bus power system test case

The three bus power system was designed in RSCAD and the test case is shown in Figure 5.15.

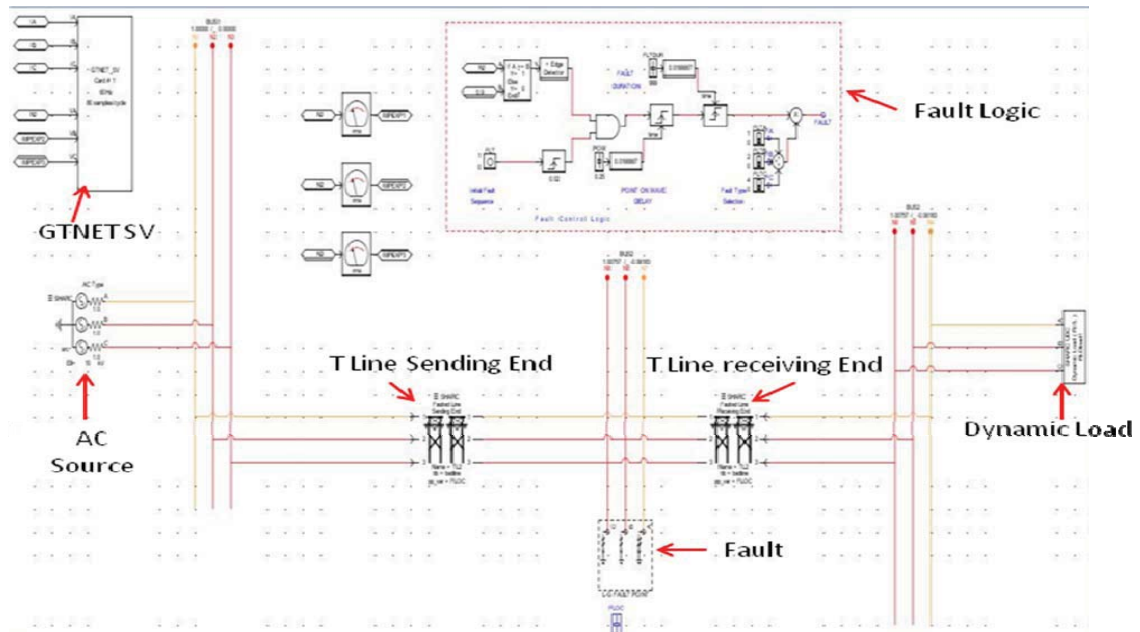


Figure 5.15

Three Bus power system test case in RTDS



The power system, shown above in Figure 5.15 was mainly used to achieve the objective of transferring the current and voltage signals from one side to the other side. In this test, the RMS bus voltages and instantaneous transmission line current values were transferred. The configuration of the GTNET SV in the RSCAD is shown in Figure 5.16.

| rtds_risc_ctl_GTNET_SV |   |                  |                 |                 |       |
|------------------------|---|------------------|-----------------|-----------------|-------|
| CONFIGURATION          |   | IEC 61850 CONFIG | CHANNEL ENABLES | CHANNEL SCALING |       |
| Name                   | Description                                 | Value            | Unit            | Min             | Max   |
| IEDName                | GTNET iedName                               | IED_A            |                 |                 |       |
| APPID                  | APPID (hex)                                 | 4000             |                 |                 |       |
| VLANPRI                | VLAN priority                               | 4                |                 | 0               | 7     |
| VLANID                 | VLAN ID (hex)                               | 000              |                 |                 |       |
| LDpre                  | LDName prefix                               | 000              |                 |                 |       |
| LDsuf                  | LDName suffix                               | 4                |                 | 0               | 99    |
| MACH                   | Output multicast address (6 MSB hex digits) | 010CCD           |                 |                 |       |
| MACL                   | Output Multicast address (6 LSB hex digits) | 040000           |                 |                 |       |
| INCRT                  | Include refresh time field in message       | FALSE            |                 |                 |       |
| INCSSF                 | Include sample sync field in message        | TRUE             |                 |                 |       |
| INCSR                  | Include sample rate field in message        | FALSE            |                 |                 |       |
| SMPRT                  | Sample rate (samples/cycle)                 | 80               |                 |                 |       |
| SYSFREQ                | Nominal system frequency (Hz)               | 60               |                 |                 |       |
| CONFREV                | ConfRev                                     | 4                |                 | 0               | 65535 |

Update Cancel Cancel All

Figure 5.16

#### General Configuration for GTNET block in RSCAD

Figure 5.16 shows the different options in the IEC 61850 configuration. Some of these are the VLAN priority number, VLAN ID, MAC address of the data receiving hardware (mainly protection relays), sample rate, refresh time field and nominal frequency. In the present application, the GTNET data is flushed into the LAN so that it can be collected by the data capture program. The tests were conducted with a sample rate of 80 samples/cycle at 60 Hz nominal frequency.



### 5.5.2 Remote Monitoring experiments using RTDS

The sampled values of power system voltage and current signals are transferred over TCP/IP and are monitored during no fault and fault conditions. The data was captured by the LabVIEW server from rack1 and was sent to the rack2 through TCP/IP. The data can be monitored in LabVIEW also. By monitoring in rack 2, one can increase the flexibility of sending back control signals for the fault correction. Here, the RMS voltage values and instantaneous current values have been transferred. Figure 5.16 shows the current and voltage signals in rack 1 during normal conditions. The magnitudes of the transferred values are kept low so that no significant errors are introduced in the data transfer during this data validation experiment. During normal conditions, one can send values of higher magnitudes utilizing the scaling option provided by GTNET SV in the RTDS. Care must be taken for the NI 6733 card, which is used to convert the digital data into analog data because signals whose magnitudes are greater than  $\pm 10$  volts will damage the PXI equipment. So, proper scaling has to be considered before the data transfer.

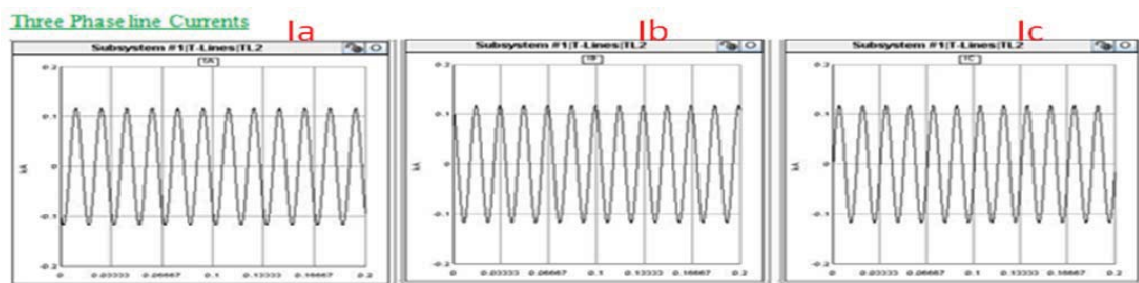


Figure 5.17

Instantaneous Line Currents in Rack 1 during steady state condition

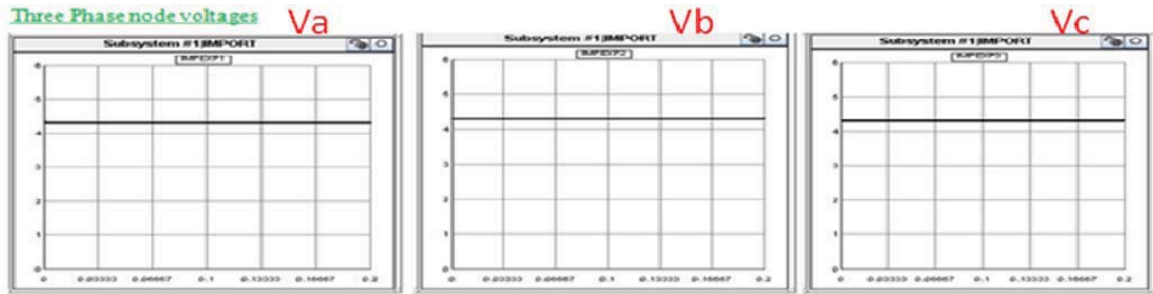


Figure 5.18

RMS Node voltages in Rack 1 during steady state condition

Figure 5.19 shows the line currents and node voltages in LabVIEW client running in Real Time during normal conditions. It has been captured by the PC, which is interfacing the NI PXI. It shows the three RMS and three instantaneous values. The amplitude of the fourth and the eight signals are zero, which is set to zero in the GTNET SV. These two signals correspond to the  $I_N$  and  $V_N$  signals of the GTNET SV. As mentioned before, these signals are not time referenced, so the x axis is just a time scale, which counts the number of data points for the transferred signals.



Figure 5.19

Line Currents and Node voltages in LabVIEW client during steady state condition

The current and voltage signals in rack 2 are shown in Figure 5.20. The magnitudes of these signals are scaled down using the GTNET SV.

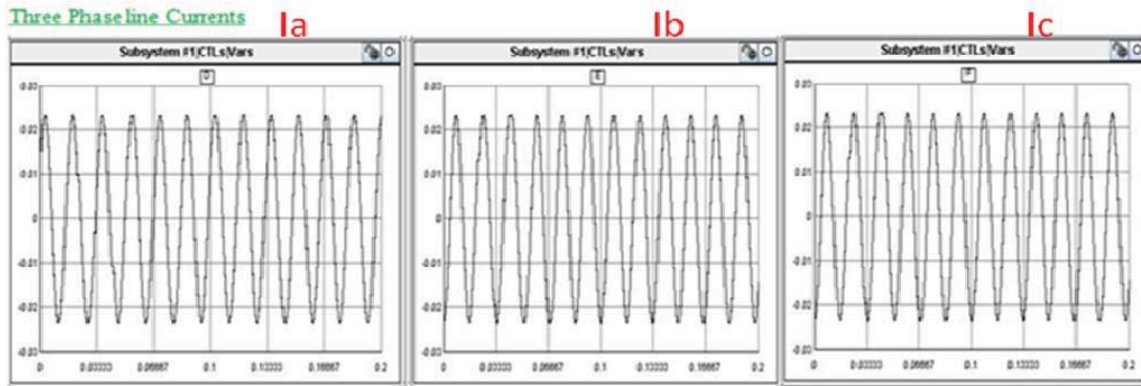


Figure 5.20

Instantaneous Line Currents in Rack 2 during steady state Condition

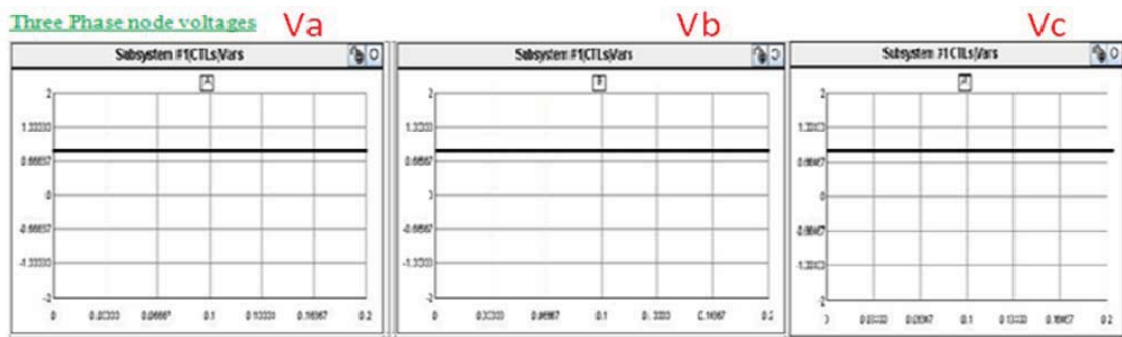


Figure 5.21

RMS Node voltages in Rack 2 during steady state Condition

Now, the same system is tested under a fault condition and the fault behavior is validated in both the racks. A bolted fault of 15 cycles at bus 3 was applied in the power system test case shown in Figure 5.15. The fault behavior is also captured in the LabVIEW client running on the NI PXI. The fault behavior in rack 1 is shown in Figure 5.22.

### Three Phase line Currents

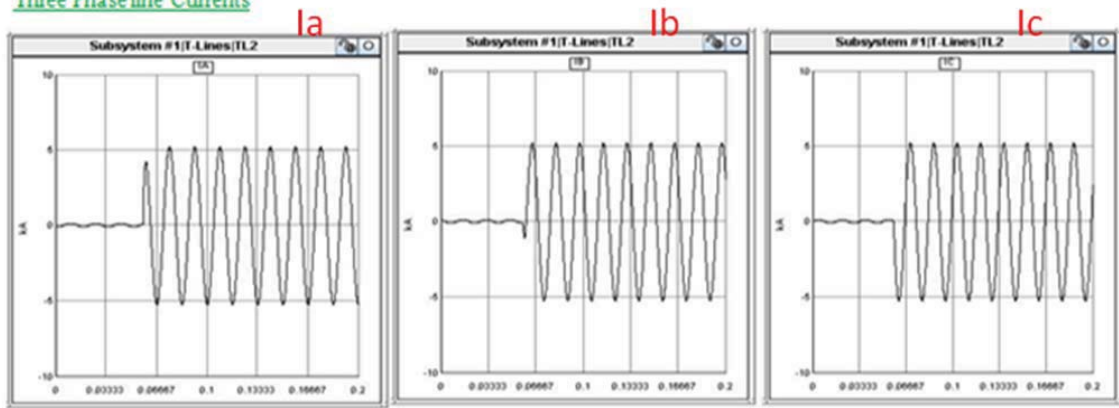


Figure 5.22

Instantaneous Line Currents in Rack 1 during Fault Condition

### Three Phase node voltages

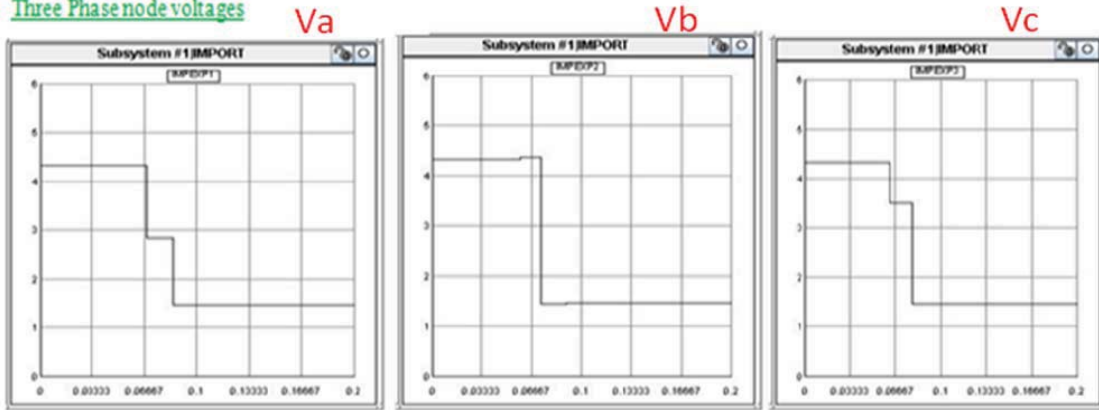


Figure 5.23

RMS node voltages in Rack 1 during Fault Condition

The same behavior is observed in LabVIEW client also, which is shown in the Figure 5.24. The rack 2 also exhibits the same kind of behavior validating the data transfer between both the racks. Hence, it can be concluded that the test bed between the simulators works well during normal as well as fault conditions. Thus, it can be used for

distributed simulation of power systems using RTDS. The rack 2 behavior is shown in Figure 5.21.

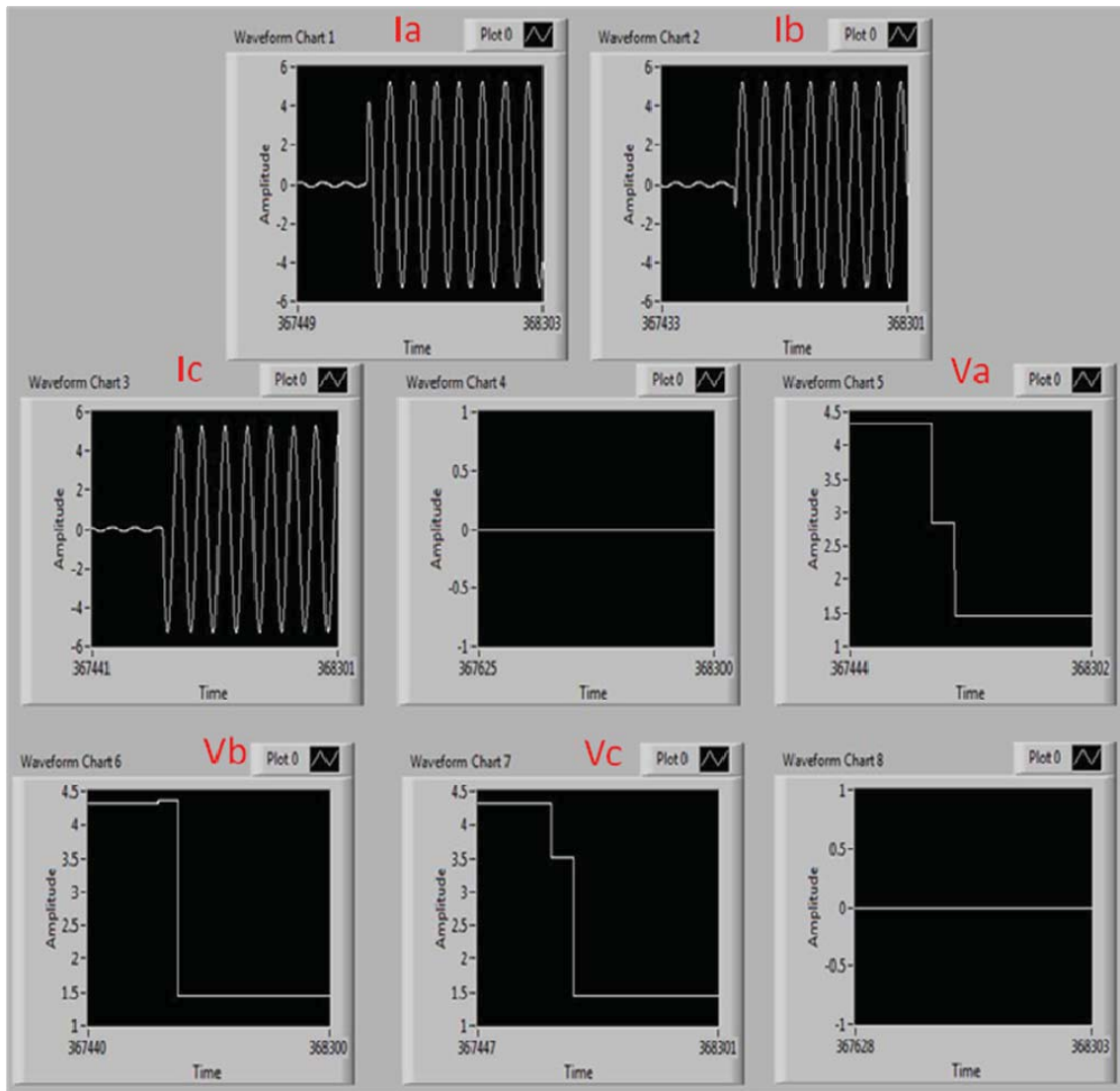


Figure 5.24

Line Currents and Node voltages in LabVIEW client during fault conditions

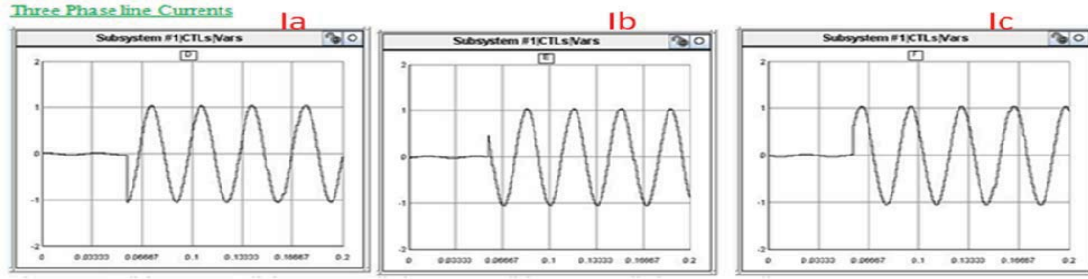


Figure 5.25

Instantaneous Line Currents in Rack 2 during Fault Condition

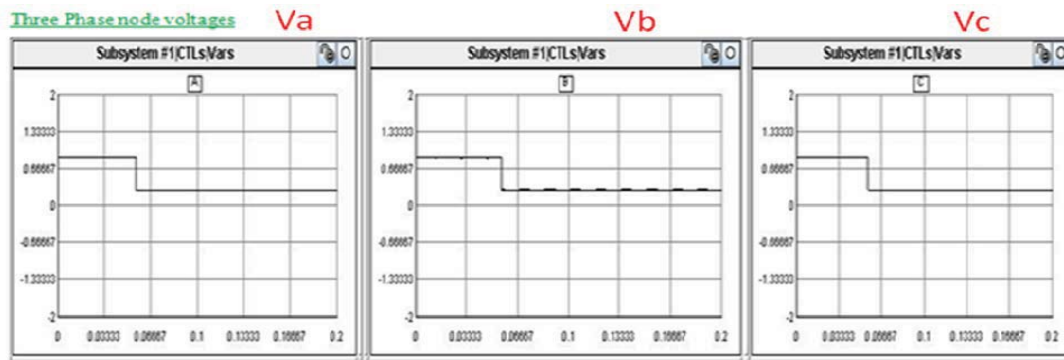


Figure 5.26

RMS Node voltages in Rack 2 during Fault Condition

### 5.5.3 Calculation of Data rates

The other aspect of the experiments between MSU and TAMU involves the calculation of data transfer rates for the power system data between the simulators. Initially, different experiments were performed where 118 bytes of data was transferred in each and every sample coming out of GTNET SV in the RTDS. Later experiments involved transfer of 36 bytes of data per sample to increase the data transfer speed. In the first kind of experiments, the GTNET data was captured by a PC in the LAN and it was

sent to a PC on the other side using TCP/IP. Here, all the 118 bytes of data coming out of GTNET are transferred to the other side. After receiving the data, these were just stored in an array list and no parsing was done. The calculated data rates are shown in Table 5.1. It contains information, such as data rate from Wireshark, calculated data rate, number of packets from C# and Wireshark, total time taken for data transfer including and excluding the time for packets that are required for connection establishment. Finally the average time taken and average output rate for 100,000 GTNET packets are calculated. The data rates, shown here are calculated using C# packet capture program developed by RTDS Inc.

Table 5.1

Results of Data transfer Experiment 1 using C#

| No.            | Data rate for the duration From Wireshark(Mb/s) | Data rate Calculated (Mb/s) | Number of Packets from C# | Total time taken for data transfer (sec) | Time taken – the final stop message (seconds) | Avg. time (ms) for 1 GTNET packets and Avg. rate (Actual value) |
|----------------|---|-----------------------------|---------------------------|--|---|---|
| 1              | 2.681   | 2.60                        | 16463                     | 38.26                                    | 36.44   | ~0.208, 4.53 Mb/s   |
| 2              | 2.538   | 2.48                        | 16371                     | 40.43                                    | 38.26   | ~0.208, 4.53 Mb/s   |
| 3              | 2.279   | 2.58                        | 16284                     | 44.97                                    | 36.72   | ~0.208, 4.53 Mb/s   |
| 4              | 2.452   | 2.61                        | 16472                     | 41.80                                    | 36.31   | ~0.208, 4.53 Mb/s   |
| 5              | 2.635   | 2.58                        | 16513                     | 38.90                                    | 36.71   | ~0.208, 4.53 Mb/s   |
| <b>Average</b> | <b>2.517</b>                                    | <b>2.57</b>                 | <b>16420.6</b>            | <b>40.87</b>                             | <b>36.88</b>                                  | <b>~0.208, 4.53 Mb/s</b>  |



The calculated data rates shown in Table 5.1 are obtained using the formula shown below.

$$\text{Data rate Calculated} = [(B - P*H)*8] / (\text{final stop message time (sec)}*1E6)$$

in Mb/s where

B = Total number of bytes from Wireshark summary

P= Total number of packets from Wireshark summary

H= Average Header length for each packet

B, P, H can be found from the file, which has the data captured during the experiment.

The second, fifth and sixth column of the table were obtained directly from Wireshark analysis after performing the experiment. The time taken (after the final stop message), shown in the seventh column is calculated from the recorded data in Wireshark. It shows the exact time taken for data transfer between the simulators excluding the data packets which are required for establishment (opening and closing) of a client server connection. In the C# program, which is used for data capture and data transfer, a stop message is included immediately after the transfer of last data packet to calculate the exact time taken for data transfer.

In the second set of experiments, the data on the receiving end PC was separated into 118 bytes and stored in an array list with each element being 118 bytes. The received data was finally written into a text file to validate the data transfer. The experiment 2 details are shown in Table 5.2.

Table 5.2

Results of Data transfer Experiment 2 using C#

| No.            | Data rate for the duration From Wireshark(Mb/s) | Data rate Calculated (Mb/s) | Number of Packets from C# | Total time taken for data transfer (sec) | Time taken – the final stop message (seconds) | Avg. time (ms) for 1 GTNET packets and Avg. rate (Actual value) |
|----------------|---|-----------------------------|---------------------------|--|---|---|
| 1              | 2.164   | 2.58                        | 16244                     | 47.44                                    | 36.73   | ~0.208, 4.53 Mb/s   |
| 2              | 2.186   | 2.56                        | 16324                     | 46.89                                    | 37.09   | ~0.208, 4.53 Mb/s   |
| 3              | 2.304   | 2.57                        | 16134                     | 44.44                                    | 36.80   | ~0.208, 4.53 Mb/s   |
| 4              | 2.322   | 2.61                        | 16235                     | 44.18                                    | 36.31   | ~0.208, 4.53 Mb/s   |
| 5              | 2.184   | 2.56                        | 16326                     | 46.97                                    | 37.03   | ~0.208, 4.53 Mb/s   |
| <b>Average</b> | <b>2.232</b>                                    | <b>2.576</b>                | <b>16252.6</b>            | <b>45.98</b>                             | <b>36.79</b>                                  | <b>~0.208, 4.53 Mb/s</b>  |

In the third set of experiments, the GTNET data has been processed to 36 bytes from 118 bytes before sending it to the other side. This processing takes place on the PC using LabVIEW that is used to capture the data coming out of GTNET. Each signal consists of 8 bytes, where 4 bytes are always set to zero to maintain resolution. So, the total signals contribute to 64 bytes and the rest 54 bytes contains different information pertaining to signals such as sample count, Ethernet header, protocol header and packet size etc. In these experiments, the extra four bytes for each signal as well as other unnecessary data were removed. So, the 36 bytes corresponds to 32 bytes of power system data and four bytes of information about the signals such as two bytes for sample

count and two bytes for packet size. The calculations for this kind of experiment showed good results with respect to the data rates. These are shown in Table 5.3.

Table 5.3

Results of Data transfer Experiment 3 using C#

| No.            | Data rate for the duration From Wireshark(Mb/s) | Data rate Calculated (Mb/s) | Number of Packets from C# | Total time taken for data transfer (sec) | Time taken – the final stop message (seconds) | Avg. time (ms) for 1 GTNET packets and Avg. rate (Actual value) |
|----------------|---|-----------------------------|---------------------------|--|---|---|
| 1              | 1.127   | 1.384                       | 4782                      | 27.63                                    | 20.97   | ~0.208,1.382 Mb/s   |
| 2              | 1.149   | 1.383                       | 4946                      | 27.10                                    | 20.93   | ~0.208,1.382 Mb/s   |
| 3              | 1.106   | 1.386                       | 4835                      | 28.05                                    | 20.82   | ~0.208,1.382 Mb/s   |
| 4              | 1.167   | 1.380                       | 4882                      | 26.67                                    | 20.96   | ~0.208,1.382 Mb/s   |
| 5              | 1.123   | 1.389                       | 4815                      | 27.64                                    | 20.81   | ~0.208,1.382 Mb/s   |
| <b>Average</b> | <b>1.134</b>                                    | <b>1.384</b>                | <b>4852</b>               | <b>27.41</b>                             | <b>20.89</b>                                  | <b>~0.208,1.382 Mb/s</b>  |

The data rate calculations using LabVIEW has shown more accuracy and compatibility. The LabVIEW server and client were used on either sides for sending and receiving the data, respectively. Sending 36 bytes seems to be a more productive option in terms of data rates. The actual data rate of GTNET for 118 bytes is 4.53Mbps and 36 bytes is 1.382Mbps. These experiments were performed with TAMU and all the data rates shown depend on the Internet traffic at that instance. When data loss occurs due to communication breakdown or any other unseen circumstances, different numerical

methods are adopted to deal with such kind of situation. Some of the methods explained in Chapter IV are adopted such as the extrapolation method etc.

Table 5.4

Results of Data transfer Experiment for 118 bytes (When MSU RTDS was Client)

| Iteration number | Data Rate (Mbps) | No. of data packets captured (Wireshark) |
|------------------|------------------|--|
| 1                | 2.021            | 66911                                    |
| 2                | 1.895            | 55171                                    |
| 3                | 1.875            | 60895                                    |
| 4                | 1.740            | 52421                                    |
| 5                | 1.938            | 59996                                    |
| Average          | 1.893            | 59078                                    |

Table 5.5

Results of Data transfer Experiment for 118 bytes (When MSU RTDS was Server)

| Iteration number | Data Rate (Mbps) | No. of data packets captured (Wireshark) |
|------------------|------------------|--|
| 1                | 2.871            | 63703                                    |
| 2                | 2.927            | 65952                                    |
| 3                | 3.024            | 68538                                    |
| 4                | 3.011            | 67804                                    |
| 5                | 3.029            | 70176                                    |
| Average          | 2.972            | 67234                                    |

Table 5.6

Results of Data transfer Experiment for 36 bytes (When MSU RTDS was Client)

| Iteration number | Data Rate (Mbps) | Execution time (min) | No. of data packets captured (Wireshark) |
|------------------|------------------|----------------------|--|
| 1                | 1.734            | 2                    | 52065                                    |
| 2                | 1.732            | 3                    | 58137                                    |
| 3                | 1.721            | 4                    | 62773                                    |
| 4                | 1.730            | 5                    | 68598                                    |
| 5                | 1.728            | 6                    | 73021                                    |
| Average          | 1.729            | 4                    | 62918                                    |

Table 5.7

Results of Data transfer Experiment for 36 bytes (When MSU RTDS was Server)

| Iteration number | Data Rate (Mbps) | No. of data packets captured (Wireshark) |
|------------------|------------------|--|
| 1                | 1.489            | 34011                                    |
| 2                | 1.411            | 34210                                    |
| 3                | 1.389            | 33304                                    |
| 4                | 1.376            | 31916                                    |
| 5                | 1.425            | 33453                                    |
| Average          | 1.418            | 33378                                    |

Tables 5.4 and 5.5 represent the data rate calculations with 118 bytes of data transfer with MSU as client and server, respectively. In the case of 118 bytes, calculated data rates were higher and closer to the actual value when MSU acted as server. Tables

5.6 and 5.7 represent the data rate calculations with 36 bytes of data transfer with MSU as client and server, respectively. In the case of 36 bytes, calculated data rates were closer to the actual value when MSU acted as a client.

## **5.6 Summary**

This chapter has discussed different schematics of getting the data in and out of RTDS using LabVIEW. These LabVIEW models are discussed in detail with respect to their modeling and operation. The test bed for data communication between the simulators has been developed in cooperation with TAMU. Finally, different kinds of experiments between MSU and TAMU have been presented. The results of data rate calculations have been presented using two different platforms of data capture.

## CHAPTER VI

### DEVELOPMENT OF WIDE AREA MONITORING AND CONTROL TESTBED WITH SIMULATION RESULTS

#### 6.1 Introduction

This chapter deals with the development of a test bed for wide area monitoring and control using RTDS and SVP. An angle difference scheme has been implemented and modeled to demonstrate the applicability of RTDS and SVP in Wide Area Monitoring & Control Systems (WACMCS).

#### 6.2 Wide Area Monitoring and Control (WAMCS)

The core idea of the WAMC systems is the centralized processing of the data collected from various locations of a power system, aiming at the evaluation of the actual power system operating conditions with respect to its stability limits [2]. Nowadays PMUs are being increasingly deployed by power utilities throughout the world. When optimally placed in the electricity network, these can provide more comprehensive information about the system. The PMUs turn the state estimation measurements into real time measurements and can enhance the state estimation efficiency. Wide area control

addresses automatic healing capabilities to some extent by proposing decisive smart topology changes and control actions with the goal of maintaining the integrity of the grid under adverse conditions [26].

### **6.3 Transient Stability Enhancement**

Transient stability has been mainly considered under angular stability. Transient stability refers to the ability of the system to maintain synchronism after being subjected to a major fault, such as loss of generation, line switching operations, faults and sudden load changes [34]. The main motive of studying transient stability is to analyze that for a given disturbance, whether the system is either transiently stable (or) unstable. One of the major challenges in the operation of a large interconnected power system is to ensure that the generators will remain in synchronism following a large system disturbance, such as loss of a large generator (or) a transmission line [33].

### **6.4 Test System Modeling and Control Algorithm**

The test case that has been used in this work is shown in the Figure 6.1. The power system has been modeled in RTDS with the machine and system control dynamics. Power System Stabilizer and Type ST1 excitation system are considered along with the generators. The typical exciter gain value for all the generators is 200. Generators G1, G2 and G3 are rated for 500MW, 700MW and 600MW, respectively. L1 and L2 are RL type dynamic loads rated 800MW and 1000MW, respectively. The RL load type is referred as dynamic load because the P and Q values can be dynamically adjusted during a simulation.



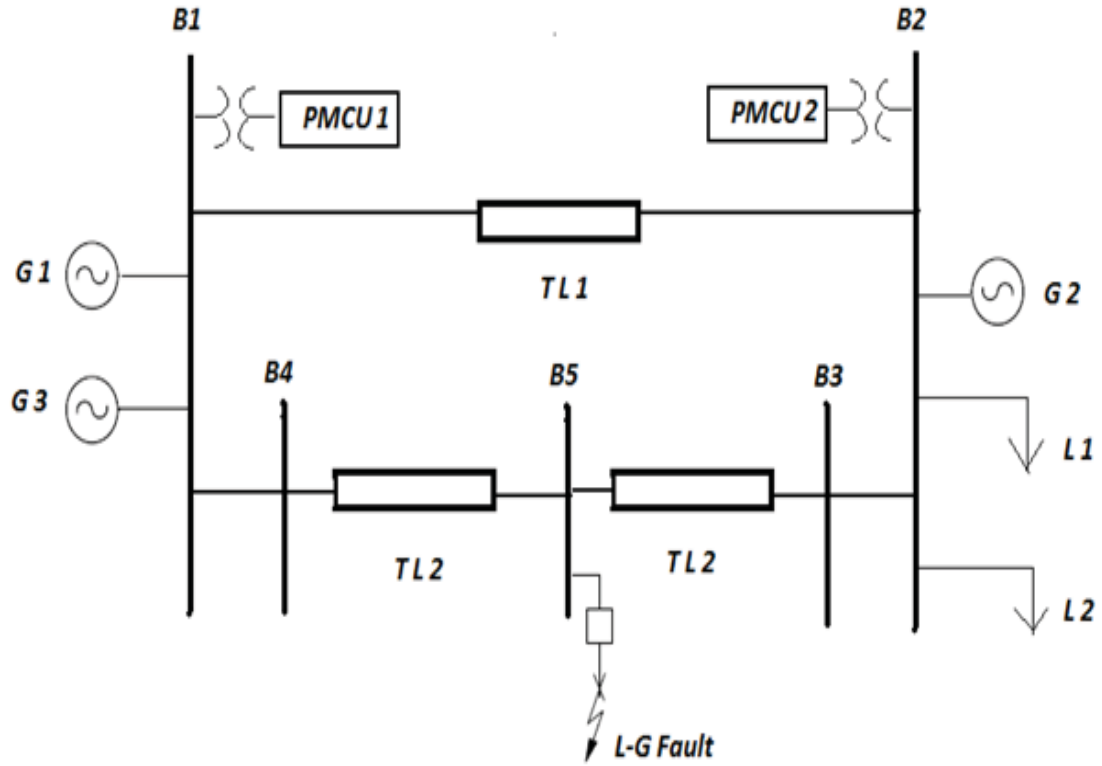


Figure 6.1

Test Case used for Wide Area Protection Scheme

In this test case, three generators and two dynamic loads have been used. Two Phasor Measurement and Control Units (PMCU) have been placed at Bus 1 and Bus 2. The angle difference between these buses is measured for steady state condition and also for fault condition. The modeling for the above test case in RSCAD is shown in Figure 6.2.

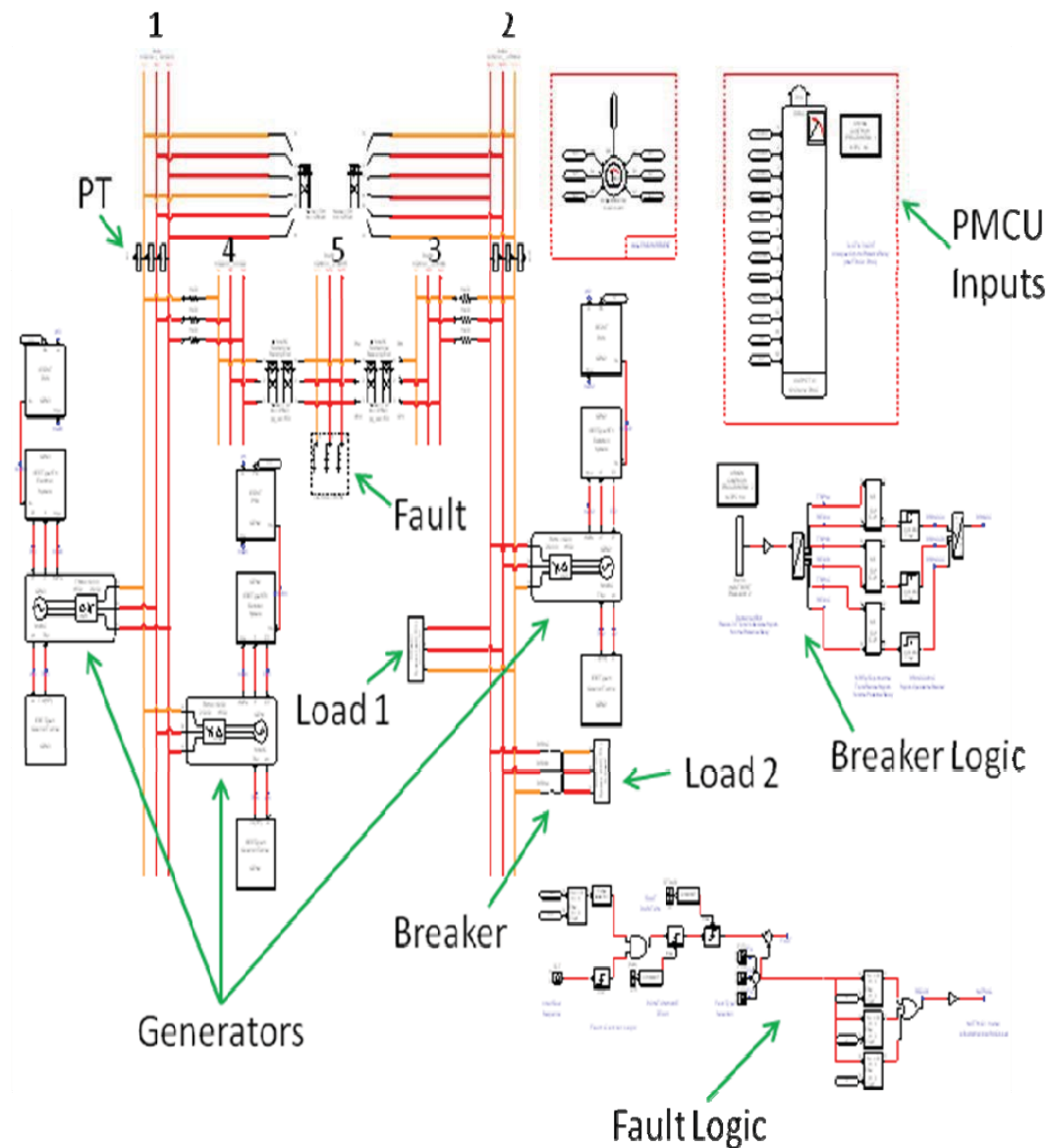


Figure 6.2

### Test Case Modeling in RSCAD

Basically, the PMCUs and RTDS act in a closed loop fashion like hardware in the loop. The PMCU measurements from RTDS are sent into the synchrophasor vector processor as input data into the wide area control algorithms. An angle difference scheme

has been built in the synchrophasor vector processor, as shown in Figure 6.3. This scheme has been programmed in a fashion to receive bus voltage angles and calculate the angle difference.

The wide area control algorithm has been designed and implemented in the synchrophasor vector processor. The SVP receives the phasor inputs from the PMUs and computes the difference between positive sequence voltage angles at Bus 1 and Bus 2. SVP asserts the corresponding alarm based on thresholds set at 150 degrees (0ms pickup timer) for Level 1 and 300 degrees (5ms pickup timer) for Level 2. If the angle difference exceeds the threshold, the alarm is asserted and it activates the Fast Operate (FO) function block. When the FO function block is activated, it sends the corresponding fast operate command (FOS command used for sending control signals) to the external control device (in this case, it is the PMCU 1) with the assigned IDCODE, when the Enable (EN) input is true. The angle difference schematic is shown in Figure 6.3.

In Figure 6.3, the PADM is the Phase Angle Difference Monitoring Block which has inputs such as angle 1 and angle 2 in radians, two levels of thresholds and pickup timers along with Second Of Century (SOC) and Fraction Of Second (FOS) of the incoming data. The outputs of these PADM blocks are angle difference, two alarms, SOC and FOC of the output along with an indicator (OK) for valid PADM output. The Rising edge Trigger (R\_TRIG) is used to enable the Fast Operate (FO) block corresponding to the Alarm. The Falling edge Trigger (F\_TRIG) is used to clear the control command after the FO block sends it to the PMCU.

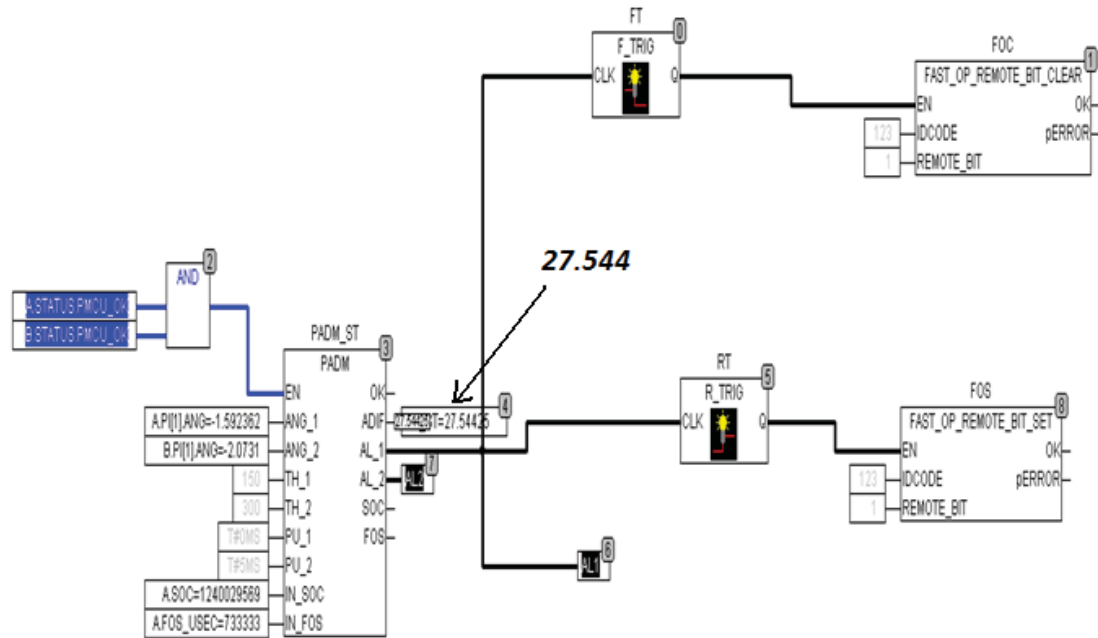


Figure 6.3

### Function Block Diagram Programming in SVP

#### 6.5 Simulation Results for Wide Area Control Scheme

Initially, the phasor voltages, currents and angles were sent to the SVP through PMCUs during the stable condition of the power system (running in RTDS). The angle difference between Bus 1 and Bus 2 was calculated in SVP using the PMU measurements and it was validated with the angle difference calculation in the RTDS. Figure 6.3 shows the angle difference as 27.544 degrees in the SVP which is also the same in RTDS as shown in Figure 6.4.

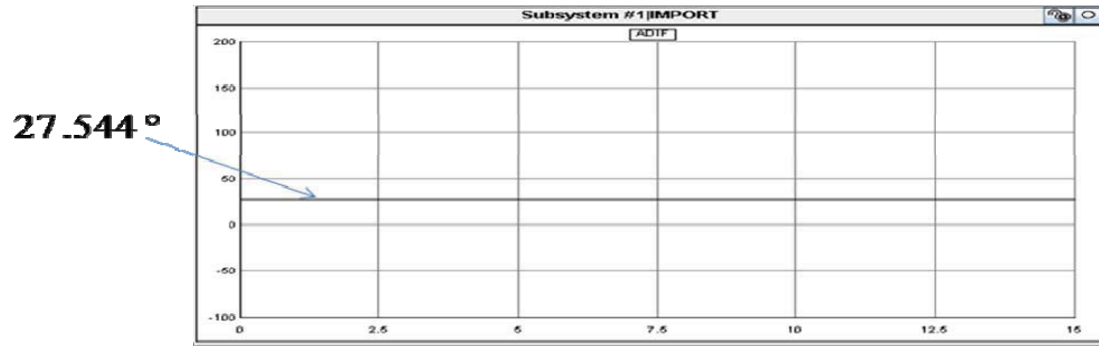


Figure 6.4

Angle Difference between Bus 1 and Bus 2 terminal node voltages in RTDS during steady state

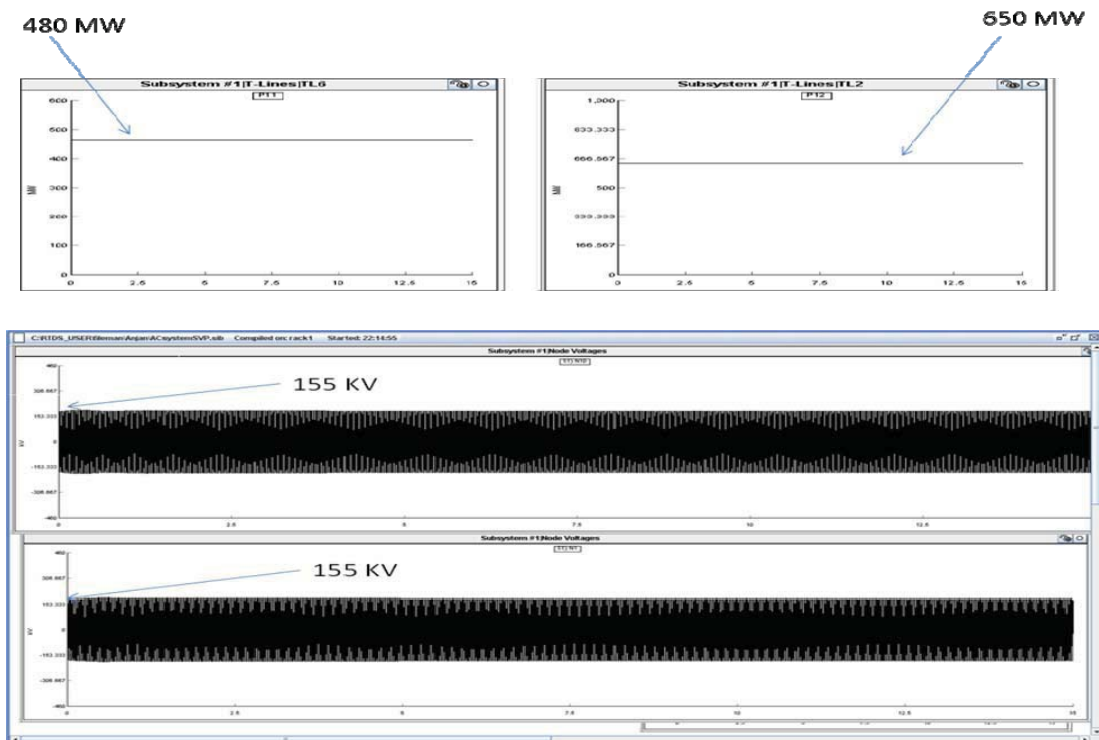


Figure 6.5

Transmission Line 1 and Transmission Line 2 Real Power between Bus 1 and Bus 2, three phase Node voltages at Bus 1 and Bus 2

The real power flow in transmission lines between Bus 1 and Bus 2 and node voltages of the Bus 1 and Bus 2 are shown in Figure 6.5. The system in RTDS has been simulated for 15 seconds with a time step of 100 microseconds. A fault of 28 cycles (467 ms) was applied on Bus 5 and undamped growing oscillations were observed in the system. The system's critical clearing time was observed as 450 ms cycles. In this case, the generators become unstable. The rotor angle velocity increases beyond limits and generators lose synchronism. The angle difference between Bus 1 and Bus 2, transmission line real power flow and the bus terminal voltages are shown in Figures 6.6, 6.7 and 6.8, respectively.

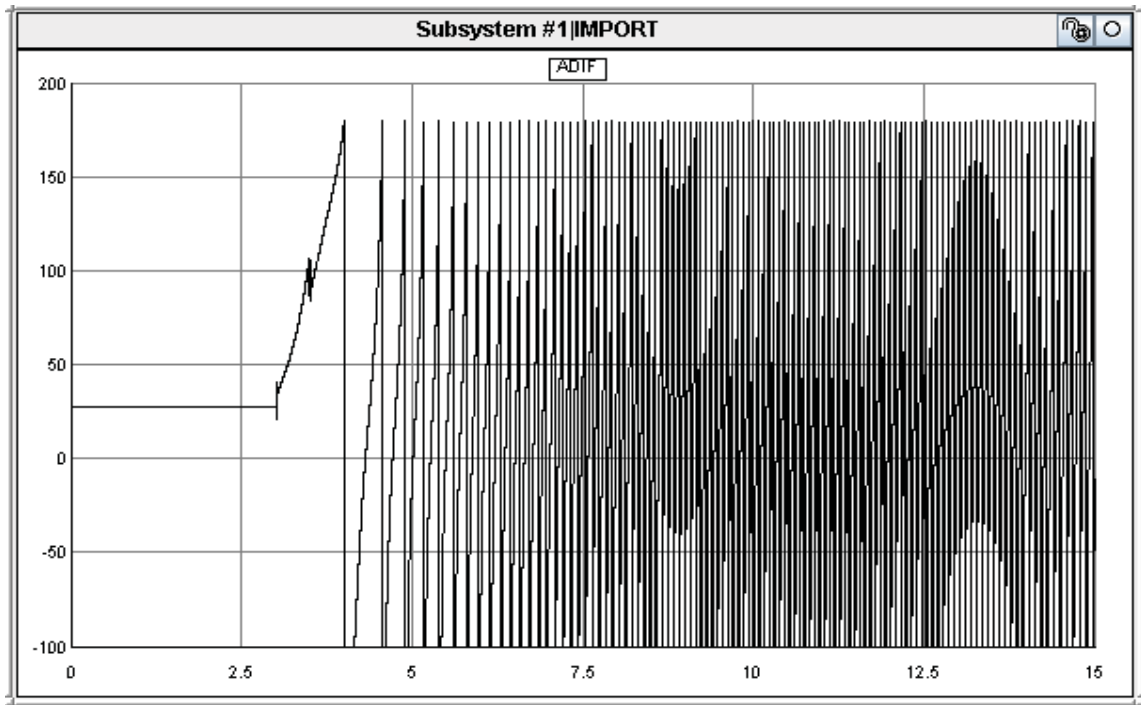


Figure 6.6

Angle Difference between Bus 1 and Bus 2 during Fault condition

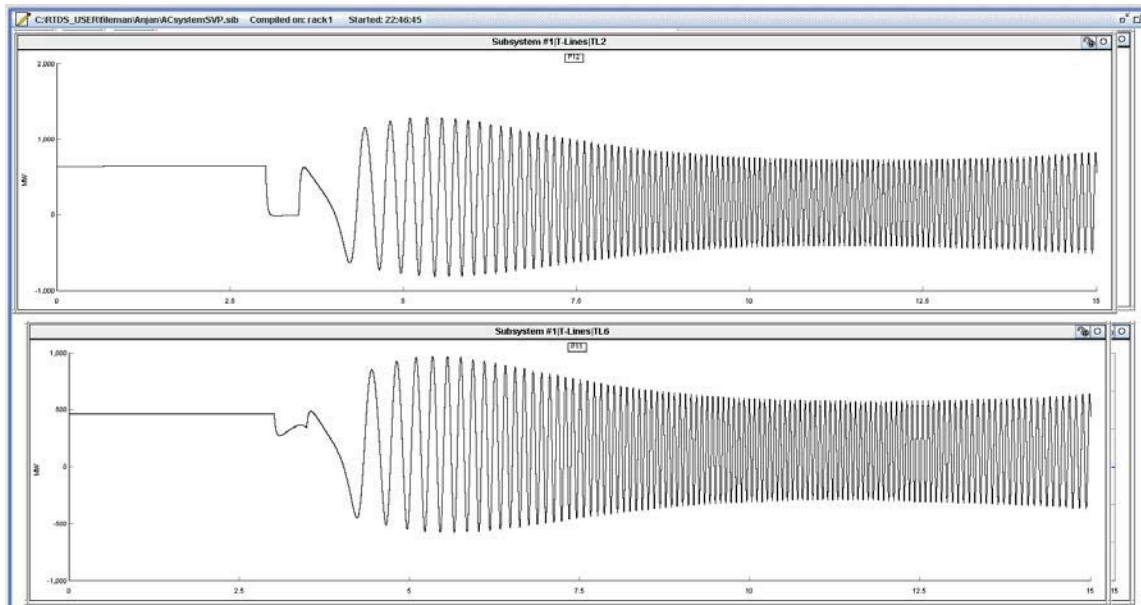


Figure 6.7

Transmission Line 1 and Transmission Line 2 Real Power between Bus 1 and Bus 2 during Fault condition

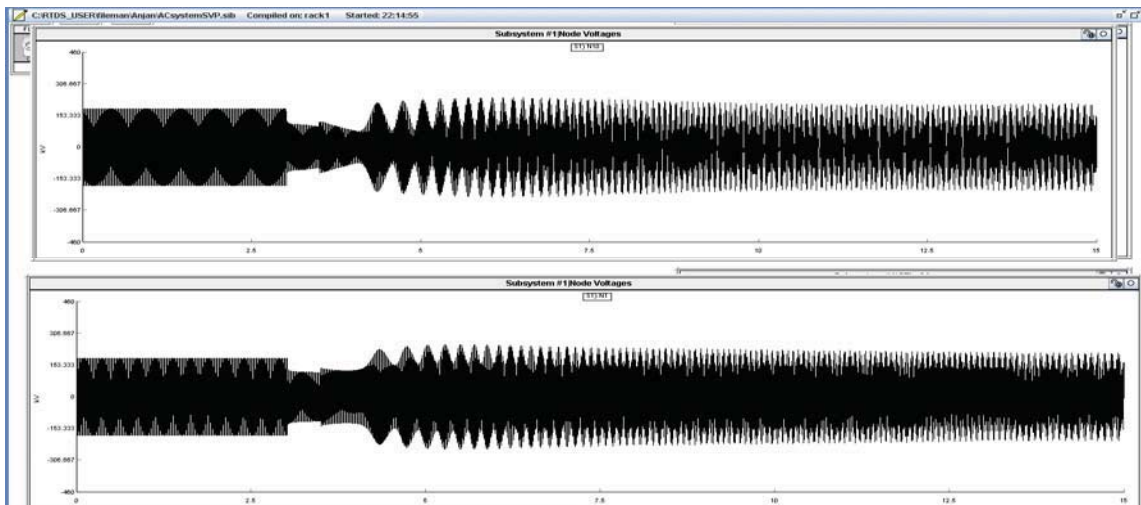


Figure 6.8

Three phase Node voltages at Bus 1 and Bus 2 during Fault condition

The synchrophasor vector processor action is shown in Figure 6.9, where the SVP calculates the angle difference and initiates the alarm, which corresponds to the activation of a control signal for the system protection by shedding the Load 2. The control signal is transferred from the SVP to the PMCU and the PMCU opens up a breaker, which allows load shedding.

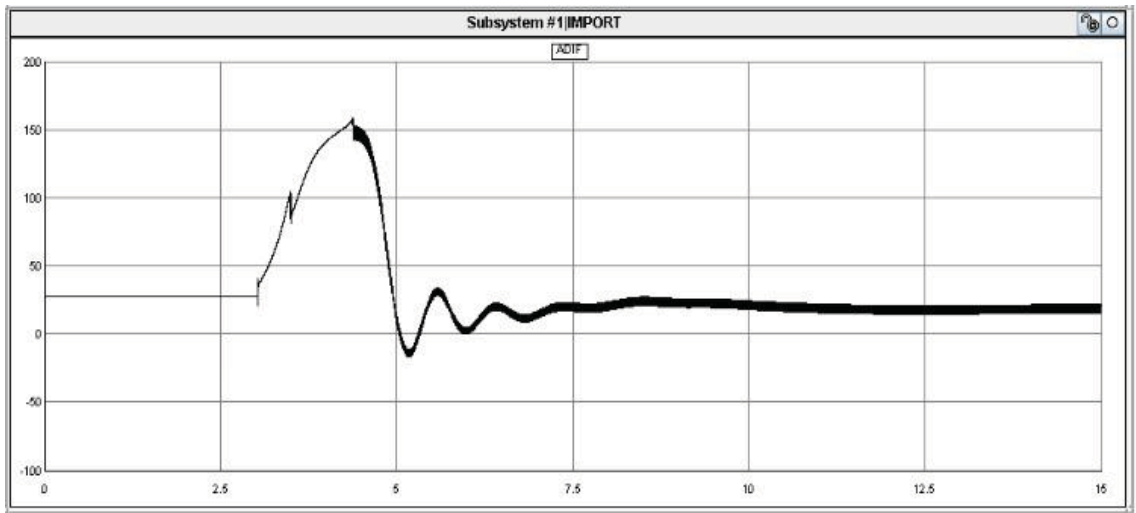


Figure 6.9

Angle Difference between Bus 1 and Bus 2 after SVP action

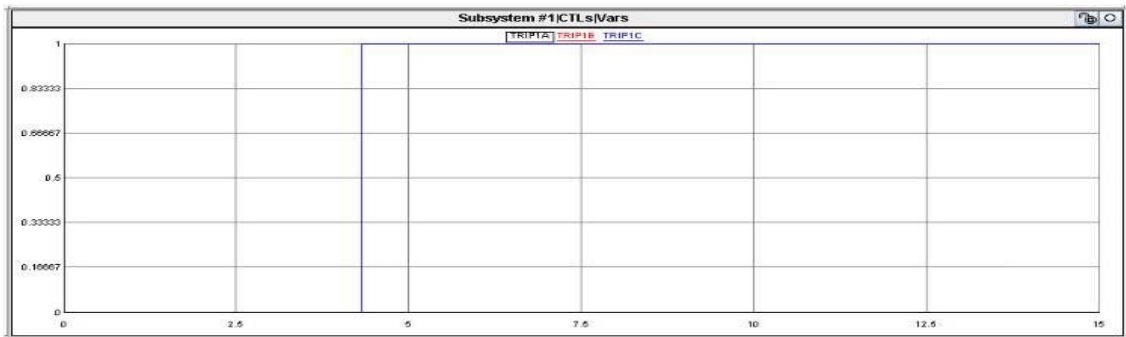


Figure 6.10

Control (trip) signal for load shedding after SVP action



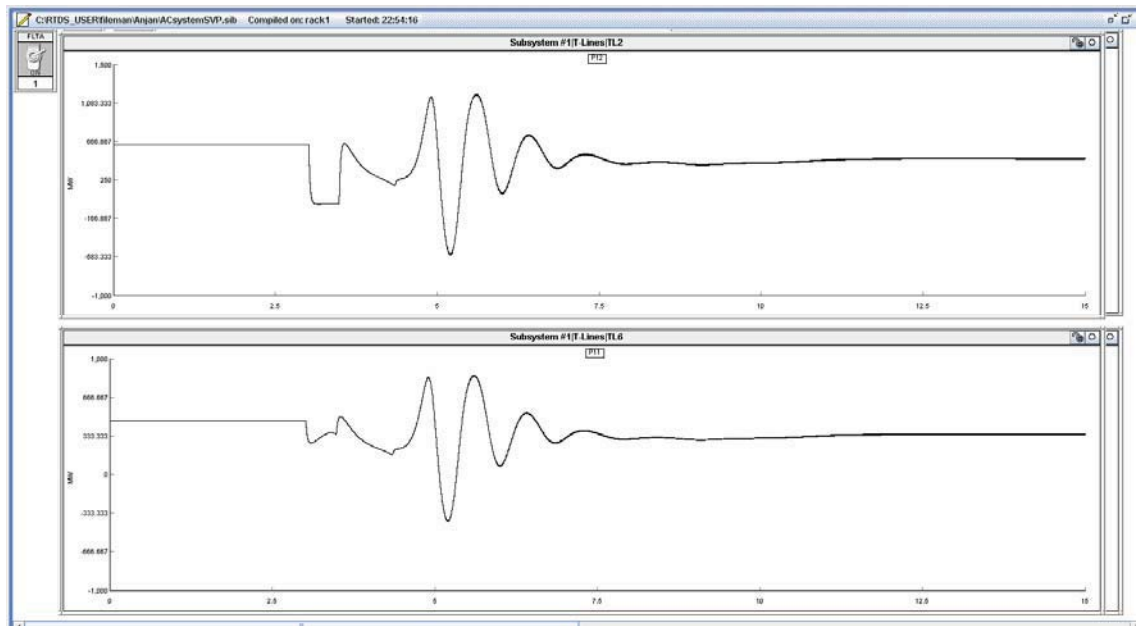


Figure 6.11

Transmission Line 1 and Transmission Line 2 Real Power between Bus 1 and Bus 2 after SVP action

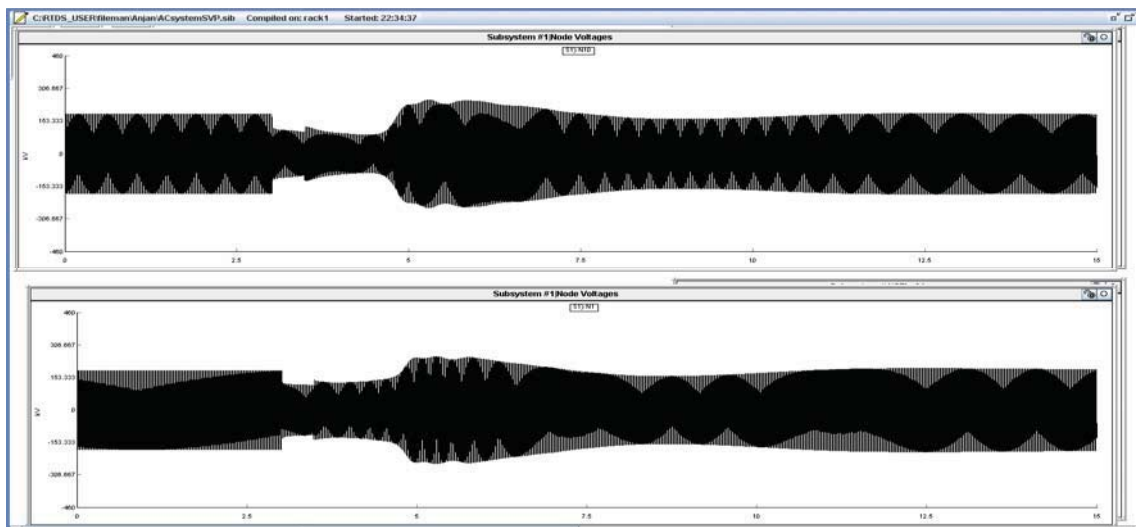


Figure 6.12

Three phase Node voltages at Bus 1 and Bus 2 after SVP action

A similar type of fault was applied at Bus 4 for validating the operation of the SVP. The SVP action is shown below. Here the angle difference between the two buses doesn't settle down immediately, and rather it takes some time, after the action of the SVP to settle down. This behavior is shown in Figure 6.13.

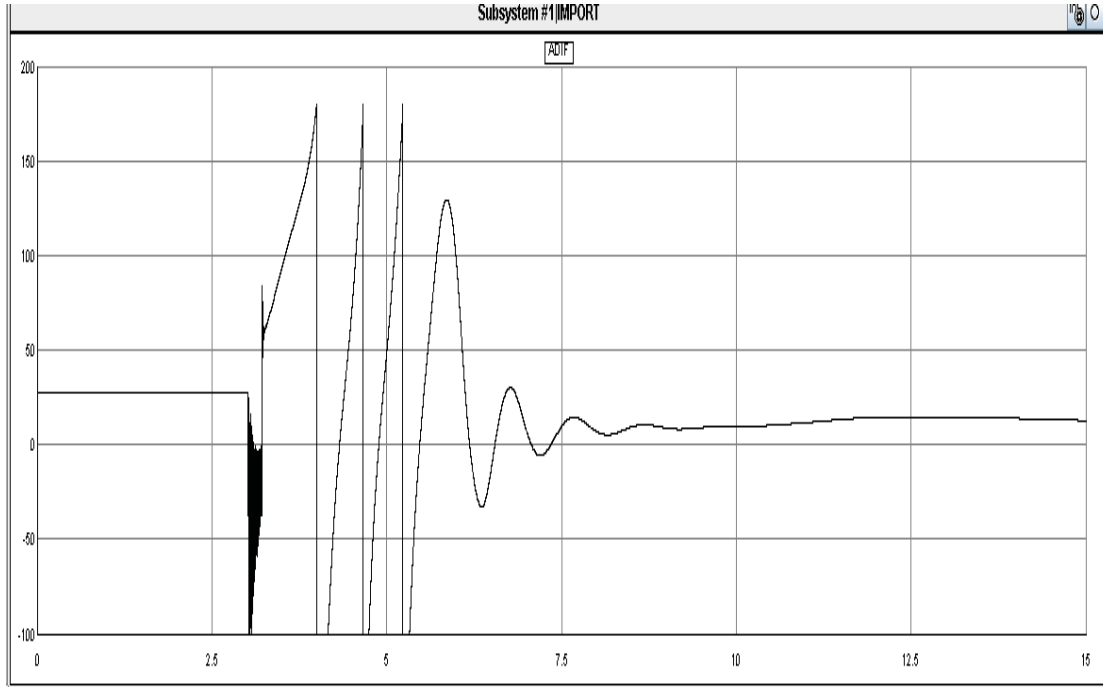


Figure 6.13

Angle Difference between Bus 1 and Bus 2 terminal node voltages in RTDS after SVP action for fault on Bus 4

## 6.6 Analysis of Distributed Simulation Locally with Simulation Results

Decoupled and non decoupled simulations were performed using the test case shown in Figure 6.2. The simulation time and resource usage for both the cases are compared and analyzed. When the test case, shown in Figure 6.2, is simulated on one rack, the minimum simulation time step required is 93.875 micro-seconds. The time step

information and the network solution for the Subsystem 1 (non decoupled system) are shown in Figure 6.14.

```

TIME - STEP INFORMATION
SUBSYSTEM 1
=====

Longest Computation Interval Start --> T0: 3PC 4A (9.8000 us)
Longest Computation Interval T0 --> T2: 3PC 2A (15.8000 us)
Longest Computation Interval Start --> T2: 3PC 1C (84.8750 us)

Backplane Communication Speed = 125.000000 ns
T0 communication time 5.3750 us
T2 communication time 7.0000 us

Minimum time-step 93.8750 us

```

Figure 6.14

Time Step Information for Subsystem 1 (non decoupled system)

```

RISC-based Network Solution for Subsystem #1 uses -->
RPC-GPC Card #1 Processor A
Network Solution Statistics for 1A:
Number of nodes on this processor: 18
Number of G values passed to this proc: 6
Number of matrix pointers passed to this proc: 0
Number of columns dynamically decomposed by this proc: 9
Number of multiply-subtract operations
for dynamic decomposition on this proc: 11
Number of clocks pre-T0: 874
( time = 8.740000e-001 uSec. )
Number of clocks in main: 2330
( decomposition: 761 )
( forward-backward: 1041 )
( time = 2.330000e+000 uSec. )
The fill of the decomposed lower matrix is: 48 of 171 locations.

```

Figure 6.15

Network Solution for Subsystem 1 (non decoupled system)

The test case, shown in Figure 6.2, has been decoupled into two subsystems and these are shown in Figures 6.16 and 6.17.

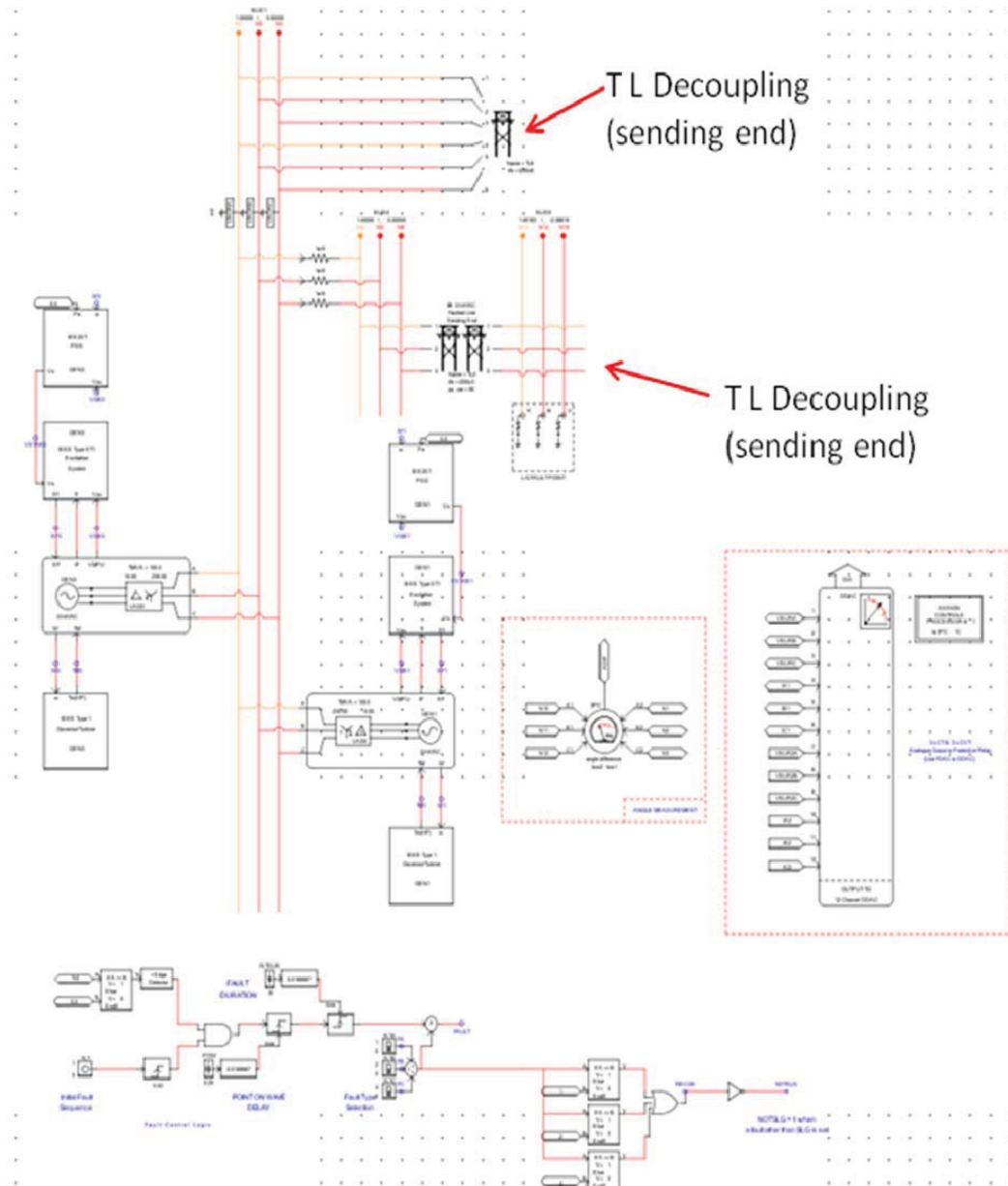


Figure 6.16

Subsystem 1 Modeling in RSCAD

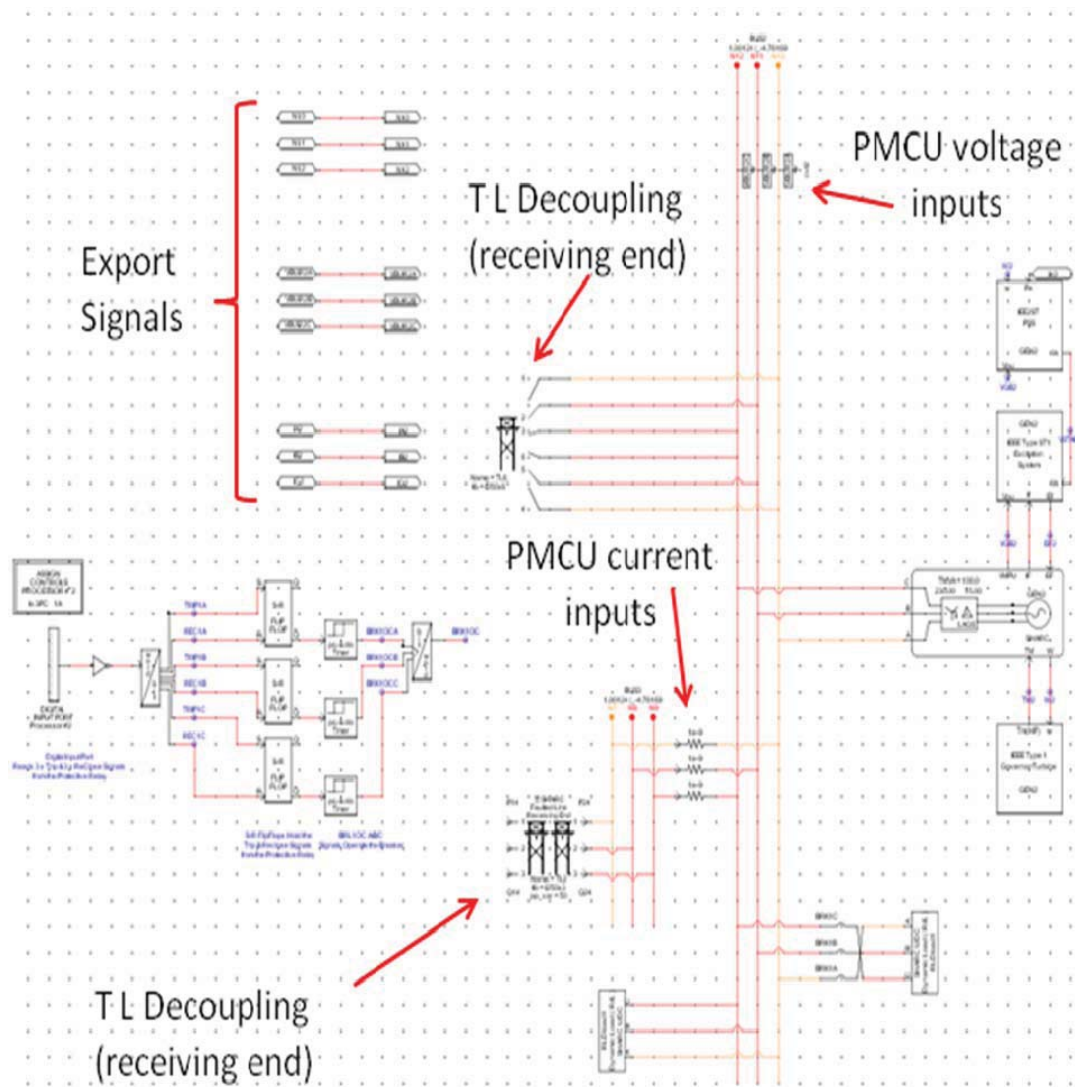


Figure 6.17

### Subsystem 2 Modeling in RSCAD

Here, the power system has been decoupled using the two transmission lines that are connecting Bus 1 and Bus 2. In Subsystem 2, the different signals, that are exported mainly comprise of Bus 2 node voltages, PMCU measured voltages and currents. Figure 6.17 shows the exported signals, two transmission lines that are decoupled and the

PMCU inputs. The time step information and the network solution details for the Subsystem 1 are shown in Figure 6.18.

```

T I M E - S T E P   I N F O R M A T I O N
S U B S Y S T E M   1
=====

note: Stats. listed here are for the current subsystem only.
      See the end of this file for global stats.
Longest Computation Interval Start --> T0:  3PC 1A  (8.2250 us)
Longest Computation Interval      T0 --> T2:  3PC 1A  (15.7250 us)
Longest Computation Interval Start --> T2:  3PC 1C  (60.6500 us)

Backplane Communication Speed = 125.000000 ns
T0 communication time 2.7500 us
T2 communication time 4.8750 us

Minimum time-step 67.5250 us

```

Figure 6.18

Time Step Information for Subsystem 1 (decoupled system)

```

RISC-based Network Solution for Subsystem #1 uses -->
      RPC-GPC Card #1 Processor A
Network Solution Statistics for 1A:
Number of nodes on this processor: 9
Number of G values passed to this proc: 0
Number of matrix pointers passed to this proc: 0
Number of columns dynamically decomposed by this proc: 3
Number of multiply-subtract operations
      for dynamic decomposition on this proc: 4
Number of clocks pre-T0: 499
      ( time = 4.990000e-001 uSec. )
Number of clocks in main: 1113
      ( decomposition: 312 )
      ( forward-backward: 534 )
      ( time = 1.113000e+000 uSec. )
The fill of the decomposed lower matrix is: 24 of 45 locations.

```

Figure 6.19

Network Solution for Subsystem 1 (decoupled system)



In the Subsystem 1, the minimum time step is 67.525 micro-seconds, as shown above. This means that the whole system, in Subsystem 1 can be solved within the specified minimum time step. Basically, in the RTDS, time steps are divided into communication intervals and computation intervals referred to as T0 to T4 [18].

```

TIME - STEP INFORMATION
SUBSYSTEM 2
=====
note: Stats. listed here are for the current subsystem only.
      See the end of this file for global stats.
Longest Computation Interval Start --> T0: 3PC 3B (9.8000 us)
Longest Computation Interval T0 --> T2: 3PC 1B (15.8000 us)
Longest Computation Interval Start --> T2: 3PC 4B (25.3750 us)

Backplane Communication Speed = 125.000000 ns
T0 communication time 3.5000 us
T2 communication time 4.1250 us

Minimum time-step 35.2250 us

```

Figure 6.20

Time Step Information for Subsystem 2

```

RISC-based Network Solution for Subsystem #2 uses -->
RPC-GPC Card #1 Processor A
Network Solution Statistics for 1A:
Number of nodes on this processor: 12
Number of G values passed to this proc: 6
Number of matrix pointers passed to this proc: 0
Number of columns dynamically decomposed by this proc: 6
Number of multiply-subtract operations
for dynamic decomposition on this proc: 7
Number of clocks pre-T0: 520
( time = 5.200000e-001 uSec. )
Number of clocks in main: 1576
( decomposition: 546 )
( forward-backward: 681 )
( time = 1.576000e+000 uSec. )
The fill of the decomposed lower matrix is: 30 of 78 locations.

```

Figure 6.21

Network Solution for Subsystem 2

Here the Subsystem 2 time step information and network solution are provided. The minimum time step is 35.225 micro-seconds. Thus, the minimum time taken to solve the complete system is 67.525 micro-seconds, as shown in Figure 6.22. The network solution is obtained in a decoupled fashion by simulating the power system on two racks. Decoupling the system has reduced the total simulation time step from 93.875 micro-seconds to 67.525 micro-seconds.

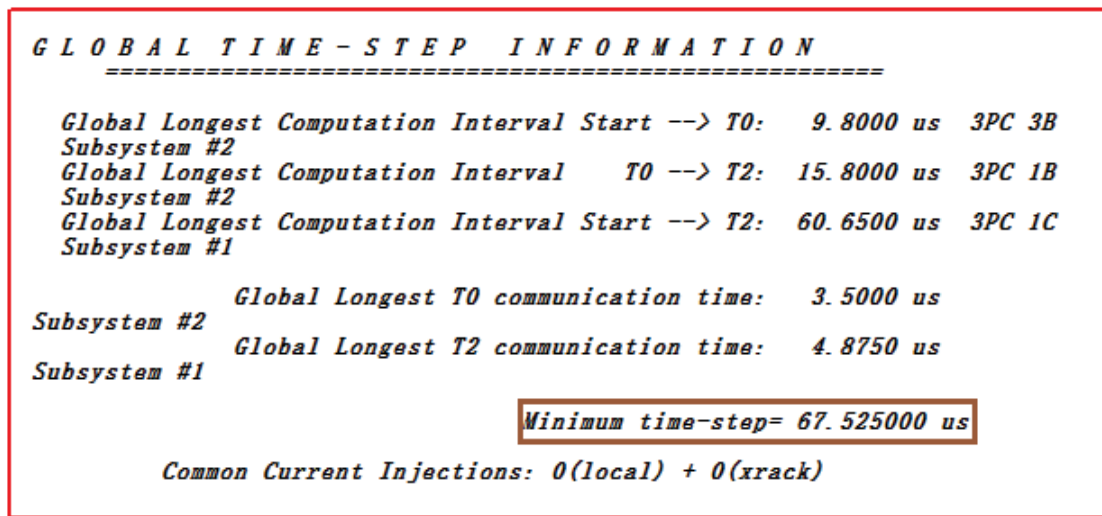


Figure 6.22

Global Time Step Information for the complete system

While partitioning subsystems in RTDS, the only requirement is that the travel time of all transmission lines linking the created sub-systems must be at least one time-step. In other words, any transmission line connecting one subsystem to another must have a certain minimum length. For instance, with a 50us time-step, the minimum length for an inter-rack T-line is about 15 km. If this requirement is not met, the simulation



results will be different than those obtained with the simulation on a single rack. Signals received from other racks are delayed by one time-step. Hence, sometimes it is wise to delay locally generated control signals, so that they line up with control signals imported from other racks.

Calculations for power system components and control components are necessarily computed on different processors. There is a one time step delay any time, when signals are exchanged between components calculated on different processors. This is true regardless of whether those processors are on the same rack (sub-system) or not. Figure 6.24 represents a graphical picture of how both the power system signal and a control system signal are passed from one processor to another. Figure 6.23 shows the different communication and computation intervals between the processors in different subsystems for a single time step. Time steps are divided into communication intervals and computation intervals, referred to as T0 to T4 [18].

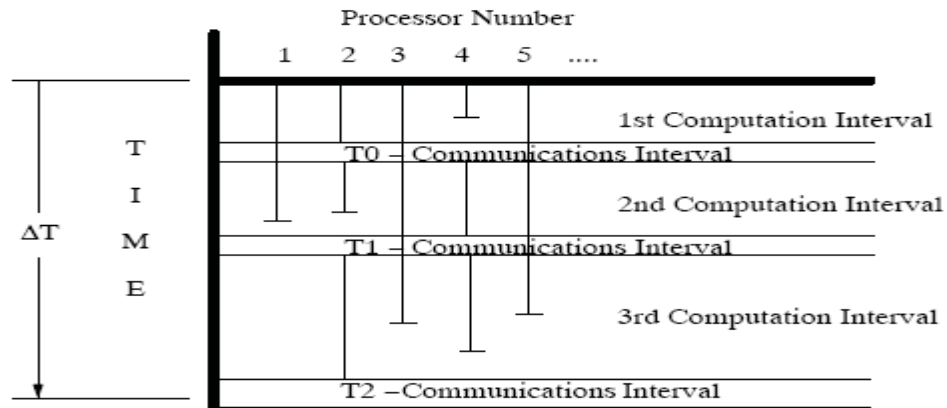


Figure 6.23

Communication and computation intervals between different processors in subsystems [18]

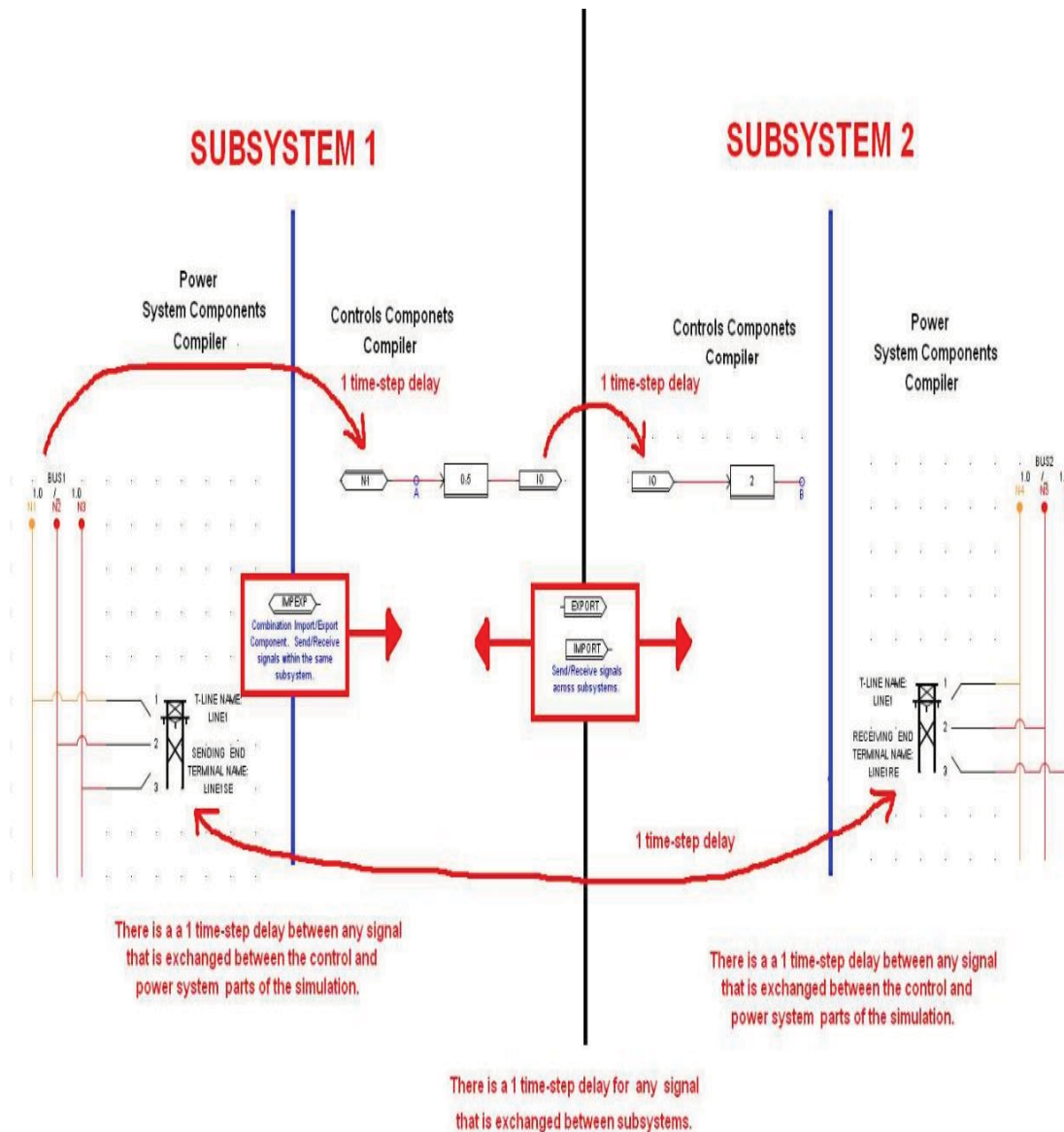


Figure 6.20

Power system and control system signal exchange between Subsystems

## 6.7 Discussions

Until now, different options and challenges of distributed simulations have been discussed. The modeling and development of two different kinds of test beds using the RTDS have been presented. First, a test bed for data transfer between two Real Time

Digital Simulators has been developed and validated with different kinds of remote monitoring experiments. The remote monitoring experiments were performed between MSU and TAMU during fault and no fault conditions. The results of these experiments are shown in all the three stages of data transfer. The three stages consist of Rack1, LabVIEW client and Rack2. Data rates have been calculated using two different data capture programs, such as C# and LabVIEW and are tabulated. The data rate calculation experiments were performed using 118 bytes and 36 bytes of data from GTNET. The 36 bytes of data transfer proved to be a better option as compared to the 118 bytes when the delay was considered. Three main experiments were considered during data transfer between the simulators. These are

- Data transfer without archiving the data
- Data transfer with archiving and without parsing
- Data transfer with archiving and parsing

Then, a wide area control test bed was designed using the SVP and RTDS. An angle difference scheme was implemented using the SVP and the test bed was validated. Finally the test case used for wide area monitoring was simulated in decoupled and non decoupled fashion. The simulation time steps were compared and analyzed.

For distributed simulation, when power system signals are exchanged externally (via A/D and D/A converters) between the racks, care should be taken with respect to the inherent time delays in the conversion process, which can cause the simulation to become unstable. One of the important steps required to perform distributed simulation between two racks is the modeling of a transmission line. The transmission line model should obey the physical relations governing transmission line signal propagation. The sending

end and receiving end transmission line signals, which are calculated from the transmission line equations, should be buffered. The communication of the buffered values should be fast enough so that these can be applied and/or injected after a delay, which appropriately represents travel time of that signal. These are some of the main aspects that should be considered while designing a transmission line model for distributed simulation. Then, the Global Bus Hub (GBH) hardware, which synchronizes the timing signal of each rack, should be replaced by another source which can make sure that two racks run in step. One of the conditions for successful decoupling of the power system is that the communication time should be faster than the travelling time of the shortest transmission line. So, the Inter Rack Communications card (IRC card) which provides dedicated communication channels should be replaced.

## **6.8 Summary**

In this chapter, the simulation results of wide area control and decoupled simulations were presented. The modeling of the test system in RTDS and the control algorithm in the SVP were discussed and analyzed. The signal exchange, between the subsystems has been presented using a schematic and the simulation time steps for decoupled and non decoupled cases were compared. Finally, the chapter concludes with some discussions pertaining to distributed simulation using RTDS.

## CHAPTER VII

### CONCLUSIONS AND FUTURE WORK

#### 7.1 Conclusions

Transient analysis and steady state analysis of a power system require the solution of non linear DAE (differential algebraic equations), which are computationally time consuming and require vast computational resources. With the trend towards more sophisticated and realistic modeling, computational time increases with size and complexity. Power system modeling and simulation are computationally expensive and distributed simulation provides a good way to implement the solution utilizing distributed resources. The concept of distributed simulation allows the power system to be partitioned into subsystems and processed simultaneously for easy circuit formulation and faster simulation.

Some of the main advantages of the distributed simulation are:

- Reducing computational time of the simulation
- Utilizing and interconnecting the distributed resources
- Improving the scope of remote studies and online operation and control studies

This research investigated the different hardware and software options for performing distributed simulation locally and geographically. It also analyzed and presented the different ways of getting the digital power system data in and out of RTDS. The work describes the development of the geographically distributed test bed for data transfer between two real time power system simulators. The test bed was validated with remote monitoring experiments under no fault and fault conditions. The data transfer rates for the power system data, were also calculated and compared with three different kinds of experiments. These experiments were performed for 118 bytes and 36 bytes of data transfer. Later, a wide area monitoring and control test bed, using the RTDS and SVP was developed. An angle difference scheme was implemented for the power system test case developed in RSCAD. Distributed simulation on this test case was performed locally using the IRC and GBH in RTDS. The results of decoupled and non decoupled simulations were presented and discussed.

The contributions of this work are:

- Different options for performing distributed simulation locally and geographically using the RTDS have been analyzed and discussed.
- A test bed for performing distributed simulation between two simulators has been developed and validated.
- Different data transfer experiments have been performed between Mississippi State University and Texas A&M University for remote monitoring.
- A wide area monitoring and control test bed has been developed and validated.
- Analysis of decoupled and non decoupled simulations has been carried out locally.

## 7.2 Future Work

Although the developed test bed allows data transfer between the simulators, geographically distributed simulation requires a transmission line model, which can deal with the delays during data communication. Further investigation of the high speed communication infrastructure, such as internet 2 provides advanced features like QoS (Quality of Service) and higher capacity for real-time remote instrument access and control. Further investigations as listed will provide the geographically distributed simulation higher flexibility and accuracy.

- Time stamping the data that is being transferred over the communication network for dealing with data loss, accuracy and synchronization.
- Extend the proposed data transfer test bed for remote interactions and data exchange between utilities.
- Increasing the scope for performing on-line transient stability analysis by fast calculations of differential algebraic equations in distributed simulation.

## REFERENCES

- [1] J. Arrillaga, N.R.Watson, Computer Modeling of Electrical Power Systems, 2<sup>nd</sup> Edition, John Wiley & Sons, Ltd, 2005.
- [2] Marek Zima, Mats Larsson, Petr Korba, Christian Rehtanz and Goran Andersson, “Design Aspects for Wide-Area Monitoring and Control Systems”, Invited Paper, *Proceedings of the IEEE*, Vol 93, No 5, May 2005.
- [3] Xiao-Ping Zhang, Christian Rehtanz and Bikash Pal, Flexible AC Transmission Systems: Modelling and Control, Birkhäuser, 2006.
- [4] Jian Wu, “A Generalized Three Phase Coupling Method for Distributed Simulation of Power Systems”, PhD Dissertation, Dept of Electrical and Computer Engineering, Mississippi State University, Mississippi, 2006.
- [5] S. Ayasun, S. Vallieu, R. Fischl, and T. Chmielewski, “Simulation-Stimulation Interface for Hardware in-the-Loop Studies”, *MED POWER*, Athens, Greece, Nov. 2003.
- [6] D. Potter, “Using Ethernet for industrial I/O and data acquisition,” *Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference*, 1999, vol. 3, May. 1999, pp. 1492 – 1496.
- [7] S. Ayasun, S. Vallieu, R. Fischl, and T. Chmielewski, “Electric machinery diagnostic/testing system and power hardware-in-the-loop studies,” *Proceedings of the 4th IEEE International Symposium on Diagnostics for Electric Machines*, Aug. 2003, pp.361 – 366.
- [8] J.L. Bastos, J. Wu, Noel N. Schulz, R.Liu and A.Monti “Distributed Simulation using the Virtual Test Bed and its Real-Time extension”, *Proceedings of the 2007 Summer Computer Simulation Conference (SCSC'07)*, San Diego, CA, July 2007.
- [9] Qinghua Huang, Jian Wu, Jimena L. Bastos, Noel N. Schulz, “Distributed Simulation Applied to Shipboard Power Systems”, *Proceedings of the IEEE Electric Ship Technologies Symposium*, 2007, pp.21-23 May 2007.



- [10] D. E. Barry, C. Pottle and K. Wirgau, "A Technology Assessment Study of Near Term Computer Capabilities and Their Impact on Power Flow and Stability Simulation Program," *EPRI-TPS-77-749 Final Report*, 1978.
- [11] F. M. Brasch, J. E. Van Ness and S. C. Kang, "Evaluation of Multiprocessor Algorithms for Transient Stability problems," *EPRI-EL-947*, 1978.
- [12] K. Lau, D. J. Tylavsky and A. Bose, "Coarse Grain Scheduling in Parallel Triangular Factorization and Solution of Power System Matrices," *Proceedings of the IEEE PES Summer Meeting, Minneapolis*, July 1990.
- [13] A. Abur, "A Parallel Scheme for the Forward/Backward Substitutions in Solving Sparse Linear Equations," *IEEE Transactions on Power Systems*, Vol. 3, No. 4, pp. 1471 – 1478, 1988.
- [14] Vladimir B.Dmitriev-Zdorov, "Generalized coupling as a way to improve the convergence in relaxation-based solvers", *Proceedings of the conference on European design automation*, Geneva, Switzerland 1996.
- [15] K. Martin and J. Carroll, "Phasing in the Technology", *IEEE Power and Energy Magazine*, September-October 2008, Volume 6, Issue 5, 2008.
- [16] A.G. Phadke, "System of Choice", *IEEE Power and Energy Magazine*, IEEE, September-October 2008, Volume 6, Issue 5, 2008.
- [17] J. Thorp, A. Abur, M. Begovic, J. Giri, R. Avila-Rosales, "Gaining a Wider Perspective", *IEEE Power and Energy Magazine*, September-October 2008 Publication, Volume 6, Issue 5, 2008.
- [18] Real-Time Digital Simulator and its Manuals <http://www.rtds.com/>
- [19] National Instruments <http://www.ni.com/>
- [20] NI-PXI Controller <http://zone.ni.com/devzone/cda/tut/p/id/4811>
- [21] Sunil Kumar Palla, "Development of Over Current Relay and Power System Simulator using National Instruments Devices in Real Time", M.S thesis, Dept of Electrical and Computer Engineering, Mississippi State University, Mississippi, 2008.
- [22] SEL Instruction Manuals, Application Guide, Other Related Documents <http://www.selinc.com/>
- [23] National LabVIEW Software, Toolkits, LabVIEW Manuals <http://www.ni.com/labview/>

- [24] Trapezoidal Rule of Integration <http://numericalmethods.eng.usf.edu>
- [25] H.W. Dommel, "Digital Computer Solution of Electromagnetic Transients in Single- and Multiphase Networks", IEEE Transactions on Power Apparatus and Systems, Volume: PAS-88, Issue: 4, April 1969.
- [26] Damien Tholomier, Laurent Schmitt and Jay Giri, "Impact of Recent Blackout: Return of Experience of Experience on Utility Operational IT Infrastructures", *2007 iREP Symposium- Bulk Power System Dynamics and Control - VII, Revitalizing Operational Reliability*, August 19-24, 2007, Charleston, SC, USA.
- [28] WinPcap: The windows packet capture Library <http://www.winpcap.org/>
- [29] Wireshark <http://www.wireshark.org/about.html>
- [30] Internet 2 <http://www.internet2.edu/about/>
- [31] V. Dmitriev- Zdorov, R. Dougal, V. Lyashev, M. Maksimov, V. Popov and E. Solodovnik, "Distributed Simulation of the Electromechanical systems in the Virtual Test Bed", *Fifth IASTED Intern Conf on Power and Energy Systems*, Tampa, FL, November 2001, pp. 248-252.
- [32] Micheal Kleinberg, Karen Miu and Chika Nwankpa, "Network Modeling for Distributed Simulations of Unbalanced Power Systems", *Proceedings of the 2007 Summer Computer Simulation Conference, SCSC 2007*, San Diego, California, USA, July 16-19, 2007.
- [33] J. Duncan Glover, Mulukutla S. Sarma and Thomas J. Overbye, *Power System Analysis and Design*, Fourth Edition, Thomson Learning Inc., 2008.
- [34] Prabha Kundur, *Power System Stability and Control*, Second Reprint 2006, Tata McGraw-Hill Edition, 2006.
- [35] Krishnanjan Gubba Ravikumar, Noel N. Schulz, and Anurag K. Srivastava, "Distributed Simulation of Power Systems Using Real-Time Digital Simulator", *Proceedings of the IEEE PES Power System Conference & Exhibition*, March 2009, Seattle, WA.
- [36] H. T. Su, K. W. Chan and L. A. Snider, "Investigation of the use of electromagnetic transient models for transient stability simulation", *Proceedings of the 6th International Conference on Advances in Power System Control, Operation and Management*, APSCOH2003, Hong Kong, November 2003.

- [37] J. Tang, R.Hovsapien, M. Sloderbeck, J. Langston, R. Meeker and P.G. McLaren, "The CAPS Power System Security Test Bed", CRIS, *Third International Conference on Critical Infrastructures*, Alexandria, VA, September 2006.
- [38] Langston, S. Suryanarayanan, M. Steurer, P. F. Ribeiro, M. Andrus and S. Woodruff, "Experiences with the simulation of a notional all-electric ship integrated power system on a large-scale high-speed electromagnetic transients simulator," in *Proc. 2006 IEEE Power Engineering Society General Meeting, Montreal, QC, Canada, June 2006*.
- [39] A. Guzman, S. Samineni and M. Bryson, "Protective Relay Synchrophasor Measurements during Fault Conditions" presented at the *Power Systems Conference: Advanced Metering, Protection, Control, Communication, and Distributed Resources*, 14-17 March 2006.
- [40] Edmund O. Schweitzer, III, David Whitehead, Armando Guzman, Yanfeng Gong and Marcos Donolo, "Advanced Real-Time Synchrophasor Applications" Schweitzer Engineering Laboratories, Pullman, March 2009.
- [41] K.K.Fung, S.Y.R. Hui and C. Christopoulos, "Concurrent programming and simulation of decoupled power electronic circuits", *IEEE Proc.-Sci. Meas. Technol.*, Vol. 143, No. 2, March 1996.
- [42] Richard Benjamins, John Davies, Elmar Dorner, John Domingue, Dieter Fensel, Ozelin López, Raphael Volz, Alexander Wahler and Michal Zaremba, "Service Web 3.0".  
Available online at:  
[http://www.sti-innsbruck.at/fileadmin/documents/technical\\_report/ServiceWeb3v4b.pdf](http://www.sti-innsbruck.at/fileadmin/documents/technical_report/ServiceWeb3v4b.pdf)
- [43] IEEE Power and Energy Society, "Power System Relaying Committee"  
Available online at: <http://www.pes-psrc.org/>
- [44] IEEE Std C37.118-2005 (Revision of IEEE Std 1344-1995), Standard for Synchrophasors for Power Systems, 2006.