Development of a Platform for Wide-Area Monitoring Applications

G V N Yatendra Babu

A Thesis Submitted to
Indian Institute of Technology Hyderabad
In Partial Fulfillment of the Requirements for
The Degree of Master of Technology

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Department of Electrical Engineering

December 2014

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

G. Yatendra.

(Signature)

(G V N Yatendra Babu)

CONMP03.

(Roll No.)

Approval Sheet

This Thesis entitled Development of a Platform for Wide-Area Monitoring Applications by G V N Yatendra Babu is approved for the degree of Master of Technology from IIT Hyderabad

B. Ravi

(Dr. Ravikumar Bhimasingu) Examiner
Dept. of Elec. Eng.
IITH

Y Pradeep

(Dr. Pradeep Kumar Yemula) Examiner
Dept. of Elec. Eng.
IITH

Vaskar Sarkar

(Dr Vaskar Sarkar) Adviser
Dept. of Elec. Eng.
IITH

Saswata Bhattachar

(Dr. Saswata Bhattacharya) Chairman
Dept. of MSME.
IITH

Acknowledgements

I wish to thank my supervisor Dr. Vaskar Sarkar for giving constant thrust and motivation. There are a number of people without whom this thesis might not have been written, and to whom I am greatly indebted.

Dedication

Dedicated to my family and my supervisor Dr. Vaskar Sarkar.

Abstract

Wide-area monitoring system (WAMS) is introduced to facilitate the power system operator with system-wide information. WAMS applications are designed to aid the system operator in operating power system efficiently. The Power system is considered as a critical infrastructure. Being critical infrastructure and complex system, integration of new applications into the system is not permissible without rigorous testing. In this thesis, a platform is developed for design and testing of WAMS applications. The platform is developed with the help of real-time digital simulator, data acquisition equipment, and visual interface. With the help of real-time tools, the platform emulates the power system in real-time. In-house development of visual interface is carried out for this platform. The visual interface is designed to incorporate user interaction as a component in the platform. The integration of the components are discussed in this thesis. The platform operation is demonstrated with 39 bus New England test system.

Contents

List of Figures

List of Tables

Nomenclature

α	impedance angle
δ	rotor angle
ω	machine speed
ω_s	rated synchronous speed
θ	Voltage angle
E'_d	d-axis generated emf
E'_q	q-axis generated emf
E_{fd}	field voltage
H	inertia constant
I_d	d-axis stator current
I_q	q-axis stator current
K_A	exciter amplifier gain constant
K_E	exciter gain constant
K_F	stabilizing transformer gain
m	number machines
n	number of buses
P_C	speed governor power change setting
P_L	load Active power
P_{SV}	steam valve position
Q_L	load Reactive power
R_D	percentage droop constant
R_f	exciter rate feedback
R_s	stator resistance
S_E	exciter saturation function
T'_{do}	d-axis open circuit transient time constant
T'_{qo}	q-axis open circuit transient time constant
T_A	exciter amplifier time constant

T_E	exciter time constant
T_F	stabilizing transformer time constant
T_M	mechanical torque to shaft
T_{CH}	steam chest time constant
T_{FW}	friction and windage torque
T_{SV}	steam valve time constant
V	terminal voltage
V_R	voltage regulator input voltage
V_{ref}	exciter reference voltage
X_d'	d-axis transient reactance
X_q'	q-axis transient reactance
X_d	d-axis steady-state reactance
X_q	q-axis steady-state reactance

Chapter 1

Introduction

The power system is highly complex in nature and huge in size. Operating such a system is difficult without thorough information about the system status. Traditional supervisory control and data acquisition (SCADA) collects system information from several sources and presents it to the system operator. However, the data reporting rate of SCADA is in the order of several seconds. This is only suitable to analyze steady state status of the system. Different data sources of SCADA will have different time references, which provides unsynchronized data. To fill the gap in the existing monitoring technology, wide-area monitoring system (WAMS) is introduced. Main components of WAMS are measurement equipment, communication networks, and data evaluation algorithms (WAMS applications) [1]. WAMS applications are critical to take system level decisions. Any misinterpretation of the data may lead to system collapse. So, before integrating any applications into WAMS, thorough testing of the application is required. Available options for testing WAMS applications are field testing and off-line testing. In field testing, targeted WAMS application is deployed in field and evaluates the performance of the application. In this case, the test cases are limited as it is not possible to create wide range of situations. In case of off-line testing, several cases can be constructed as per the requirement. The targeted application can be tested with the constructed test cases. This demands the need of a platform for development and testing of WAMS applications.

1.1 Constraints in Existing Methodologies

A replica of an actual system is required to test the future technologies or theories to validate their effectiveness. However, it is very hard to build the physical replica of a realistic power system at laboratory level. Therefore, scaled physical replica and digital emulation of the system are available options. Scaled physical replica is emulating system behavior with the basic elements such as resistor, inductor, and capacitor. In this method, it is not possible to replicate big systems because of cost involved and space requirement. Besides, significant loss of details occurs in this method of replication.

The system studies involve solving a set of equations. Most of the simulators and/or tools available in the market are off-line tools. In case of off-line tools, the execution time is not deterministic due to several reasons, such as software and hardware limitations. Off-line tool can be adapted in use of real world application by running it in real-time. In order to perform time constrained execution of the simulation, special design of software and hardware is required.

Real-time digital simulators [2] are designed to simulate system model in real-time. Real-time digital simulator is basically a computer with real-time software and interfacing hardware. These simulators provide facility to connect external hardware (sensors, controllers, and so on). In this method, simulator's processing capability limits the targeted system size. However, it is possible to build system without losing significant details.

1.2 Proposed Methodologies

Real-time digital simulator alone is not sufficient to replicate complete feature set of power system. The power system is a combination of communication system, monitoring system, core power system components and so on. So, other modules should be added to simulator in order to emulate power system.

Keeping the above mentioned issues in view, this thesis discusses the development of a real-time platform for the emulation of power system by using Real-time digital simulator, data acquisition, and visual interface tools. These components works with different technologies and so, many constraints are involved in developing an integrated environment. In this work, an effort was made to integrate different technologies in order to develop a platform, which enables the development and testing of WAMS applications.

Real-time digital simulator is used to simulate core power system model in real-time. The particular simulator is equipped with the input-output ports (digital and analog) to communicate with external hardware. The power system model is built in such a way that, it will use data from hardware ports. For example, the analog input and output ports can be used to receive set-points and send measurements respectively. Besides, the digital input and output ports can be used to receive external trip signal and send relay status.

National instrument-PCI eXtensions for Instruments(NI-PXI) [3] with relevant acquisition cards is used for data acquisition. To meet connection requirements, an interface module is used to connect simulator ports and NI-PXI. A program is developed in LabVIEW to process the data received by hardware ports. A data logging software is developed in LabVIEW to log the data in database. In this platform, MySQL is used as database server to store data. An interface program is developed to bridge data acquisition hardware and database server.

The visual interface helps in visualizing the generated data by simulator. It is equipped with basic user actions, such as set point changes and external trigger to the system. This visual interface can play a vital role in developing WAMS applications, which needs user interaction. Besides, it will help to study the effect of user decision or interaction on system operation and system behaviors under emergencies [4]. Python is used to develop the visual interface.

1.3 Organization of The Thesis

The remaining part of this thesis is arranged as follows: In Chapter 2, a brief introduction to WAMS is provided. In addition to that existing platforms in this area are reported. In Chapter 3, conceptual framework and requirements of the platform are presented. In chapter 4, implementation of the proposed platform along with the software and hardware aspects are reported. Conclusions are presented in Chapter 5.

Chapter 2

Wide Area Monitoring Systems

2.1 Introduction

Monitoring system is combination of measurement equipment, communication networks, and so on. As mentioned in Chapter 1, thorough system-wide information is required in order to operate the grid efficiently. Conventional monitoring system of power system consists measuring devices such as voltage and current sensors. These conventional devices measures data from different locations and transmit to control center. Monitoring applications in control center suggest corresponding actions based on the data. Sometimes, the data generated by measuring devices might be incomplete or inconsistent and the measurements from different locations consist significant error in time reference. In other words, conventional monitoring system often provides incomplete and unsynchronized measurements. In general, the data reporting rate of conventional monitoring system is in the order of seconds. Under disturbed conditions, system behavior changes rapidly. In order to apply efficient remedial actions, thorough monitoring of the system is required. Conventional monitoring system limits the capability of observing the system behavior under rapid changing conditions. With the unsynchronized data, it is also very difficult to perform post fault analysis without knowing exact time stamps of the measurements.

In order to overcome the difficulties faced by conventional monitoring system, WAMS is introduced. WAMS can be designed to complement the existing monitoring systems such SCADA. WAMS is based on time synchronized measurements in phasor measurement units (PMU) and phasor data concentrators (PDC). These devices are deployed at selected substations and connected to central facility with low latency communication network.

This chapter provides an introduction to WAMS. Few monitoring applications are also explained in brief. In addition, some of the existing platforms are revisited.

2.2 Local Protection, SCADA, and WAMS for Monitoring

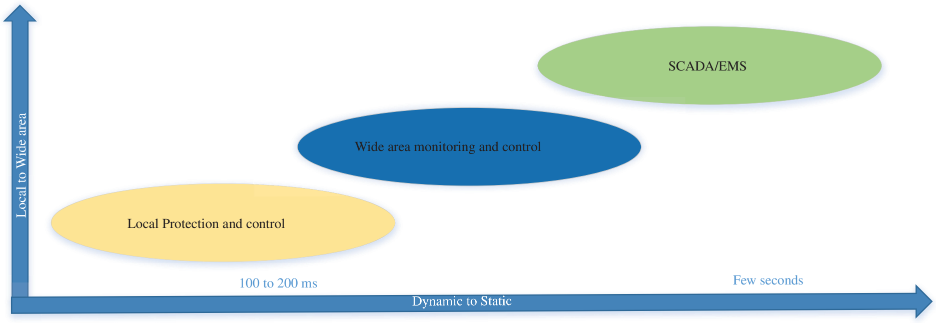


Figure 2.1: Comparison of monitoring systems

Timely detection and responding to an emerging problematic situation is critical to save the power system from a system wide failure. As mentioned earlier, the conventional monitoring system mainly consist of local protection and SCADA. This local protection and control equipment responds to disturbances in tens of milliseconds. However, these control actions (operating circuit breakers and so on) are not devised by considering system-wide response. As soon as it responds to disturbance, it sends a message to the control center. In control center, SCADA monitors the changes caused by disturbance in system by processing the status messages sent by local equipment. Here, the SCADA applications provide control action suggestions. Because of the amount of data to process and the length of communication network in large interconnected systems, the response rate of SCADA is in order of seconds. In abnormal condition, the system state changes rapidly. Under certain conditions, the SCADA delivers outdated solution, which is irrelevant to the situation. To fill the gap in reaction time by complementing the existing technology, WAMS is incorporated into the monitoring system. Comparison of different levels of monitoring and protection equipment with respective to the response time and system coverage is shown in Figure 2.1. To ensure fast response, the communication network must provide sufficient bandwidth for data transmission. Comparison [1] of reaction time, reporting rate, and quantity of data is presented in Table 2.1.

Table 2.1: Comparison of local protection, WAMS, and SCADA

	Protection devices	WAMS	SCADA
Coverage	Local	Global or Regional	Global or Regional
Reporting rate	Only when event occurs	Continuous with 20 ms rate	Continuous up to few seconds
Reaction time	Tens of milliseconds	Hundreds of milliseconds	Few seconds
Data quantity	Few messages	High	High

2.3 WAMS Overview

The main components in WAMS are: PMUs, PDCs, communication networks, data storage, and application software. Typical WAMS structure [1] is shown in Figure 2.2. The number of PMUs and PDCs are chosen according to power system requirements.

2.3.1 WAMS Architecture

Figure 2.2 shows general architecture of WAMS. PMUs are installed at selected substations or in all substations as per the system requirement. All of the PMUs are synchronized to a common time source. The PMUs transmit the information to the PDC, which is another important device in this structure. These PDCs are located at substation, regional center, and central control center.

All devices are connected to high speed communication network. As this PMU technology uses IEEE C37.118 standard for communication, any type of communication system with sufficient bandwidth can be used to support this monitoring system. In particular, the central PDC needs high bandwidth to support the volume of data. Central PDC acquires information from local, regional PDCs, and PMUs. Central PDC aligns the received data according to the time reference. At each stage, PDC provides information to system operator to assess the system state. But the assessment is limited by the information available with the corresponding PDC. Transmission system operator (TSO) PDC is an individual

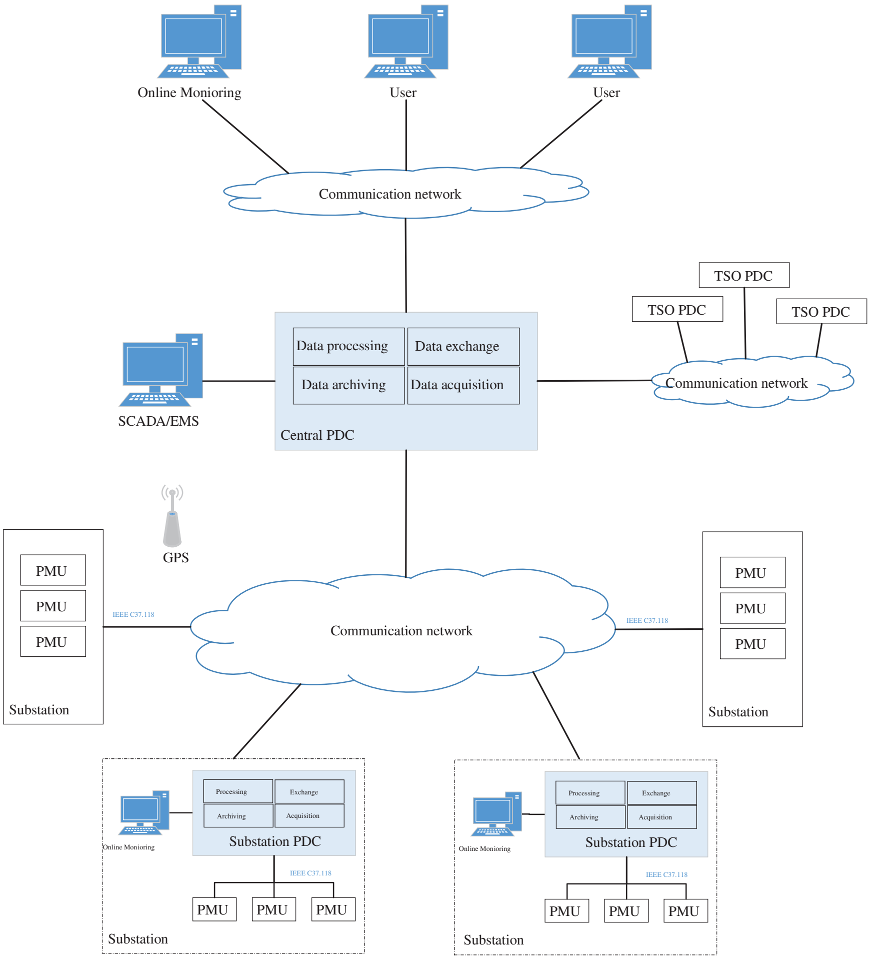


Figure 2.2: General WAMS structure.

component to serve TSO's monitoring systems to monitor their respective boundaries. So, the information available with the TSO-WAMS may not necessarily be a system-wide information. However, the information exchange depend on the system environment and participants in the system operation. For efficient assessment of the system state, it is advised to have system-wide information. With the available information, a typical WAMS can provide the following advantages [5]:

- real-time visualization of power systems,
- effective utilization of installed capacities,
- design of an advanced early warning system,
- design of adaptive protection and control system,

- real-time congestion management and so on.

Phasor Measurement Units

PMU is a device, which measures voltage and current on power system [6]. These measured quantities are time stamped with common time reference. The common time reference provides time synchronization of measurements throughout the system. The measurement equipment should maintain synchronism within 1μs of timing source. The satellite based global positioning system (GPS) is used as the synchronizing source as its widespread availability makes it possible to use it in remote places. In addition, PMU is capable of providing calculated quantities such as active power and reactive power. PMU can be configured to provide special functions, such as protective relay, fault recorder, and so on. The information generated by PMU can be stored locally or transmitted to a regional or central location in real-time.

Phasor Data Concentrator

PDC can be implemented as a stand-alone unit to receive data and store it into a data storage or redistribute the data to other devices. The PDC can be installed at substation, regional center, or central control center. Substation PDC serves as a local data storage in case of communication failure. Substation PDC acquires data from PMU and distributes it to regional PDCs and local SCADA. In addition, substation PDC can be configured to perform control actions. However, substation PDC alone is not sufficient to fulfill the requirements of WAMS. Regional PDCs are installed to receive the information from several PMUs and PDCs. These regional PDCs redistribute the data to other regional PDCs, central PDC, and regional SCADA. Central PDC collects information from regional PDCs and acts on that information.

PMU and PDC uses IEEE standard C37.118 for synchrophasor data transfer. Based on this standard, data can be transferred by using any communication system. However, the only constraint is that the communication system should provide sufficient bandwidth. The arrangement of PDCs at different levels are shown in Figure 2.2.

2.3.2 Centralized and Decentralized WAMS

Centralized WAMS

Structure of centralized WAMS [1] is shown in Figure 2.3. TSO PDCs serves to TSO WAMS. In addition to that, they exchange data with central PDC. In centralized WAMS, only one central PDC is used. All TSO PDCs are connected to central PDC. Stability analysis can be executed to great extent by using the data available at central PDC. Besides, coordinated alarming and corrective actions can be executed in case of disturbance. Centralized WAMS facilitates easy administration of data exchange between TSO and central WAMS. However, the disadvantage of being central is, if any communication link or central PDC fails then the entire WAMS becomes blind.

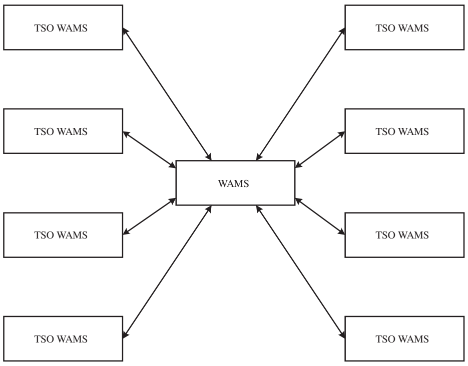


Figure 2.3: Centralized WAMS

Decentralized WAMS

Decentralized or distributed WAMS [1] contains multiple PDCs instead of single central PDC. The PDCs can be at different levels i.e. TSO, regional, and etc. The distributed WAMS comparatively provides more reliability in case of individual equipment failure. In this structure, coordination is difficult. Full scale stability analysis is not possible as the complete data is not available. This structure requires high capacity communication network as the same data sent to multiple devices. Decentralized WAMS is shown in Figure 2.4.

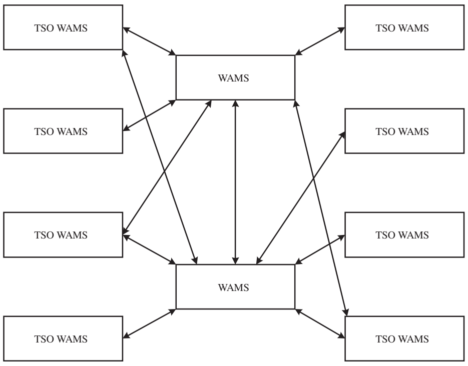


Figure 2.4: Decentralized WAMS

2.4 WAMS Applications

As mentioned earlier, WAMS applications play vital role in operating the power system. WAMS applications are data evaluation algorithms [5]. These algorithms use the data from WAMS devices to provide analytic to system operator. With these applications, the system operator can judge the system state efficiently. Some of the WAMS applications are presented below.

2.4.1 Phase Angle Monitoring

PMU, as a basic element in WAMS provides voltage and current measurements along with their phase angles. Voltage phase angles of any two buses in a power system can be compared using this technology. Phase angle difference signifies stress on the power system. An early warning mechanism can be devised to support system operator. With this application, early remedial actions can be deployed. Rate of change in angle difference can also be used for early warning mechanisms. Another use of the phase angle monitoring is in reconnecting two isolated buses. Closing of the circuit breaker shall happen only when the angle difference is below pre-set value. It can be executed with the help of this application.

2.4.2 Oscillation Detection

The purpose of oscillation detection is to detect electromechanical oscillation modes in power system. Inter-area, intra-area, and local oscillations are different modes of oscillations that are present in the system. Generators from different areas contribute to the inter-area oscillations. Generators from same area participate in intra-area oscillations. Local oscillations involve only one generator. These oscillations may cause tripping of lines or generators. The events may lead to severe disturbance.

Under stable operation of power system, these oscillations are damped with the help of controller. the undamped oscillations may become dangerous for integrity of the system. So, this application shall detect the mode of oscillations based on the available measurements.

2.4.3 Islanding Detection

When a severe disturbance occurs in system, angle and/or frequency difference between two buses or areas tends to exceed the safe limit. With this situation, the local protection scheme disconnects the existing link between two areas. If there is no other link presents between two areas, it develops an islanding situation. Islanding detection application shall identify the islanded portion based on available information.

2.4.4 Disturbance Propagation Monitoring

With this disturbance propagation monitoring, an early warning mechanism can be implemented to deal with emerging system-wide disturbance. The system-wide information enables system operator to monitor dynamic events. In some events, the disturbance can become catastrophic and spread to healthy part of the system. This disturbance propagation involves:

- cascading equipment tripping,
- cascading line tripping due to dynamic line loading,
- voltage collapse, frequency excursion, and so on.

WAMS along with disturbance propagation monitoring provides system-wide view to manage such events.

2.5 Testing Platforms

WAMS provide opportunities for better utilization of resources and efficient operation of power system. This opens the doors for innovative applications to deal with the problems of the system. However, as a critical infrastructure, power system operators does not allow integration of application from shelf. Because, the intensity of impact on society is more in case of application failures. Application failure includes providing wrong analytic or insight of the system status. As mentioned earlier, thorough testing of the application is required prior integration into system. Few institutions implemented test platforms to serve for smart grid studies. Some of them are relevant to this work and are presented below.

Smart transmission system laboratory (SmarTS Lab) [7] implemented a hardware and software based system for developing wide-area monitoring, protection, and control systems (WAMPAC). In this implementation, Real-time hardware in the loop (RT-HIL) approach is followed. In this approach, a real-time simulator along with software and hardware modules is used. The power system models are implemented in real-time simulator, communication network is emulated using a software module in a PC. External PMUs and protective relays are interfaced to real-time simulation through voltage and current amplifiers. These PMUs and relays are connected to communication network emulator, where it mimics the communication system. Besides, an external controller is also connected to simulator via voltage and current amplifier. This external controller receives control signals from WAMPAC application. These WAMPAC application is deployed in openWAMS platform.

Smart grid demonstration and research investigation lab (SGDRIL) [8, 9] developed a test bed for validation of WAMS devices and applications. Conceptually, SGDRIL implementation is same as SmarTS lab's implementation. However, the hardware specification varies from one to each other. In this implementation, a giga transceiver network interface card (GTNET) is used to communicate with devices such as PDC and other monitoring applications. In this test bed, SEL SynchroWAVe Central Software is used to translate synchrophasor data into visual information.

There are other test beds available in this area [10–16]. The main prospective of these test beds is to validate the communication protocols and smart grid devices. In this type of implementations, power system model is simulated in simulation platforms such as openDSS, PSCAD and so on. and these simulations are interfaced with communication network emulators. These test beds may not able to provide complete insight of target WAMS application.

2.6 Summary

A typical WAMS has to monitor power system in real-time. Besides, it should alert the system operator if any violations occurs in the system. Data archiving is also an inherent part of the WAMS. The archived data is helpful in performing post disturbance analysis. WAMS applications play a vital role in providing alerts to the system operator. These WAMS applications need to be developed and tested thoroughly in real environment. In centralized environment, these WAMS applications mainly act as visual aids to the system operator. These applications present the data to operator and collects the response from operator. So, both the back-end algorithms and data visualization should be developed and tested in real-time. For this purpose, a test platform can be developed by integrating different technologies together at laboratory level.

Chapter 3

Conceptual Framework of Platform

3.1 Introduction

The purpose of this framework is to provide a platform for development and testing of WAMS applications. As mentioned in Chapter 2, power system, communication, and visualization systems need to be emulated by the platform. The integration of these systems shall provide flexibility to add new devices and applications. This flexibility is important indeed, as this platform serves as testing facility. So, integration of new applications and devices is inevitable. In this chapter, a framework is presented to build a platform. In this framework, the platform is divided into different segments. The segmentation minimizes the dependency on each other. A general architecture is presented in this chapter. This architecture is prepared to emulate power system, communication system, and visual interface. Further, a work flow is also suggested in this chapter.

3.2 Segments in Platform

The entire platform is dived into three segments. These Segments are interconnected to each other as shown in Figure 3.1. Three segments are named as system model, visual interface, and data controller.

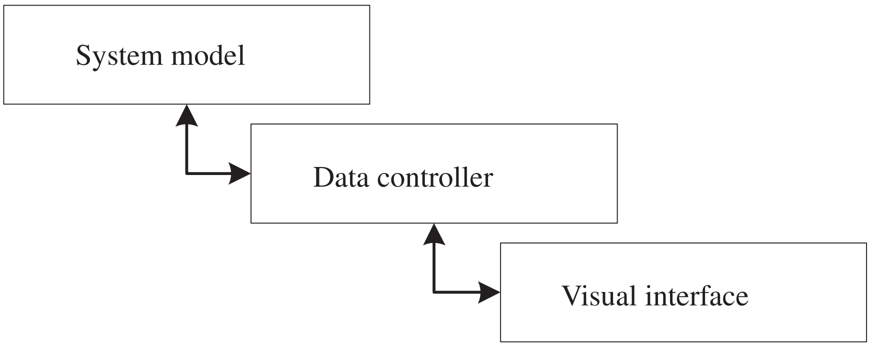


Figure 3.1: Segments in platform.

3.2.1 System Model

System model is core component in this platform. A system model can be a single object or combination of objects. In this application, the system model manages the power system model. System model is expected to simulate power system in real-time. In addition to that, it is capable of providing input to model from external devices and send model data to external devices.

3.2.2 Visual Interface

Visual interface is graphical representation of data generated by system model to operator. This segment runs in general PC environment, which gives non deterministic execution times. Hence, this segment is not a real-time system. Visual interface is an end user application, so it can have multiple instances. This interface captures user actions and inputs it to power system model through data controller.

3.2.3 Data controller

Data controller acts as a bridge between system model and visual interface. The main objective of data controller is to maintain data exchange in between system model and visual interface. This data controller acquires data from system model which is running in real-time and delivers it to visual interface which is not a real-time system. So, the data controller handles data transition between real-time and normal systems.

3.2.4 Data Exchange between Segments

Data exchange between different segments is illustrated in Figure 3.2. The data stored in database is presented to user by visual interface. Data acquisition, data latch, data buffer, data writing and data reading are considered as data controller. The application is a client software, that provides visualization to user and the functionality can be extended by adding software modules.

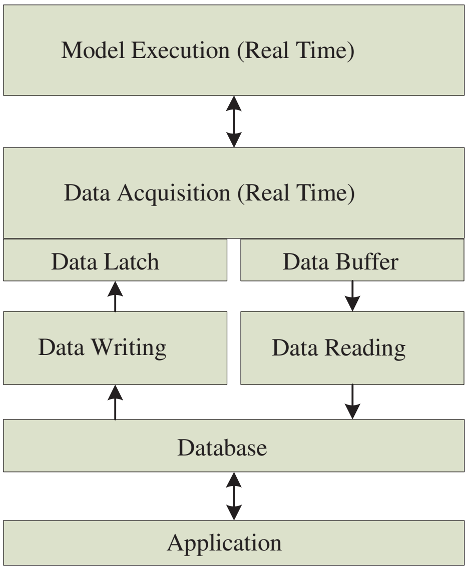


Figure 3.2: Data exchange between segments

In case of sending data to visual interface from data acquisition equipment, data need to be buffered in order to avoid the data loss while sending. Data need to be latched to previous value until next value posted by the visual interface in case of acquiring data from visual interface into data acquisition equipment. If real-time applications are used in visual interface, then there is no need of buffer.

To summarize, system model simulates power system in real-time and behaves like real world power system. Data controller emulates communication system. These are not necessarily be same technologies. However, there should be a facility to add external devices of same technology or other. This constraint makes the platform more generic and opens opportunities to perform experiments with external controllers or devices.

3.3 General Structure of Platform

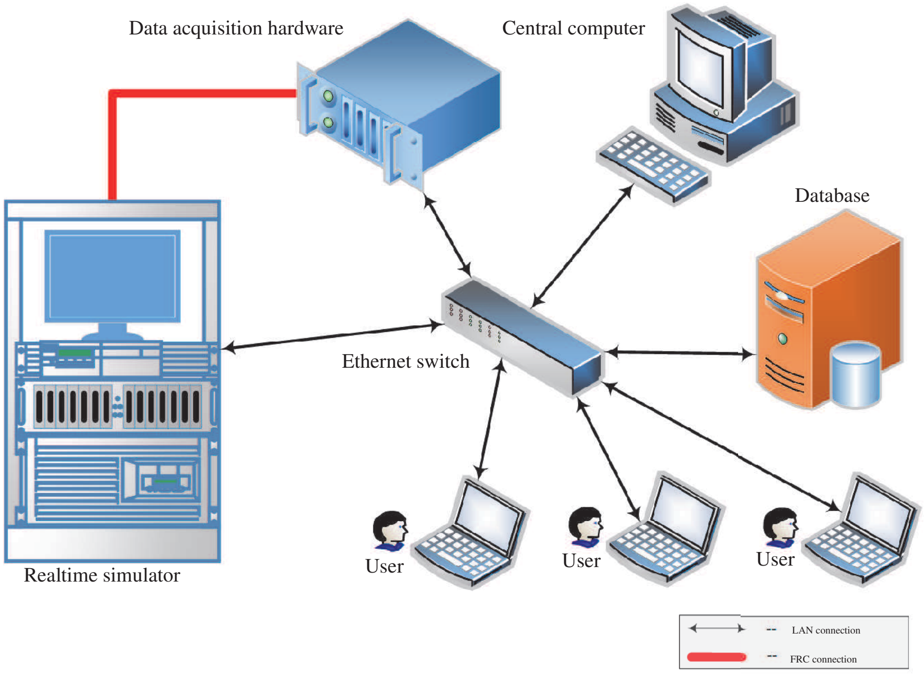


Figure 3.3: General structure of platform.

Physical structure of platform is shown in Figure 3.3. Components of this structure are:

- Real-time simulator,
- Data acquisition hardware/equipment,
- Database,
- Central console,
- Client/user console,
- Ethernet Switch.

This platform is a combination of both hardware and software. Power system is simulated by the real-time simulator in real-time. The data generated by the simulator is exchanged with the data acquisition hardware. These two system are connected through digital and analog ports. As both these devices are emulating real world systems, they are expected to run in real-time.

Data acquisition hardware/equipment is a combination of hardware and software, which emulates the communication system. This equipment exchanges data with simulator using hard-wire connection and database and visual interface using ethernet connection. This equipment shall provide required facility to handle data transfer as mentioned earlier. With all the required features, data acquisition equipment occupies data controller segment shown in Figure 3.2.

Database is used to store the data generated by the simulator for user applications. Visual representation of the power system, which is running in simulator is available at user console/computer. User console is the key component in developing WAMS applications. This console reads data from the database and presents it to the user. User console is visual interface in the structure. Visual interface being client software, it will have multiple instances. All of the devices in this structure are connected through a ethernet switch, which establishes and maintains a communication link between the devices.

A central computer/console is used to manage and monitor the equipment from a single window. The central computer is mainly used to build and deploy the models. Besides, it can be used to monitor the data at intermediate points. In other words, data in simulator and data acquisition equipment are accessible in central console. As all of the devices are connected to a single network, it is possible to access the data of all the segments at one location is possible in this structure.

External hardware can be added to this structure at two points, one is at simulator and another is to data acquisition equipment. External software modules such as WAMS applications can be integrated in this structure at any segment based on their requirement. The structure presented here is not restricted to a specific power system configuration. With the appropriate models, any system or case can be emulated by this structure.

3.4 Workflow for Platform

In order to work with the platform, required models should be developed for each of the segments mentioned above. The list of tasks to be performed are shown in Figure 3.4.

Second task contains three sub tasks. The subtasks are real-time model building, visual model building, and data model building. These three corresponds to the three segments explained earlier. The available information (bus data, line data, available ports, and so on) is used in model building. The information is divided into several sets to provide the required information for these sub tasks. It is possible to perform these subtasks independently without having knowledge on each other, because the information delivered from first task is sufficient at individual level.

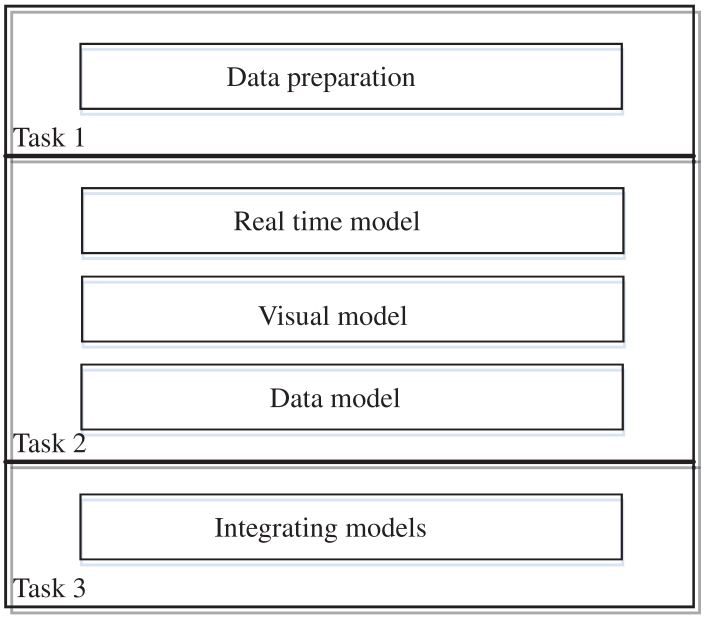


Figure 3.4: Work flow

Real-time model building is modeling of power system components. The modeled power system components are combined together to build the entire power system model. The required system specification and parameters are provided by the data preparation task/module. This task is very crucial. Any inaccuracies in system modeling may lead to wrong results at the end. So proper care should be taken in modeling these components. For a given power system it is onetime job, but if the system configuration changes, the model also gets changed. But the changes in component models are minimum, so reusing of component models are possible. This model will be running in real-time simulator. As mentioned earlier this model will be the core power system replica, and serves as the real core power system in the platform.

Visual model building is representing system model graphically by using single diagram. Data model building consists designing database, signal routing, and so on. Once these three models are ready, the integration is the final task. If the model building process followed as per the data provided by the data preparation task, this task is pretty straight forward. In this task, all the models can be deployed into corresponding devices. This concludes the basic platform. If any application is proposed for testing with this platform, the application should be integrated at corresponding location in the platform. In most of the cases all of the models have an open ends to provide or receive data, so it is quite possible to integrate the application in to platform.

3.5 Summary

A Framework is developed to maintain generality in the structure. Platform is divided in to three segments, system model, data controller, and visual interface. Data exchange between different segments is happen through ethernet and hard-wire connections. However, for data exchange, corresponding software modules are required. A general structure of the platform showing different components is presented. To work with the platform, a work flow is also suggested.

Chapter 4

Implementation

4.1 Introduction

As mentioned earlier, the implementation of the framework involves usage of different devices and technologies. This chapter provides description about integration of these devices. This platform is a combination of off-line and real-time systems. Real-time systems are power system model and data acquisition. However, part of the data acquisition and visualization applications are not a real-time systems.

For power system simulation, real-time digital simulator is used. National instrument's products are used to implement the software and hardware parts of the data acquisition. MySQL database server is configured to store the data generated by the power system model. Python is used to design the visual interface.

4.2 Hardware and Software Resources

The selection of hardware and software to implement the platform shall support interconnection of each other. Otherwise, connection adapters shall be available. Compatible software and hardware are required to implement individual components such as power system model, data acquisition application, and visual interface.

4.2.1 Real-time Simulator

The real-time simulation of the power system is carried out in the OPAL-RT product eMEGAsim [17]. eMEGAsim is capable of running simulation with step size as small as 10μs; however the step size is determined by the model complexity and hardware capabilities. The hardware description is given in Table 4.1.

Table 4.1: eMEGAsim hardware description

Features	Description
Number of Cores	2 HIL boxes each with 12 Intel i7 3.33 GHz cores
Analog Inputs	48 no (+/- 100V)
Analog Output	48 no (+/- 16V)
Digital Input	192 no (+/- 4 to 30V)
Digital Output	192 no (+/- 4 to 30V)

The eMEGAsim is supplied with a software (RT-LAB) [2] to develop, deploy, execute, and monitor the system model. RT-LAB is fully compatible with MATLAB. MATLAB allows users to develop detailed power system models. So, the power system model is developed in MATLAB and deployed in eMEGAsim.

4.2.2 Data Acquisition Equipment

For data acquisition, the equipment shall be capable of running in real-time. NI-PXI platform with PXIe-8115 controller is used as a core component in data acquisition equipment. This controller consists an Intel Core i5 2.5 GHz processor. From here onwards, NI-PXI and allied hardware is referred as NI-PXI.

NI-PXI is capable of running in real-time, as it is equipped with real-time operating system. It can be used to emulate communication system. For data acquisition from eMEGAsim input-output cards are used along with NI-PXI. Input and output hardware description [18–20] of NI-PXI module is given in Table 4.2 as per the usage.

Table 4.2: NI-PXI hardware description.

Component	Purpose	Connection ports
NI 7811R	Digital Input Digital Output	80 (0 to 5.5V) 80 (0 to 3.3V)
NI 6254	Analog Input	32 (+/- 10V)
NI 6723	Analog Output	32 (+/- 10V)

The maximum voltage of digital channels is 3.3V. However, eMEGAsim input range starts from 4V. So, eMEGAsim fails to detect the high signal from NI-PXI. Another issue is with the physical connectors, eMEGAsim has DB37 connectors whereas, NI-PXI has DB68 connectors. In order to match the connection requirements, a custom interface module is used as a part of data acquisition equipment. This interface module matches the voltage levels of both the equipment. In addition to that, the interface module acts as a connection adapter. Interface module has a 5V DC supply to raise the digital channel voltage to maximum 5V. In case of connectors, this module is equipped with DB37 and DB68 connectors. With the help of interface module, the eMEGAsim and NI-PXI exchanges data. Physical connection structure in between eMEGAsim and NI-PXI is shown in Figure 4.1. This concludes the hardware part of the data acquisition equipment.

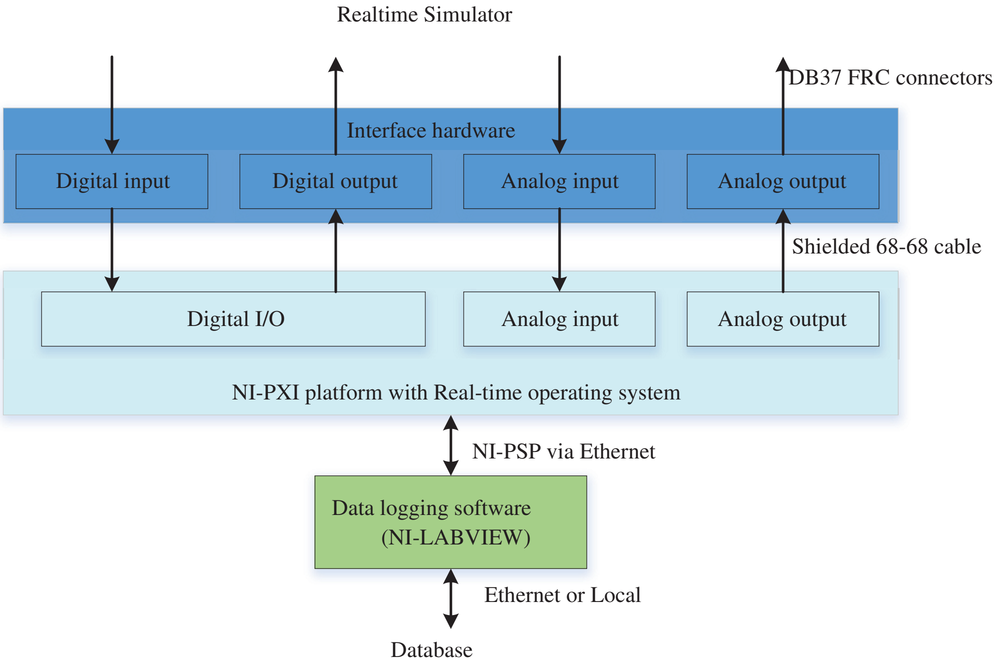


Figure 4.1: Block diagram of Data Acquisition System.

The acquired data through the ports has to be rearranged and exchange with the other devices. This task is the software part in this data acquisition. The default platform for application development for NI-PXI is LabVIEW. So, LabVIEW based application is developed for data exchange. In order to exchange data, this application needs to run in real-time.This data acquisition application exchanges data with eMEGAsim through NI-

PXI ports via interface module.

As a part of data acquisition, the received data need to be stored in database. For this purpose, a data logging software is developed in LabVIEW. Data reading or writing operations take some time to execute, so it cannot be executed in real-time. Hence, this application will be executed in normal PC. This data logging software exchanges data with the data acquisition application in NI-PXI though national instrument's publish and subscribe protocol (NI-PSP) [21]. NI-PSP is a national instrument's (NI's) proprietary communication protocol, which is compatible with ethernet connection. The connection structure is shown in Figure 4.1.

NI-PXI and LabVIEW are used to implement the entire data acquisition segment. This combination allows reconfigurable data acquisition system. LabVIEW software modules facilitates this equipment to implement any communication system as per the requirement.

4.2.3 Visual Interface

In this work, database and graphical user interface (GUI) are considered as a part of visual interface. This part of the platform is completely a software application. MySQL is used as a database server. GUI application is expected to exchange data with wide range of applications. At the same time, the development environment should allow modular design. Most of the times, the WAMS applications are integrated at user level i.e. in GUI application. The integration of application can be as a back-end service or as a front-end visual aid. For GUI application, Python programming language is used. The hardware and software used in this platform are summarized in Table 4.3.

Table 4.3: List of software and hardware resources

Segment	Type	Description
System model	Hardware	OPAL-RT's eMEGAsim
System model	Software	MATLAB and RT-LAB
Data controller	Hardware	NI-PXI and allied hardware
Data controller	Software	LabVIEW
Visual interface	Software(database)	MySQL
Visual interface	Software(GUI)	Python

4.3 Data Transfer Requirement between Devices

A typical data flow between the systems is shown in Figure 4.2. As mentioned earlier, data exchange between eMEGAsim and data acquisition system is through physical data channels of the respective elements, and remaining system is through ethernet connection.

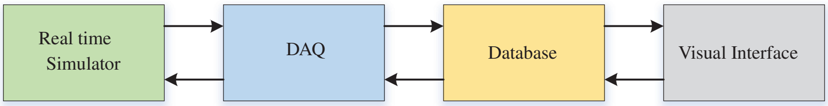


Figure 4.2: Data flow between systems

In this application, the data generated by the system model in eMEGAsim can be:

- voltage magnitude and angle,
- current magnitude and angle,
- frequency,
- calculated values such as active and reactive power (if incorporated in system model),
- controller set-points,
- relay, circuit breaker, and equipment status.

Relay, circuit breaker, equipment status data can be transmitted through digital ports, as the data mainly contains either 1 or 0. Remaining data is numeric. It provides two options for data transfer. First option is, converting numerical data into a digital sequence and transferring it through digital ports. Second option is, directly transferring through analog channels. In this application, analog ports are used to transfer the data.

For a given test system, number of measurements or data points are very high when compared to the available ports. For example, 39 bus power system gives a minimum of 200 measurements. However, the choice of measurements remains in user hands. For WAMS applications choosing large number of measurements is inevitable for clear picture of the system. It is not possible to use dedicated port for each measurement.

In this platform, to resolve the issue, data multiplexing is used to transfer the data. For multiplexing, few ports are reserved for select lines. Select lines are numeric values or combination of bits, which provides indexing. The arrangement of the data can be visualized as a two dimensional matrix. First, the data generated by the system model is serialized and then column-wise rearranged into a two dimensional matrix. To identify the data transmitted, the select lines should accompany the data. So, the select lines are padded at the end of matrix. At the end, number of columns is equal to number of ports (including select lines). This multiplexing applies for both digital and analog ports. The end result of the data format of a single row is shown in Figure 4.3. Data 1 to Data n is the data generated by the eMEGAsim and select line is the data identifier.

Data 1	Data 2	——	Data n	Selection Line

Figure 4.3: Data Frame

In general, the data is transferred in the form of voltage. As the hardware restricts the voltage level, the entire analog data should be in the limits. So, the analog data is scaled to meet the requirements. Detailed data handling in eMEGAsim is illustrated in Figure 4.5.

In case of data transfer between database, visual interface, and data acquisition, the transfer is through ethernet connection. All the systems are equipped with 100Mbps network interface cards. So, this provides enough bandwidth for data transfer between the systems.

4.4 Real-time System Model

The eMEGAsim enables parallel execution of models. To leverage the advantage provided by the eMEGAsim, the entire system model will be divided into several subsystems. In this platform, real-time system model is divided into three groups:

- system model,
- substation model,
- data handling subsystem.

These groups are interconnected to each other. The layout of the subsystem groups is shown in Figure 4.4.

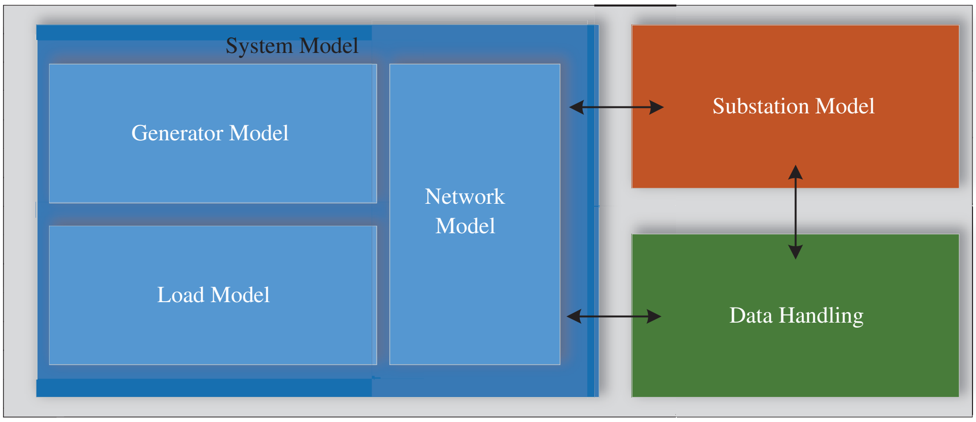


Figure 4.4: Subsystems of real-time model

4.4.1 System Model

System model contains core components of power system model. Core components such as generator, load and transmission line models are considered as a part of the system model. These models do not change much at component level, when exposed to changes in power system model. However, system-wide changes are inevitable, in case of different test cases. This group further divided into several subsystems to achieve minimum possible step size, which will give more accurate simulation of the power system.

4.4.2 Substation Model

This group mainly consists measurement, bus-bar, and protection equipment models. Substation model handles all the protection related tasks. Any application related to protection can be added in this substation model. It needs to be divided into several subsystem, if the complexity of this model increases. This model changes with the configuration of the system.

4.4.3 Data Handling Subsystem

This subsystem handles data formatting and the interface with external devices. Structure of data handling subsystem remains same, irrespective of the targeted power system model. However, the variable i.e, data size and scaling will change according to requirements.

Tasks of the data handling system are:

- accessing input and output ports,
- multiplexing and demultiplexing,
- data classification, and scaling.

Different stages in data handling are shown in Figure 4.5. In order to implement this subsystem, user should know the information about targeted system. Information such as:

- number of available analog and digital channels,
- minimum and maximum limits of the power system model output,
- number of measurements or data points.

Accessing input and output ports

RT-LAB software provided with a library to access the digital and analog ports installed in eMEGAsim. This task is responsible for reading and writing data to hardware ports. The task is divided into four subsystems. The four subsystems are designed to manage analog input, analog output, digital input, and digital output respectively. In these subsystems, number of available ports are defined as variable. This task is represented as channel selection in Figure 4.5.

Multiplexing and demultiplexing

This task is implemented with two subsystems by using the concept mentioned in section 4.3. Both subsystems are supplied with a common counter to supply select lines for multiplexing and demultiplexing. Hardware capability and model complexity limits the counter speed. Using separate selection lines for analog and digital data is recommended in this platform. However, a common selection line i.e. either digital or analog selection lines can be used for this purpose.

In brief, the concept mentioned in section 4.3 for multiplexing and demultiplexing is:

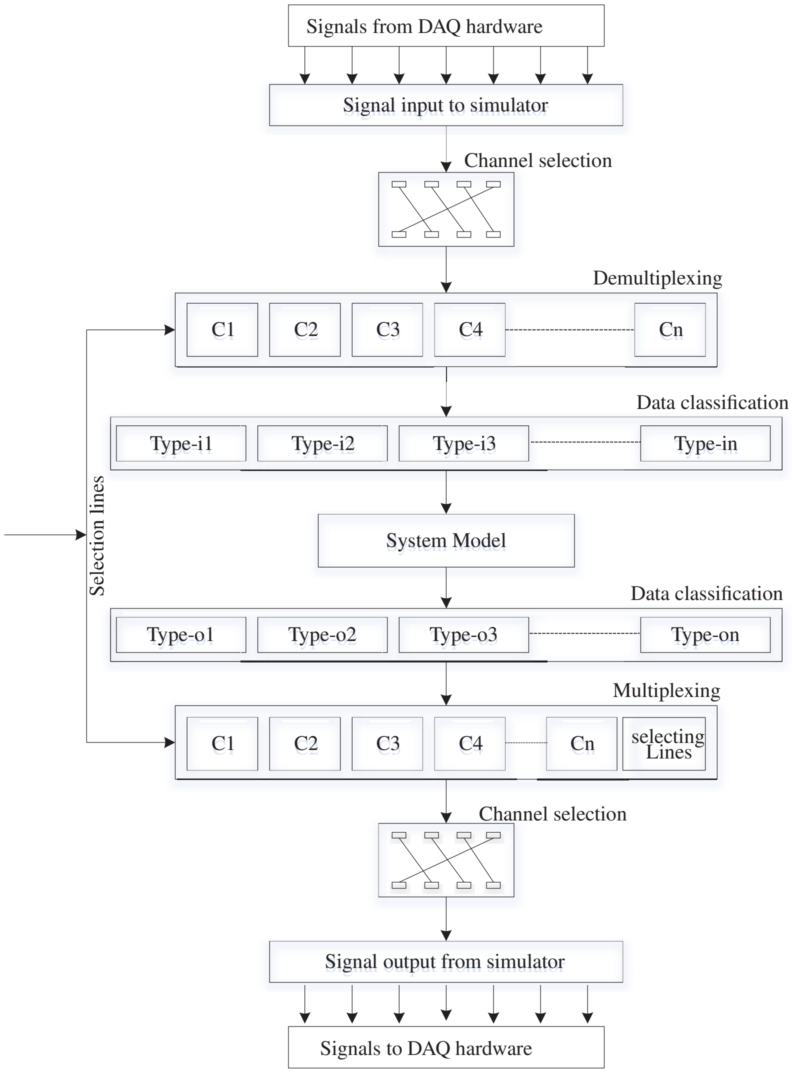


Figure 4.5: Internal data handling

- for multiplexing, arrange data in two dimensional matrix,
- sending each row of the matrix along with the select line according with the counter speed,
- receive each row from eMEGAsim input ports for demultiplexing purpose.

As shown in Figure 4.5, C1, C2, to Cn are the data columns. The multiplexing has a selection line column, whereas the data received to demultiplex does not contain select line. This is deliberately introduced into structure to maintain time synchronization between external hardware and eMEGAsim. As mentioned earlier, the common counter serves as the selection line for both the multiplexing and demultiplexing subsystems.

Data classification and scaling

Data classification is required indeed, to properly identify and organize the data generated by the power system model. This data can be classified based on type of device, type of measurement. In this platform, there are two type of data, digital and analog. Digital data consist circuit breaker status, equipment status and so on. Voltages, currents and etc. are treated as analog data.

In case of bus based classification, all measurements related to a particular bus is declared as a group. It is not mandatory, that all the buses should contain same data length. As mentioned earlier, to match connection requirements the data need to be scaled. At this stage, all groups are scaled with corresponding scaling factors. The scaling factors are chosen based on expected maximum values of that particular case. Generally, each measurement type will have a common scaling factor, i.e. for all the voltage measurements a common scaling factor is used.

Measurement based classification gives flexibility in scaling. In this case, all voltages, currents, and angels are grouped individually. In grouping process, proper sequence should be maintained. Otherwise identification of data corresponding to particular device is difficult in further stages. The data size may not be equal for all groups. As there is only one scaling factor for each group, the length of the group is irrelevant.

As shown in Figure 4.5, Type-o1 to Type-on are output data groups of power system model. The scaled and serialized groups are passed to multiplexing subsystem. Data prepared by demultiplexing subsystem is re arranged into Type-i1 to Type-in groups and scaled as required by the power system model. This classification method and scaling factors must be made available to the other systems to avoid misinterpretation of the data.

System model, substation model, and data handling subsystems are developed in MATLAB environment and deployed in eMEGAsim. This format can be adapted to any size of system that supported by the hardware. In this structure, substation model is left open for experimentation.

4.5 Data Acquisition Application

Block diagram of data acquisition application is shown in Figure 4.6. This application is developed entirely in LabVIEW. Two separate programs are developed as a part of this application. One program handles data exchange with database and the program is named as data logging software in Figure 4.1. Whereas, other handles data exchange with eMEGAsim, named as NI-PXI application.

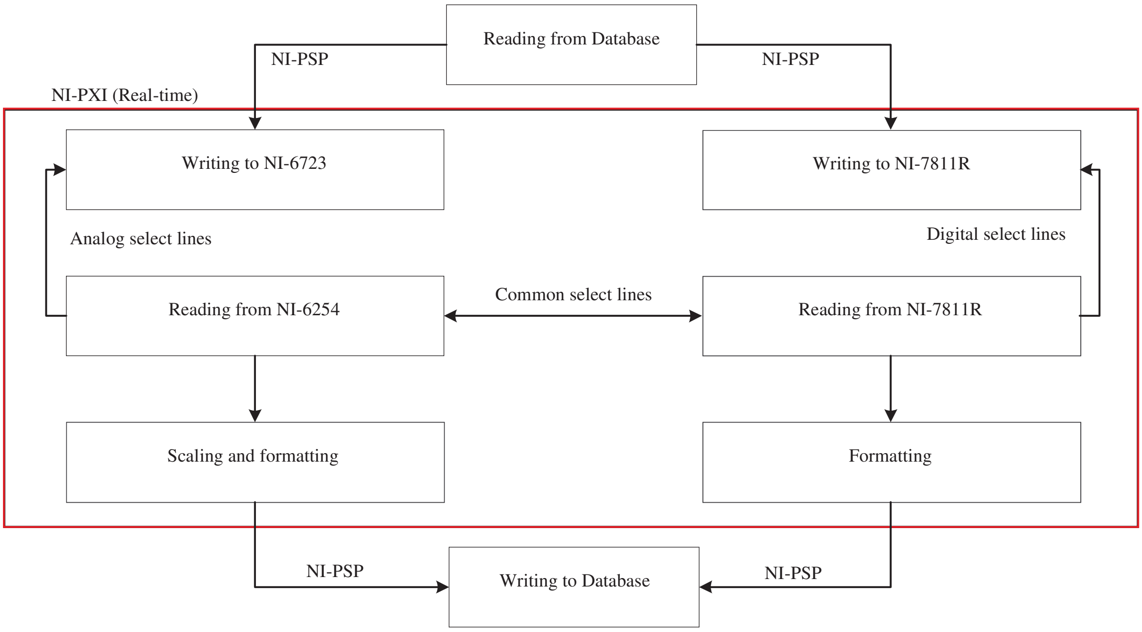


Figure 4.6: Block diagram of data acquisition application

4.5.1 Data logging software

Main task of the data logging software is to perform read and write actions on database. It exchanges data with NI-PXI through ethernet using NI-PSP. This program exchanges data with database using open database connectivity (ODBC) driver. LabVIEW provides required packages to perform these actions. To implement this program, partial information about the database structure is required.

In this program, four independent sub-programs are designed to perform read and write operations. These sub-programs designed to run in parallel. Each loop is dedicated to a task as follows:

- program-1: read analog data from NI-PXI and write to database,
- program-2: read analog data from database and write to NI-PXI,
- program-3: read digital data from NI-PXI and write to database,
- program-4: read digital data from database and write to NI-PXI.

Accessing data from database involves executing SQL queries. To execute these queries more time is required as compared to the model execution in NI-PXI. So, it cannot run in compliance with the eMEGAsim or NI-PXI. As the reporting rate of the data largely depends on this data logging software performance, care shall be taken while executing this application. Sufficient buffer shall be used in between NI-PXI and data logging software to avoid data loss. In this platform, first the data will be stored in database and then other applications uses data from database. If required, this program can be modified to exchange data with other applications without storing the data into database.

4.5.2 NI-PXI application

NI-PXI application is developed to use NI hardware for data acquisition application in real-time. Besides, it can be used to emulate communication system. This application performs following tasks:

- exchanges data with eMEGAsim,
- identifies select lines,
- applies scaling,
- rearranges data according to select lines,
- exchanges data with data logging software.

Block diagram of this application is highlighted as NI-PXI (real-time) in Figure 4.6. This application is developed in LabVIEW. Six programs are integrated together build this application.

Reading data from NI-6254

NI-6254 is used as an analog input hardware to NI-PXI. This program is used to configure the hardware, acquire analog data from eMEGAsim, and to identify the select lines from acquired data. These select lines are named as analog select lines. LabVIEW provides required library to access the data from hardware. However, user should specify configuration parameters such as sampling rate, ports index, and so on. To identify select lines from the data, the ports corresponding to these lines should be provided to this program.

Reading data from NI-7811R

NI-7811R is used as digital input hardware. The task of this program is same as the above program. However, the hardware capabilities differs from each other. So, the configuration parameters changes from NI-6254.

If separate select lines are used, then both programs are independent from each other. As mentioned earlier, a set of common select lines can be used for both analog and digital multiplexing. These select lines are transferred through either analog or digital ports. So, the identification of data received by one program depends on the other program. In this case, coordination of both the programs is required for identification data based on select lines. Being digital hardware, NI-7811R is much faster than the NI-6254 and NI-6723. Coordination between these programs can be implemented by slowing the execution of one program. However, common select lines are beneficial when size of analog and digital data is same. In most of the cases, number of select lines required for digital is less when compared with analog. For faster multiplexing, digital select lines are the suggested to use as common select lines.

Scaling and formatting

This program applies scaling on data to retrieve original values generated by eMEGAsim. The received data is rearranged into two dimensional array as per the select lines. Scaling program applies scaling on this rearranged data. In next step, the scaled data will be published to a network variable. The network variables are LabVIEW process variables published over a network for common data exchange. This network variable provides required data buffer as per the user configuration. To implement this, information about scaling factor is required.

Writing data to NI-6723 and NI-7811R

NI-6723 is used as analog output hardware to NI-PXI. Some of the ports in NI-7811R are used as digital output hardware. Both of these programs takes data from data logging software and select lines. Based on select lines this program transfers corresponding data through the output ports. If any particular case requires scaling or formatting the data, another program can be implemented as mentioned above.

The data sent by data logging software is a two dimensional array. Based on the select line sent by the eMEGAsim, the corresponding row will be transmitted through the ports. As mentioned earlier, a limited counter serves as select lines. This data acquisition program process the data and send the corresponding input to eMEGAsim before the counter changes its count. This ensures the synchronization between eMEGAsim and data acquisition system. As it is dealing with data transfer, communication related WAMS applications can be implemented in this section.

4.6 Database

In this platform, MySQL database server is used for data sharing and archiving. Database is accessed by the visual application and data logging software (LabVIEW). Any changes in database structure demands modifications in both applications. Tables in database are divided into two groups. One group of tables contains data corresponding to the data logging software, whereas other group contains data related to visual application. As a result, change in any group requires change in corresponding application.

First group of database consist four tables for analog input, analog output, digital input, and digital output. These tables will be in interaction with the eMEGAsim via data logging application. Second group is further divided into two subgroups, one group for received data and another group is to update user actions from visual interface. Number of tables in each subgroup depends on the features incorporated in the platform. For illustration, total six tables are incorporated into structure. These six tables are classified as bus data, line data, control actions (automated), relay status, relay status (user actions), and set-points (user actions).

Bus data table provides bus based information, such as bus voltage, angle, and frequency. Line data table gives current and current angle. Control actions table provides the information of automatic controller in the system. This is applicable only when any controller is implemented in either eMEGAsim or NI-PXI. Relay status table gives status of circuit breakers, relays, and equipment. These four tables correspond to the output data of eMEGAsim and NI-PXI. Remaining two tables give data to NI-PXI through data logging software by capturing corresponding user actions from visual interface. Relations between different tables are shown in Figure 4.7 and 4.8.

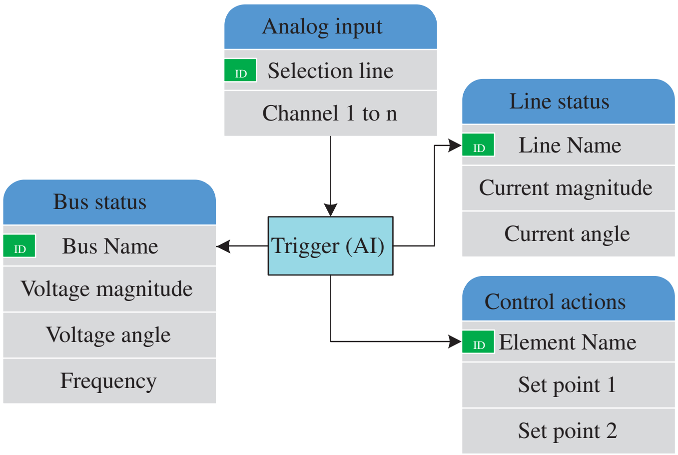


Figure 4.7: Relation between tables for analog inputs

Database triggers are used for data transfer to the tables. Trigger is an automated task programmed inside the database. This triggers are attached to the source table. In this context, analog input, digital input, relay status(UA), and set-point(UA) will be source files. These triggers are programmed to execute automatically when any update occurs in source table. These triggers are mainly used to map the data between NI-PXI and visual interface. In case of analog input and digital input tables, contain two dimensional array provided by the NI-PXI. Triggers takes formatted data from source table and rearranges

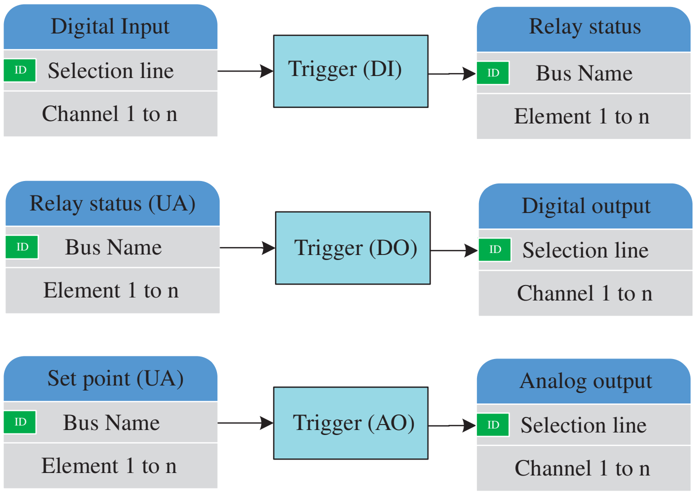


Figure 4.8: Relation between tables for digital, analog outputs, and digital inputs

into a user friendly format. Whereas triggers of other source tables work in reverse to that of above i.e. taking user data and rearranging into NI-PXI format.

4.7 Data Preparation and System Integration

Data preparation is the first task in implementing any test case. Data preparation module provides information required by the model building. The information such as bus data, line data, available ports, and so on are inputs to the data preparation module. The information requirement of different systems are listed in Table 4.4. System integration involves defining scaling factors, port mapping, and programming triggers. Based on physical connection between eMEGAsim and NI-PXI, the software modules are configured to exchange the data through designated ports. The database triggers are programmed to interface the data logging application and graphical user interface.

4.8 Visual Interface

In-house visual interface development is carried out as a part of this platform. This visual interface is intended to incorporate user action as a component in the platform. Besides, it serves as visual aid to user. Python is used for development of this interface. It consists two windows named as design window and run window.

Table 4.4: Data requirements of different systems.

System	Subsystem	Information requirement
Real-time model	System model Substation model Data handling	Power system data. Measurement requirements. Number of available ports, scaling factors, number of select lines, and type of data classification.
Data Acquisition	NI-PXI application	Number of available ports, scaling factors, data classification information, and select lines information.
	Data logging application	Database details, NI-PXI connection information.
Database		Power system data, data classification information, visual interface information, and user data requirement.
Visual interface		Power system data, database information.

4.8.1 Design Window

The design window is used to draw single line diagram of the power system. Basic components in power system such as generator, load, compensator, bus, circuit breaker, and transmission lines are implemented in this design window. The components are represented by basic shapes such as circle, rectangle, and etc. However, the program structure allows easy integration of the other components.

The design window is programmed as event based structure. These events are tagged to pointing device (mouse events). The major events in this design window are insert substation (ISS) and insert line (IL). ISS initiates a pop-up window to collect details, such as bus name, area name, number of generators, number of loads, number of compensators, and number lines terminating at that bus. With these details, the ISS event inserts the components. To insert lines between these components, IL event initiates a pop-up window for line name. With the name provided, IL inserts lines between specified components. Screen capture of design window with 39 bus New England test system single line diagram is shown in Figure 4.9.

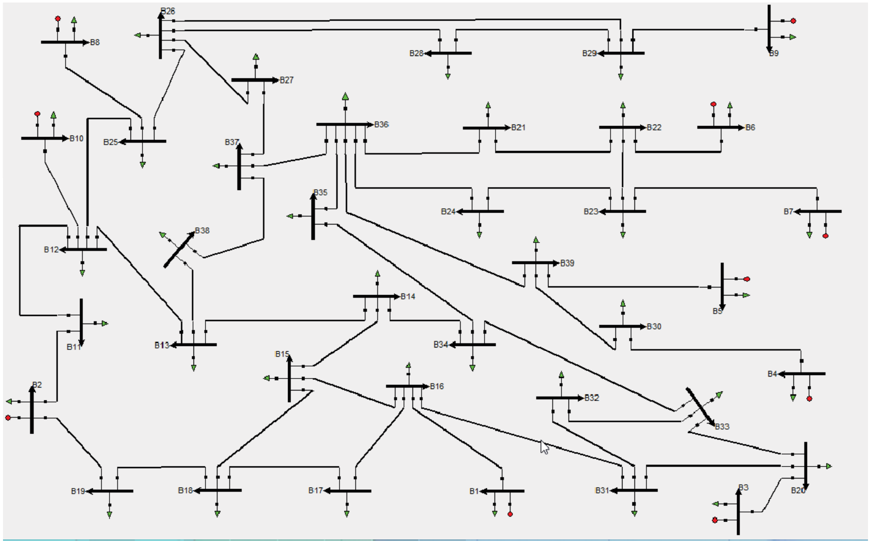


Figure 4.9: Design Window with 39 bus New England test system

4.8.2 Run Window

In control centers, operator's instincts and knowledge on system play a vital role in operating the power system. So, presenting relevant data to user is also a factor in testing WAMS applications. Run window provides this feature. The run window consists data visualization module and user actions module. This window continuously receives data from database and presents it to user. In addition to that, it captures user actions and forward them to database.

Visualization module

Visualization modules are meant to read data from database and present it to the operator. Visualization based WAMS applications can be included in this module. Data visualization mainly involves presenting the data in the form tables, graphs, and colors. In this platform, bus based filled contour and element wise color representation are implemented till the date. In case of color representation, maximum and minimum values for the color map are required. This values depends on the system constraints and application. So the values can be specifies by the user in run-time. Visualization can be used to present voltage magnitude, angles, and frequency. A sample data is presented in the form of contour in Figure 4.10.

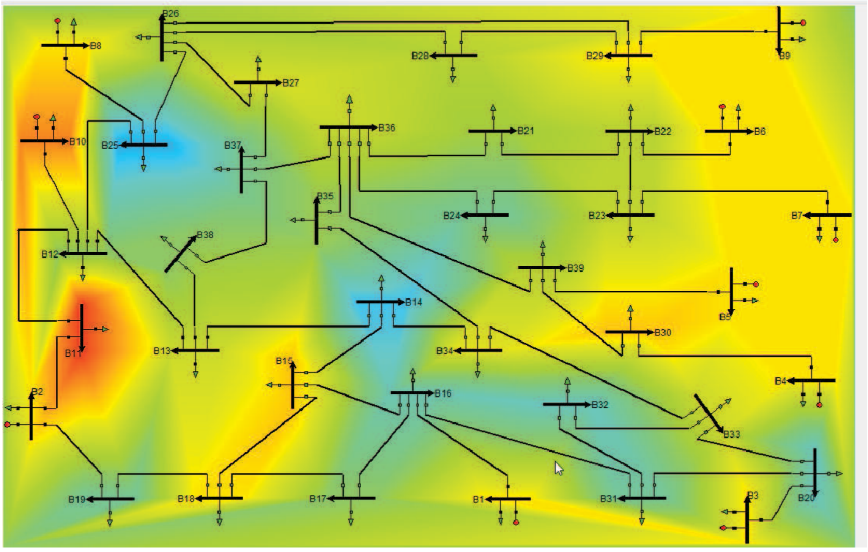


Figure 4.10: Run Window of 39 bus New England test system with contour plot.

User actions module

One of the main objective of this platform is to incorporate user actions as a part of the system. As a user application, WAMS will also suggest user actions. Validation of the WAMS application is incomplete without testing the suggested user actions with the power system in real-time. User actions such as operating circuit breakers, changing set-points, and etc. are implemented in this platform. Operating circuit breakers involves:

- connecting or disconnecting a single component or line,
- isolating a bus from other components and lines,
- islanding an area from the system.

Changing set-points consist changing generator set-points, loads, and compensator set-points. All of these are considered as external triggers to the power system model running in eMEGAsim.

These are the basic features incorporated into the visual interface. As this interface developed as object oriented program, it is easy to integrate the targeted algorithm or application for testing.

4.9 Laboratory Set-up

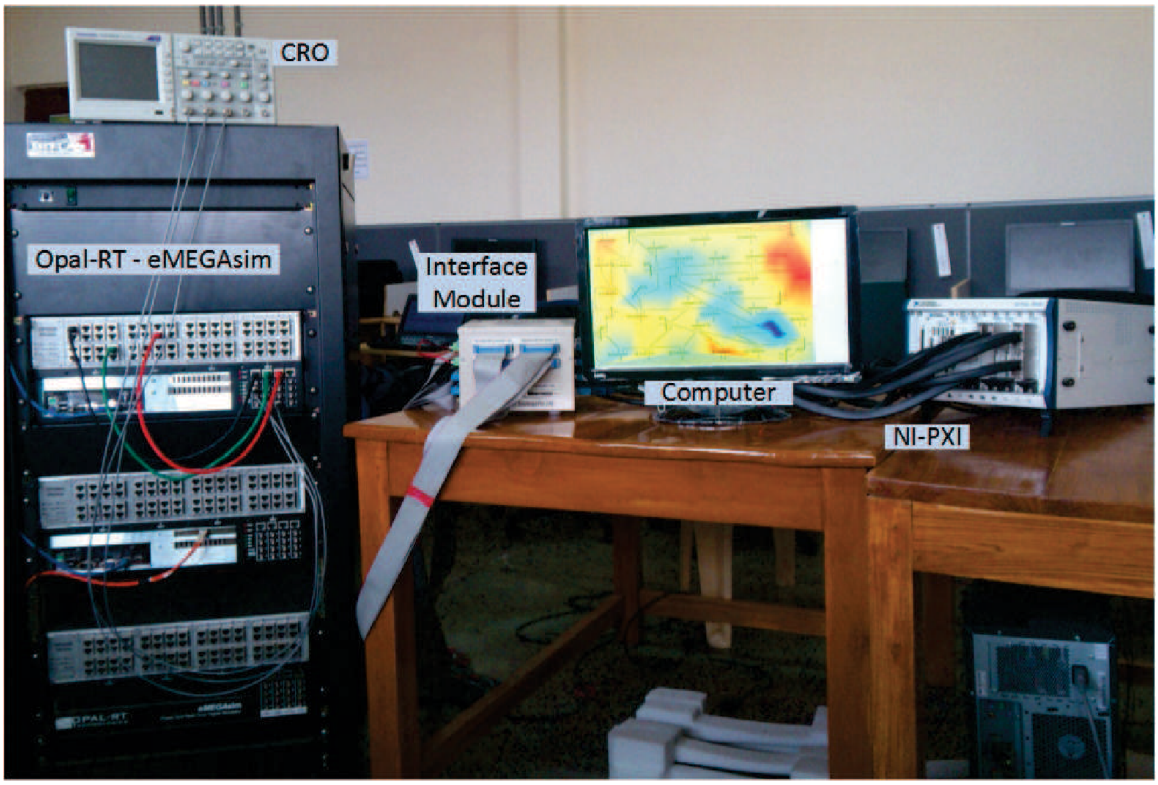


Figure 4.11: Laboratory set-up

The laboratory implementation is shown in 4.11. The eMEGAsim is connected to interface module through a meter long, 37 pin flat ribbon cables (FRCs). In addition, ethernet connection is used for monitoring eMEGAsim by using RT-LAB software. However, the the front interface connectors of eMEGAsim enables the signal monitoring of ports through digital oscilloscope. The interface module is connected to NI-PXI by using 68 pin shielded cables. Besides, the NI-PXI connected to computer through ethernet switch for application deployment and monitoring.

4.9.1 Data Transfer Check

A test case is constructed to demonstrate the data transfer from eMEGAsim to database and vice-versa. From eMEGAsim numeric values are sent through first ten analog ports. NI-PXI application read the values from eMEGAsim. Data logging application stores the received data from NI-PXI to database. In the other path, NI-PXI application retrieves

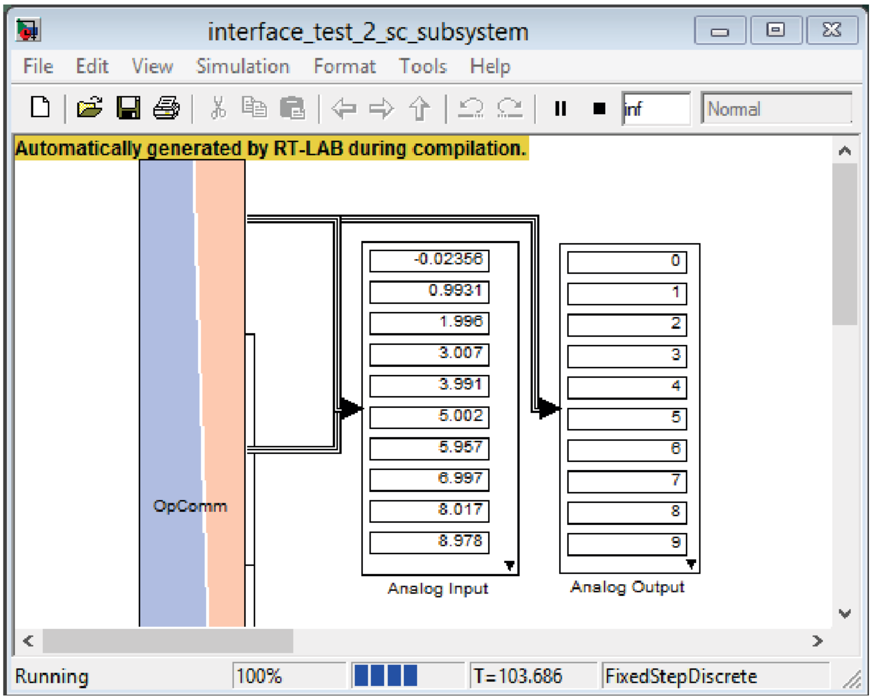


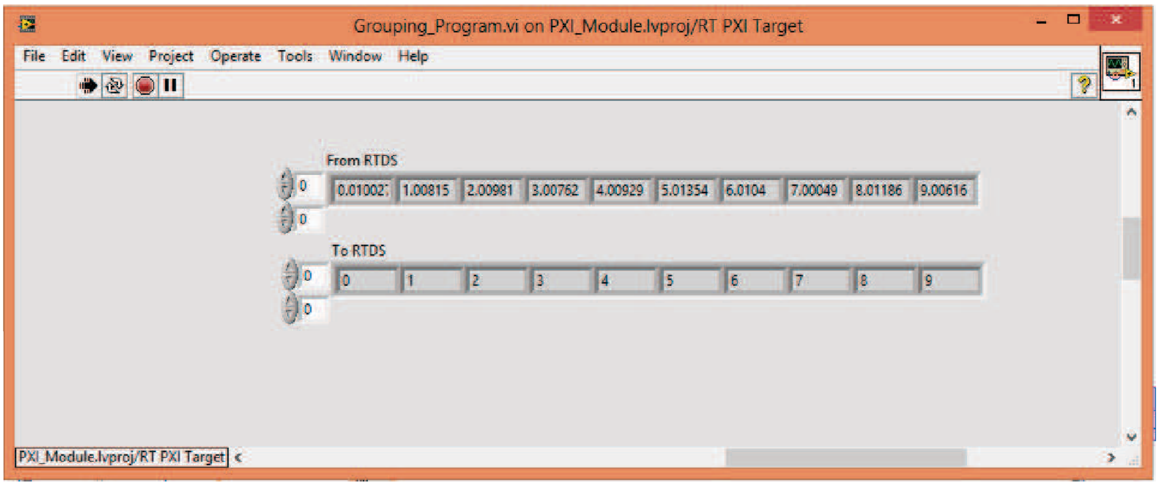
Figure 4.12: Data display in eMEGAsim console

numeric values from database and send it the eMEGAsim input ports.

As a part of this exercise, numeric values of 0 to 9 are exchanged from eMEGAsim to database and database to eMEGAsim. Monitoring screens of eMEGAsim, NI-PXI and database are shown in Figure 4.12, 4.13, and 4.14 respectively. eMEGAsim console is displaying analog signal at input and output ports. The eMEGAsim console is basically a MATLAB model deployed in eMEGAsim by using RT-LAB. NI-PXI application screen is front-end of LabVIEW program. A data explorer program HeidiSQL is used to explore the database. These screens are monitored from the central computer, which is connected to ethernet to manage these systems.

4.9.2 Multiplexing and Demultiplexing Demonstration

To demonstrate the multiplexing and demultiplexing, few test cases are constructed. It is hardware in the loop test. NI-PXI is connected in loop with the eMEGAsim. In this test cases, The select lines are generated by eMEGAsim and the same are sent as data from the eMEGAsim. NI-PXI application identifies the received data and sent back the same data to eMEGAsim. The test is repeated for different speeds of the select lines. An the signals are



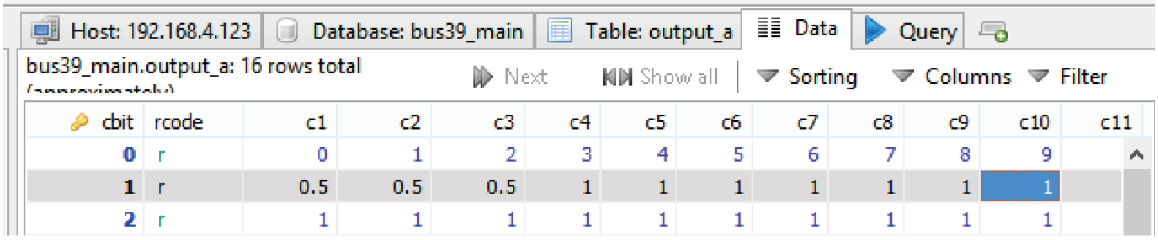
From RTDS

0	0.01002	1.00815	2.00981	3.00762	4.00929	5.01354	6.0104	7.00049	8.01186	9.00616
0										

To RTDS

0	0	1	2	3	4	5	6	7	8	9
0										

Figure 4.13: Data display in NI-PXI application



bus39_main.output_a: 16 rows total

cbit	rcode	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11
0	r	0	1	2	3	4	5	6	7	8	9	
1	r	0.5	0.5	0.5	1	1	1	1	1	1	1	
2	r	1	1	1	1	1	1	1	1	1	1	

Figure 4.14: Data display in database explorer

monitored through digital oscilloscope at eMEGAsim input-output ports.

A test case is with digital ports and 50μs for each step in select line is executed. Figure 4.15 shows the signals from two output ports and a input port. channel 1 (CH1) in the figure represents the select line, channel 2 (CH2) represents the data sent to NI-PXI, and the remaining signal is the data received by eMEGAsim. In this test, digital ports are used so, the response is much faster. Similar test is conducted with the 200μs for each step in select line. and the results are shown in Figure 4.16.

With 10ms for each step, a test case is constructed. In this case, analog ports are used. The range of the select line counter is chosen as 0 to 15. The same select line data sent through the output ports of the eMEGAsim as data. However, the hardware of this platform supports upto 10V. So, a scaling factor of 0.5 is applied to the data and select line. The analog select line with range of 0 to 15, is equal to four select lines in digital. Figure 4.17 shows screen capture of digital oscilloscope connected to eMEGAsim ports. CH1 represents the select line signal, CH2 represents data sent to NI-PXI, and CH3 represents

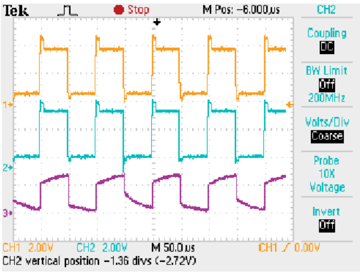


Figure 4.15: Digital signals at eMEGAsim with 50μs loop back through NI-PXI

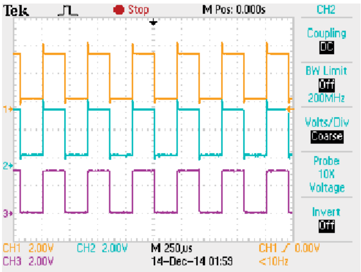


Figure 4.16: Digital signals at eMEGAsim with 200μs loop back through NI-PXI

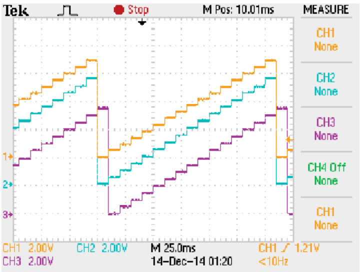


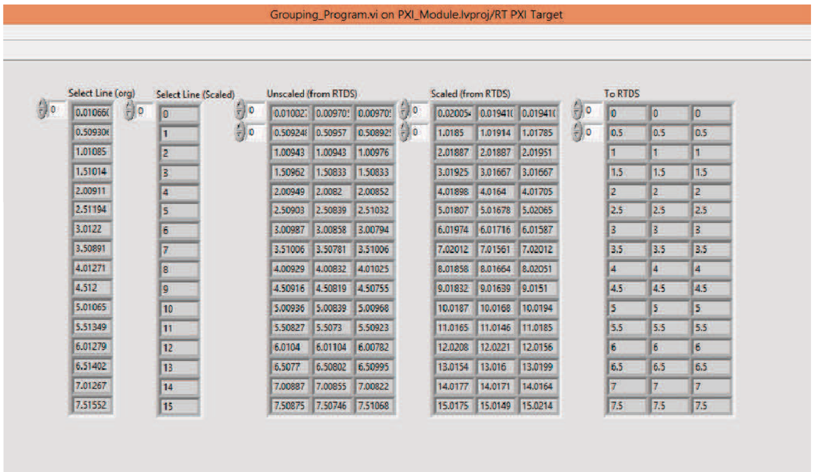
Figure 4.17: Analog signals at eMEGAsim with 10ms loop back through NI-PXI

data received by eMEGAsim. Here, A step delay is introduced by the NI-PXI application, due to the slower response of the analog hardware. It can be observed that every value is coincide with the select line as expected. Figure 4.18 shows screen capture of the NI-PXI application. It is showing original and scaled values received from eMEGAsim, select line values, and data sent to eMEGAsim after processing. The processing of select lines involves scaling and rounding operations. This is indeed required to nullify the errors added in the data transfer through analog ports. However, the rounding operation is not applicable to the data.

Figure 4.19 shows the digital equivalent of the multiplexing with select line range is from 0 to 15. As mentioned earlier, it required four bits to represent the maximum numeric of select line. So, four select lines are used to transfer the index. The same four bits are used as data from eMEGAsim. The NI-PXI application must identify the data according to the select lines. The Figure 4.19 shows the bits received from eMEGAsim and displays the data and select lines. This demonstrates the usage of multiple select lines.

4.9.3 Execution of 39 Bus New England Test System

To demonstrate the power system model visualization, 39 bus New England test system is constructed [22]. Detailed modeling and data of the system are presented in Appendix A and B respectively. The power system is modeled in MATLAB. This model consists 39 buses, 10 generators, and 46 lines. This gives 141 digital data and 405 analog data from eMEGAsim to NI-PXI, which includes circuit breaker status, voltages, currents, and set-points. Besides, 141 digital data and 98 analog data from NI-PXI to eMEGAsim. In this


Grouping_Program.vi on PXI_Module.lvproj/RT PXI Target

Select Line (org)	Select Line (Scaled)	Unscaled (from RTDS)			Scaled (from RTDS)			To RTDS		
0.01066	0	0.01002	0.00970	0.00970	0.02005	0.01941	0.01941	0	0	0
0.50930	1	0.50924	0.50957	0.50892	1.0185	1.01914	1.01785	0.5	0.5	0.5
1.01085	2	1.00943	1.00943	1.00976	2.01887	2.01887	2.01951	1	1	1
1.51014	3	1.50962	1.50833	1.50833	3.01925	3.01667	3.01667	1.5	1.5	1.5
2.00911	4	2.00949	2.0082	2.00852	4.01898	4.0164	4.01705	2	2	2
2.51194	5	2.50903	2.50839	2.51032	5.01807	5.01678	5.02065	2.5	2.5	2.5
3.0122	6	3.00987	3.00858	3.00794	6.01974	6.01716	6.01587	3	3	3
3.50891	7	3.51006	3.50781	3.51006	7.02012	7.01561	7.02012	3.5	3.5	3.5
4.01271	8	4.00929	4.00832	4.01025	8.01858	8.01664	8.02051	4	4	4
4.512	9	4.50916	4.50819	4.50755	9.01832	9.01639	9.0151	4.5	4.5	4.5
5.01065	10	5.00936	5.00839	5.00968	10.0187	10.0168	10.0194	5	5	5
5.51349	11	5.50827	5.5073	5.50923	11.0165	11.0146	11.0185	5.5	5.5	5.5
6.01279	12	6.0104	6.01104	6.00782	12.0208	12.0221	12.0156	6	6	6
6.51402	13	6.5077	6.50802	6.50995	13.0154	13.016	13.0199	6.5	6.5	6.5
7.01267	14	7.00887	7.00855	7.00822	14.0177	14.0171	14.0164	7	7	7
7.51552	15	7.50875	7.50746	7.51068	15.0175	15.0149	15.0214	7.5	7.5	7.5

Figure 4.18: Analog data multiplexing and demultiplexing display in NI-PXI application

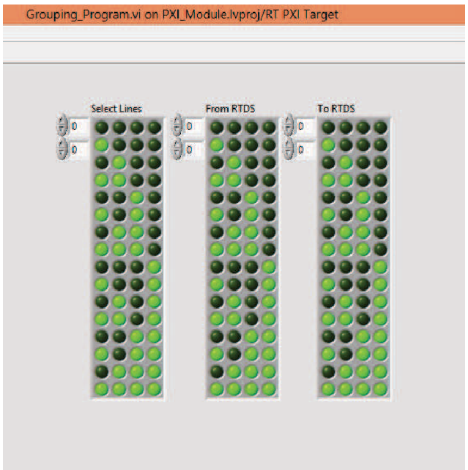

Grouping_Program.vi on PXI_Module.lvproj/RT PXI Target

Figure 4.19: Digital data multiplexing and demultiplexing display in NI-PXI application

implementation, separate select lines are used for digital and analog. For data transfer, 31 ports for analog output, 62 ports for digital output, 31 ports for analog input, and 64 ports for digital input are used for data exchange between eMEGAsim and NI-PXI. With these ports, two select lines are required for digital data and one select line with 0 to 14 range is required for multiplexing.

The data has to be arranged into a two dimensional array for multiplexing. Dummy values are appended at the end of the data to arrange the data into a two dimensional matrix. The dummy values are filtered out in later stages. The data is classified based on measurement. As a part of power system model initialization, load flow analysis is carried out. From this analysis scaling factors are derived to meet the connection requirements.

The NI-PXI application is configured with the help of the above data.

The visual model of the 39 bus New England test system is constructed using the design window. A database is created along with the visual model based on terminology used in visual model. Terminology includes bus names, line names, and system elements. From the other end, four tables are created for data logging software. The data logging software is provided with the connection details of the database.

To start the power system model from steady state, proper input values must be provided to the eMEGAsim. So, database output tables shall be populated with the default values. The start sequence of the platform is as follows: Data logging software, NI-PXI application, and eMEGAsim. The visual interface can be initiated as per the requirement.

In this test case, Voltage magnitude and angles are displayed in the form of contour plot. Two cases are executed to demonstrate the advantage of visual interface. The first case is with the normal operating conditions and the second case is with the drop in load at 13^{th} bus. The minimum and maximum limits for contour plot are chosen as 0.92 and 1.08 for voltage magnitude and -0.15 and 0.35 for voltage angles. Figure 4.20 and 4.21 shows the contour plots of voltage magnitude and voltage angle respectively under normal operation. Figure 4.22 and 4.23 shows the voltage magnitude and voltage angle under load change event. This demonstrates the use of the visual interface in determining the system state.

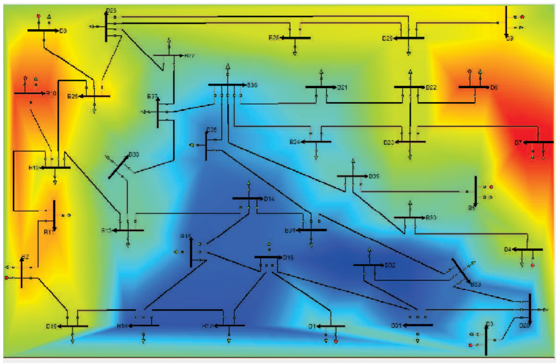


Figure 4.20: Voltage magnitude contour plot

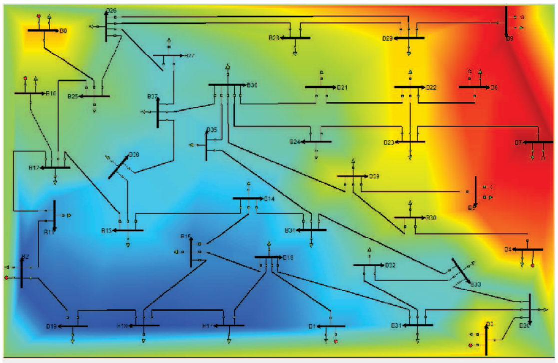


Figure 4.21: Voltage angle contour plot

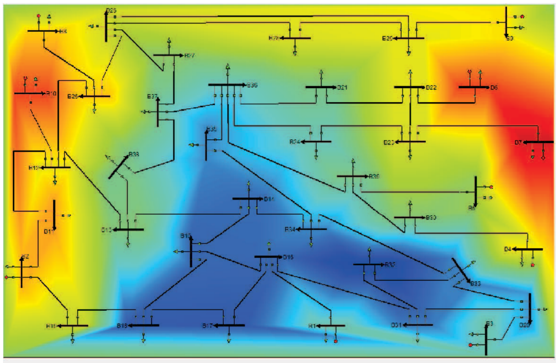


Figure 4.22: Voltage magnitude contour plot under load change

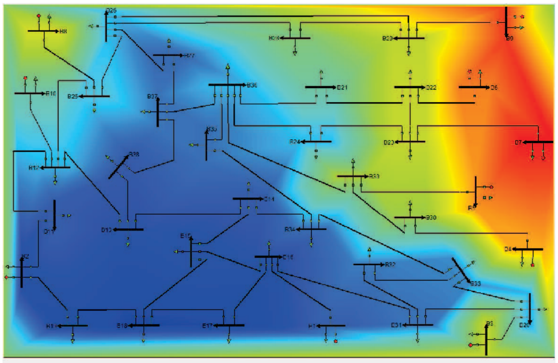


Figure 4.23: Voltage angle contour plot under load change

4.10 Summary

Power system model is divided into three groups to minimize the changes from one test case to another. However, these groups further divided into subsystems to achieve minimum step size with eMEGAsim. NI-PXI application and data logging software are developed for data acquisition. NI-PXI application provides a facility to implement communication related programs. A hardware interface module is used in between eMEGAsim and NI-PXI to meet physical connection requirements. A database scheme is developed to minimize the complexity involved in integrating data acquisition and visual interface. A visual interface is developed to incorporate user actions in power system. Visual interface provides facility to incorporate WAMS applications for testing purpose. Combination of these systems evolve into a platform for WAMS application development and testing. Demonstration of the platform is carried out with a 39 bus New England test system. Besides, data transfer and multiplexing is also demonstrated with the help the platform.

Chapter 5

Conclusions

In this thesis, a platform is developed for design and testing of WAMS applications. The platform is divided into three segments. The three segments represent three different domains such as, power system, communication system, and visual interface. These are combined together to build a platform that emulates the power system along with the other components. A work flow is presented for using the platform for development. A general structure of the platform is also suggested.

The framework is implemented with relevant software and hardware tools. For power system domain, eMEGAsim is used as a real-time simulator for the power system and other components. The NI-PXI hardware and LabVIEW software modules are used for data acquisition. A visual interface is developed using Python for the platform proposed. MySQL is used as database server.

Integration of these components are discussed in this thesis. The compatibility issues with eMEGAsim and NI-PXI is resolved with an interface module. To interface the real-time hardware and other software systems, a data logging software is developed in LabVIEW. The data logging software acts as a bridge between NI-PXI application (real-time) and database. In database, two layers are developed to minimize the interaction between visual model and NI-PXI application. Database triggers are used for interaction between these two layers.

A graphical user interface is developed to represent the power system with a graphical model. User actions module is developed to make the power system responsive to the user actions received through the user interface.

The development of the platform is carried out without loosing the generality. This platform consists open points in all the segments, which allows addition of modules to enhance the capabilities of the platform. eMEGAsim accepts the controllers to implement at power system level. Communication related modules can be incorporate in the platform, by adding the modules in NI-PXI application. Visual or user interaction based applications can be integrated in the visual interface.

Appendix A

System Model

All variables in synchronous machine and network are transformed into a reference frame to convert variables in constant values. Dynamic model of synchronous machine is considered from [22].

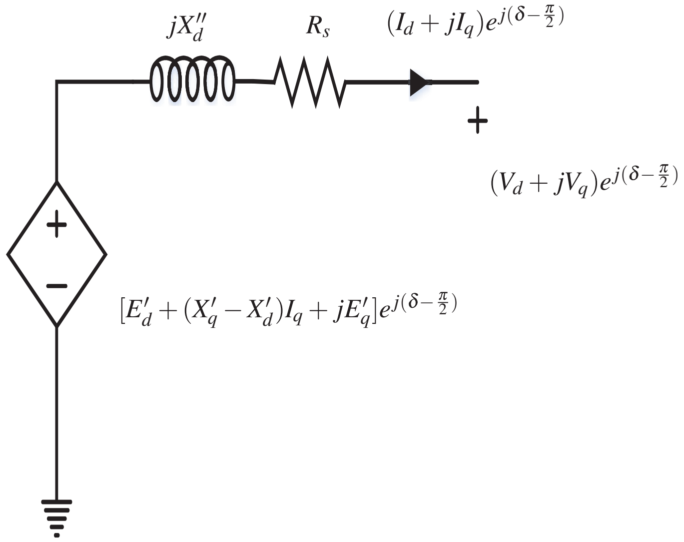


Figure A.1: Equivalent circuit of synchronous machine two axis model

Dynamics of damper windings are neglected to derive a reduced order model. The equivalent circuit of the reduced order model is shown in Figure A.1. the synchronous machine model is expressed in equations from (A.1) to (A.4). The exciter and speed governor models are represented in equations from (A.5) to (A.9). The equations represents a multi machine model for m machines. Expressions from (A.10) to (A.12) provide the constraints to interface the machine to the network.

$$T'_{doi}\frac{dE'_{qi}}{dt} = -E'_{qi} - (X_{di} - X'_{di})I_{di} + E_{fdi}, \qquad i = 1,...,m \tag{A.1}$$

$$T'_{qoi}\frac{dE'_{di}}{dt} = -E'_{di} + (X_{qi} - X'_{qi})I_{qi}, \qquad i = 1,...,m \tag{A.2}$$

$$\frac{d\delta_i}{dt} = \omega_i - \omega_s, \qquad i = 1,...,m \tag{A.3}$$

$$\frac{2H_i}{\omega_s}\frac{d\omega_i}{dt} = T_{Mi} - E'_{di}I_{di} - E'_{qi}I_{qi} - (X'_{qi} - X'_{di})I_{di}I_{qi} - T_{FWi}, \qquad i = 1,...,m \tag{A.4}$$

$$T_{Ei}\frac{dE_{fdi}}{dt} = -(K_{Ei} + S_{Ei}(E_{fdi}))E_{fdi} + V_{Ri}, \qquad i = 1,...,m \tag{A.5}$$

$$T_{Fi}\frac{dR_{fi}}{dt} = -R_{fi} + \frac{K_{Fi}}{T_{Fi}}E_{fdi}, \qquad i = 1,...,m \tag{A.6}$$

$$T_{Ai}\frac{dV_{Ri}}{dt} = -V_{Ri} + K_{Ai}R_{fi} - \frac{K_{Ai}K_{Fi}}{T_{Fi}}E_{fdi} + K_{Ai}(V_{refi} - V_i), \qquad i = 1,...,m \tag{A.7}$$

$$T_{CHi}\frac{dT_{Mi}}{dt} = -T_{Mi} + P_{SVi}, \qquad i = 1,...,m \tag{A.8}$$

$$T_{SVi}\frac{dP_{SVi}}{dt} = -P_{SVi} + P_{Ci} - \frac{1}{R_{Di}}\left(\frac{\omega_i}{\omega_s} - 1\right), \qquad i = 1,...,m. \tag{A.9}$$

$$0 = V_i e^{j\theta_i} + (R_{si} + jX'_{di})(I_{di} + jI_{qi})e^{j(\delta_i - \frac{\pi}{2})} - [E'_{di} + (X'_{qi} - X'_{di})I_{qi} + jE'_{qi}]e^{\delta_i - \frac{\pi}{2}}, \quad i = 1,...,m. \tag{A.10}$$

$$V_i e^{j\theta_i}(I_{di} - jI_{qi})e^{-j(\delta_i - \frac{\pi}{2})} + P_{Li} + jQ_{Li} = \sum_{k=1}^{n} V_i V_k V_{ik} e^{j(\theta_i - \theta_k - \alpha_{ik})}, \quad i = 1,...,m \tag{A.11}$$

$$P_{Li} + jQ_{Li} = \sum_{k=1}^{n} V_i V_k V_{ik} e^{j(\theta_i - \theta_k - \alpha_{ik})}. \qquad i = m+1,...,n. \tag{A.12}$$

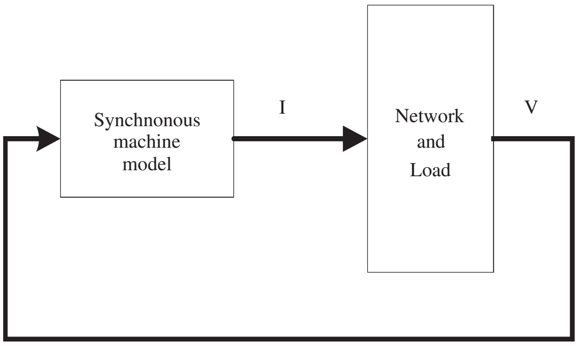


Figure A.2: Complete system model

The model equations represents *m*- machine, and *n*-bus system. The loads are represented as constant impedance loads and transmission lines are considered as R and L elements. Block diagram of the complete model is shown in Figure A.2. The synchronous machine is modeled as voltage input and current output model. Network algebraic equations are evaluated for terminal voltages by using the currents.

Appendix B

39 Bus New England Test System

The single line diagram, bus data, line data, generator data, and exciter system data are given in the following pages. They are adopted from [23].

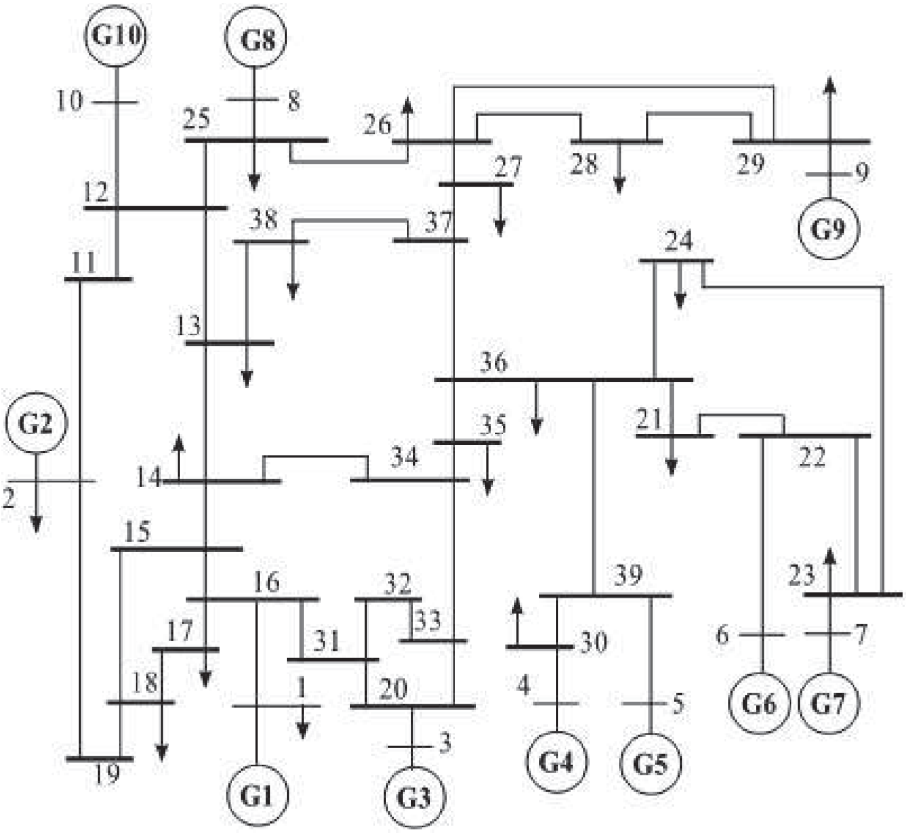


Figure B.1: Single line diagram of 39 bus New England test system

Table B.1: Bus Data

Bus	Type	Voltage	Pgen	Qgen	Pload	Qload
1	1	0.982	0	0	0.092	0.046
2	2	1.03	10	0	11.04	2.5
3	2	0.9831	6.5	0	0	0
4	2	1.0123	5.08	0	0	0
5	2	0.9972	6.32	0	0	0
6	2	1.0493	6.5	0	0	0
7	2	1.0635	5.6	0	0	0
8	2	1.0278	5.4	0	0	0
9	2	1.0265	8.3	0	0	0
10	2	1.0475	2.5	0	0	0
11	3	1	0	0	0	0
12	3	1	0	0	0	0
13	3	1	0	0	3.22	0.024
14	3	1	0	0	0	1.84
15	3	1	0	0	0	0
16	3	1	0	0	0	0
17	3	1	0	0	2.338	0.84
18	3	1	0	0	5.22	1.76
19	3	1	0	0	0	0
20	3	1	0	0	0	0
21	3	1	0	0	2.74	1.15
22	3	1	0	0	0	0
23	3	1	0	0	2.745	0.8466
24	3	1	0	0	3.086	0.922
25	3	1	0	0	2.24	0.472

26	3	1	0	0	1.39	0.17
27	3	1	0	0	2.81	0.755
28	3	1	0	0	2.06	0.276
29	3	1	0	0	2.835	0.269
30	3	1	0	0	6.285	1.03
31	3	1	0	0	0	0
32	3	1	0	0	0.075	0.88
33	3	1	0	0	0	0
34	3	1	0	0	0	0
35	3	1	0	0	3.2	1.53
36	3	1	0	0	3.294	0.323
37	3	1	0	0	0	0
38	3	1	0	0	1.58	0.3
39	3	1	0	0	0	0

Table B.2: Line Data

From Bus	To Bus	Resistance	Reactance	Line charging	Tap Ratio	Phase shift
1	16	0	0.025	0	1	0
2	11	0.001	0.025	0.75	1	0
2	19	0.001	0.025	1.2	1	0
3	20	0	0.02	0	1	0
4	30	0.0009	0.018	0	1	0
5	39	0.0007	0.0142	0	1	0
6	22	0	0.0143	0	1	0
7	23	0.0005	0.0272	0	1	0
8	25	0.0006	0.0232	0	1	0

9	29	0.0008	0.0156	0	1	0
10	12	0	0.0181	0	1	0
11	12	0.0035	0.0411	0.6987	1	0
12	13	0.0013	0.0151	0.2572	1	0
12	25	0.007	0.0086	0.146	1	0
13	14	0.0013	0.0213	0.2214	1	0
13	38	0.0011	0.0133	0.2138	1	0
14	15	0.0008	0.0128	0.1342	1	0
14	34	0.0008	0.0129	0.1382	1	0
15	16	0.0002	0.0026	0.0434	1	0
15	18	0.0008	0.0112	0.1476	1	0
16	17	0.0006	0.0092	0.113	1	0
16	31	0.0007	0.0082	0.1389	1	0
17	18	0.0004	0.0046	0.078	1	0
18	19	0.0023	0.0363	0.3804	1	0
20	31	0.0004	0.0043	0.0729	1	0
20	33	0.0004	0.0043	0.0729	1	0
21	22	0.0008	0.0135	0.2548	1	0
21	36	0.0008	0.0135	0.2548	1	0
22	23	0.0006	0.0096	0.1846	1	0
23	24	0.0022	0.035	0.361	1	0
24	36	0.0003	0.0059	0.068	1	0
25	26	0.0032	0.0323	0.513	1	0
26	27	0.0014	0.0147	0.2396	1	0
26	28	0.0043	0.0474	0.7802	1	0
26	29	0.0057	0.0625	1.029	1	0
27	37	0.0013	0.0173	0.3216	1	0

28	29	0.0014	0.0151	0.249	1	0
30	39	0.0007	0.0138	0	1	0
31	32	0.0016	0.0435	0	1	0
32	33	0.0016	0.0435	0	1	0
33	34	0.0009	0.0101	0.1723	1	0
34	35	0.0018	0.0217	0.366	1	0
35	36	0.0009	0.0094	0.171	1	0
36	37	0.0007	0.0089	0.1342	1	0
36	39	0.0016	0.0195	0.304	1	0
37	38	0.0007	0.0082	0.1319	1	0

Table B.3: Machine Data

M/c	Xls	Rs	Xd	Xd’	Xd”	Td0’	Tdo”	Xq	Xq’	Xq”	Tq0’	Tqo”	H
1	0.035	0.0015	0.295	0.07	0.07	6.56	0.041	0.282	0.17	0.17	1.5	0.15	30.3
2	0.003	0.0068	0.02	0.006	0.006	6	0.0375	0.019	0.008	0.008	0.7	1.07	500
3	0.03	0.0113	0.2495	0.0531	0.0531	5.7	0.035625	0.237	0.088	0.088	1.5	0.15	35.8
4	0.054	0.0113	0.67	0.132	0.132	5.4	0.03375	0.62	0.166	0.166	0.044	0.044	26
5	0.0295	0.0046	0.262	0.0436	0.0436	5.69	0.03556	0.258	0.166	0.166	1.5	0.15	28.6
6	0.0224	0.0113	0.254	0.05	0.05	7.3	0.045625	0.241	0.081	0.081	0.4	0.04	34.8
7	0.0322	0.0024	0.295	0.049	0.049	5.66	0.035375	0.28	0.091	0.091	1.5	0.15	26.4
8	0.028	0.0024	0.29	0.057	0.057	6.7	0.041875	0.28	0.091	0.091	0.41	0.041	24.3
9	0.0298	0.0013	0.2106	0.057	0.057	4.79	0.029936	0.205	0.059	0.059	1.96	0.196	34.5
10	0.013	0.002	0.1	0.031	0.031	10.2	0.035625	0.069	0.069	0.069	0.5	0.05	42

Appendix C

Screen Captures

Screen captures of the Software modules and other consoles are presented in this chapter.

C.1 Screen Captures of System Model Implemented in MATLAB

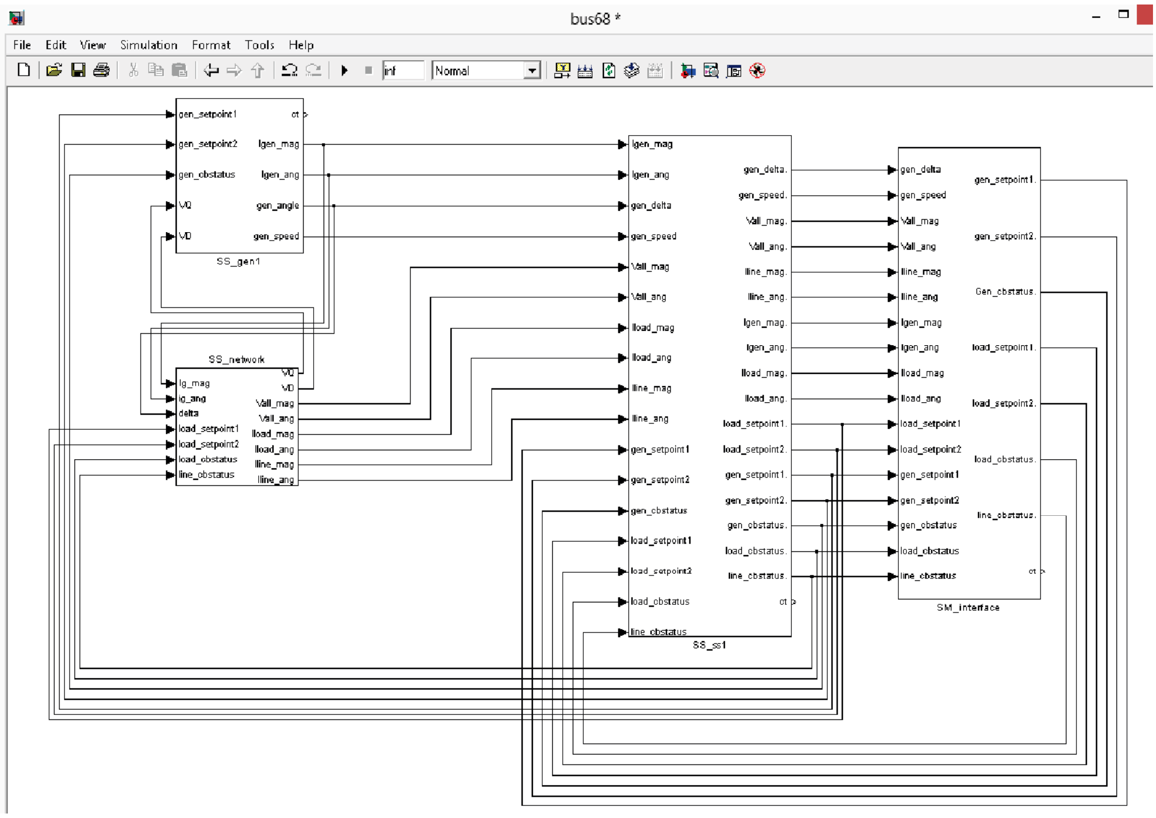


Figure C.1: Subsystems of system model

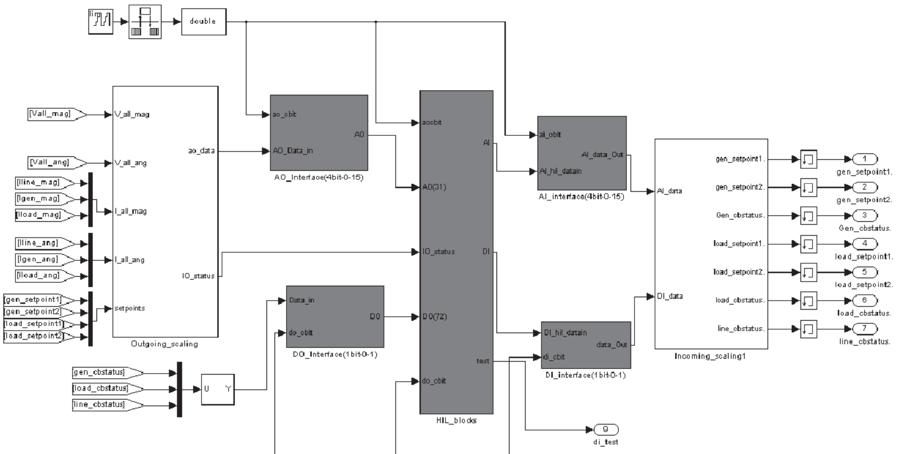


Figure C.2: Data handling in MATLAB

C.2 Screen Captures of Data Acquisition and Logging Implemented in LabVIEW

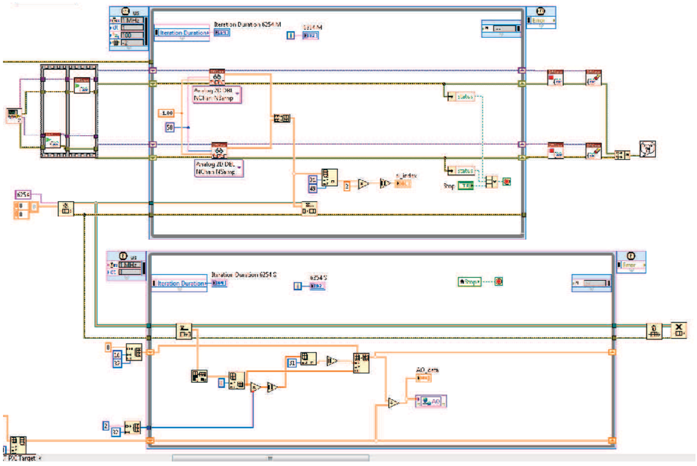


Figure C.3: Screen capture of analog data reading NI-PXI program

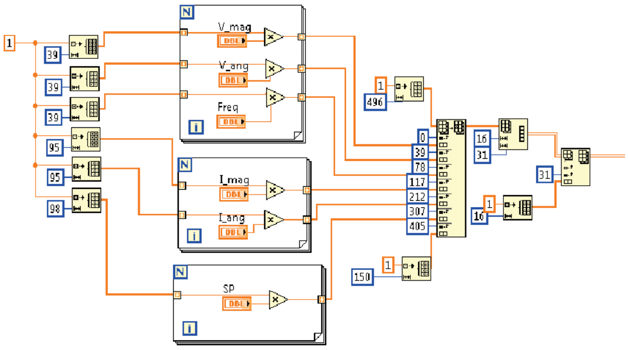


Figure C.4: Screen capture of analog data scaling program

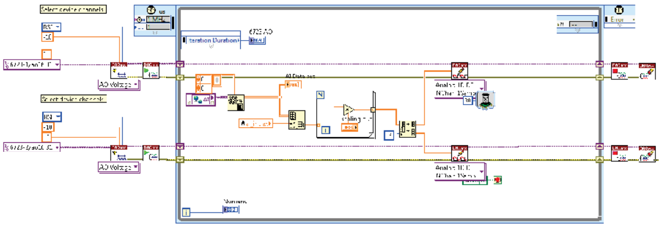


Figure C.5: Screen capture of analog data writing NI-PXI program

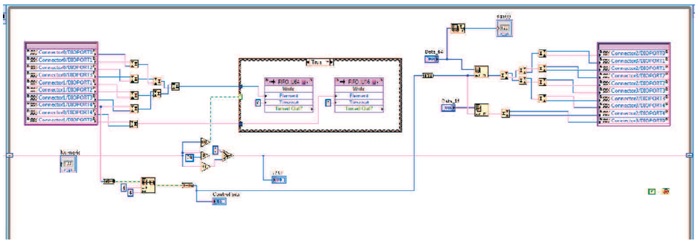
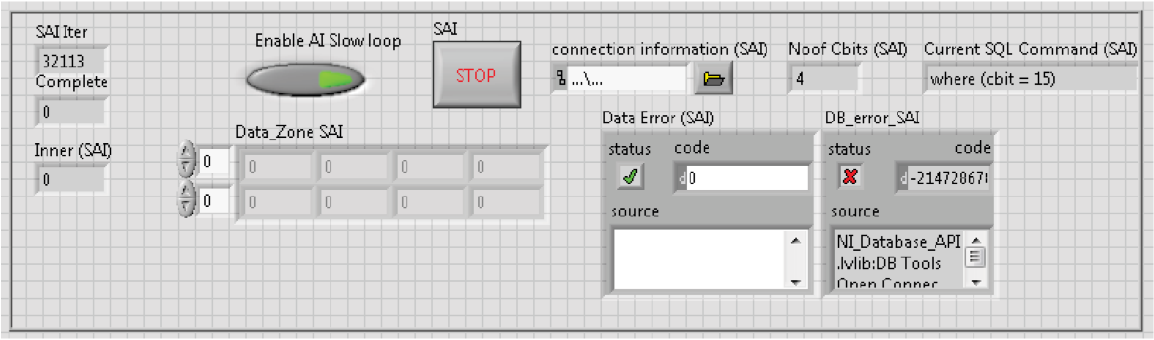


Figure C.6: Screen capture of digital data reading and writing NI-PXI program



SAI Iter
32113

Complete
0

Inner (SAI)
0

Enable AI Slow loop

SAI
STOP

connection information (SAI)
...\...

Noof Cbits (SAI)
4

Current SQL Command (SAI)
where (cbit = 15)

Data_Zone SAI

0	0	0	0	0
0	0	0	0	0

Data Error (SAI)
status ✓
code 0
source

DB_error_SAI
status ✗
code -21472867
source
NI_Database_API.lvlib:DB Tools Open Connec

Figure C.7: Screen capture of data logging software front-end

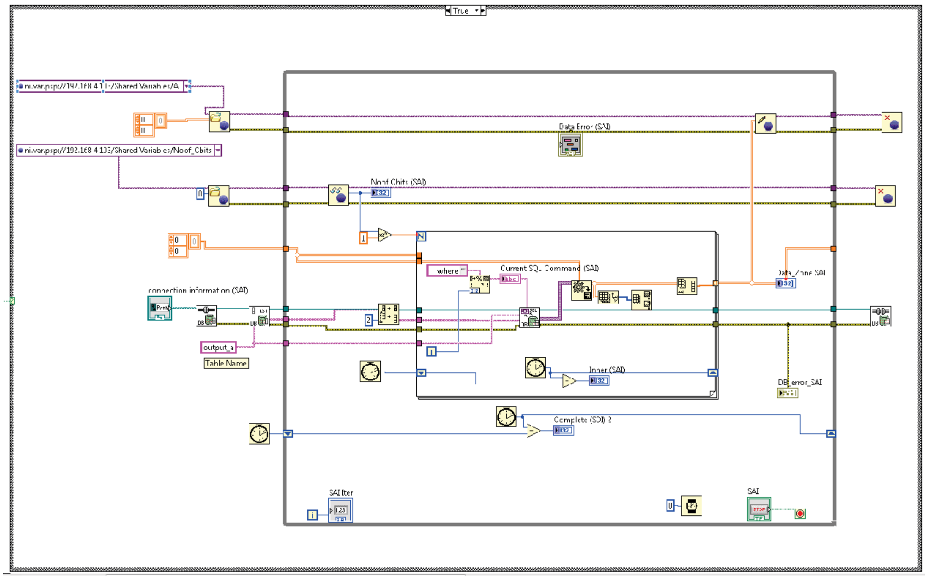
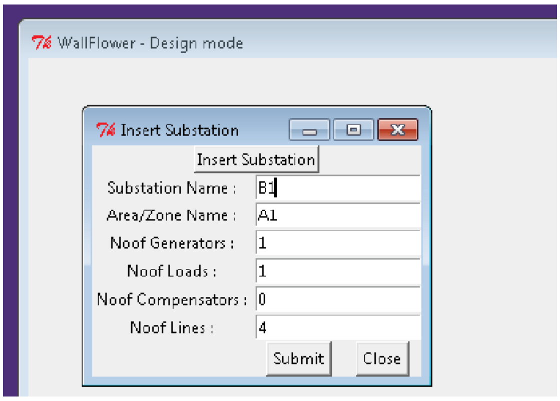


Figure C.8: Screen capture of data logging software back-end

C.3 Screen Captures of Visual Interface

C.3.1 Design Window screen captures



Substation Name : B1
Area/Zone Name : A1
Noof Generators : 1
Noof Loads : 1
Noof Compensators : 0
Noof Lines : 4

Figure C.9: Screen capture of substation data form

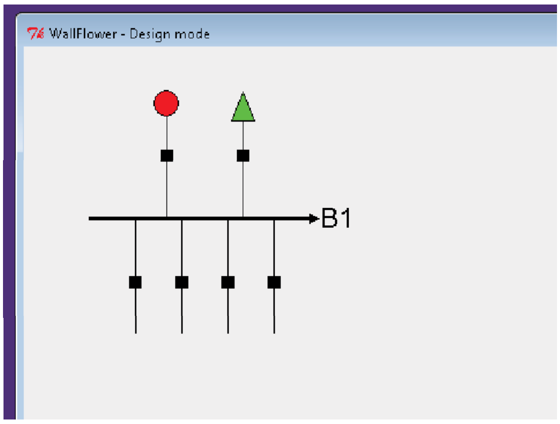
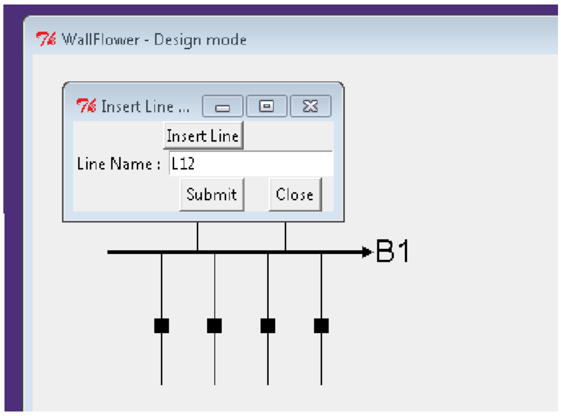


Figure C.10: Screen capture of front-end of substation data form



Line Name : L12

Figure C.11: Screen capture of insert line dialog window

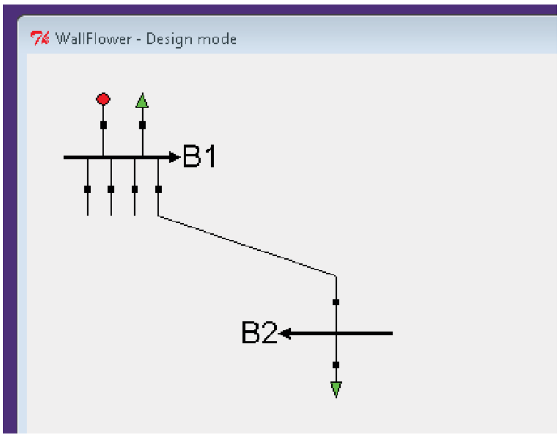
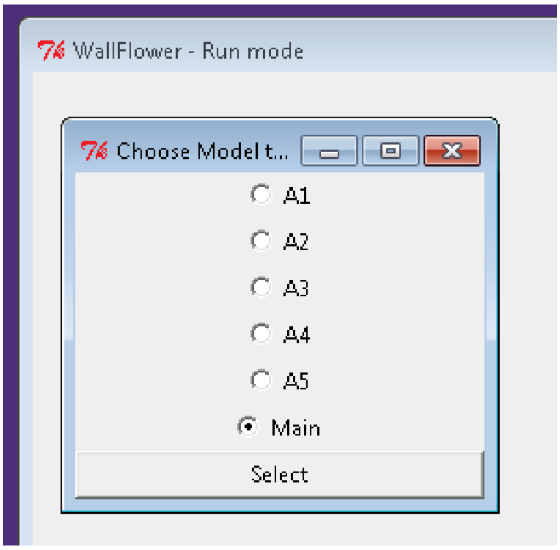


Figure C.12: Screen capture of line connection

C.3.2 Screen Captures of Run Window



○ A1

○ A2

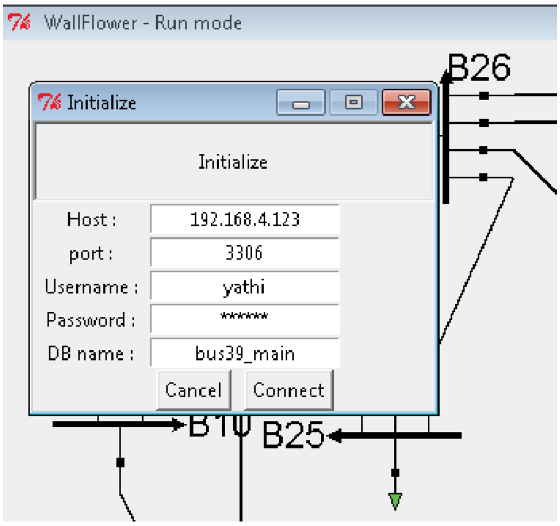
○ A3

○ A4

○ A5

● Main

Figure C.13: Screen capture of case loading options



Host : 192.168.4.123

port : 3306

Username : yathi

Password : ******

DB name : bus39_main

Figure C.14: Screen capture of initialization of model from database

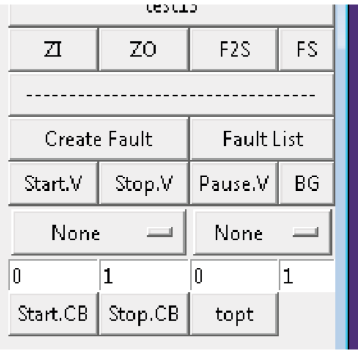


Figure C.15: Screen capture of run window menu buttons

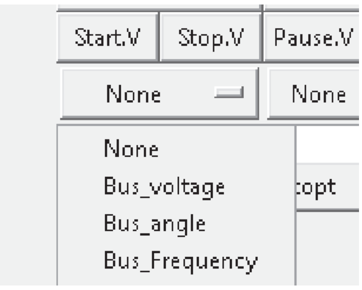


Figure C.16: Screen capture of visualization options

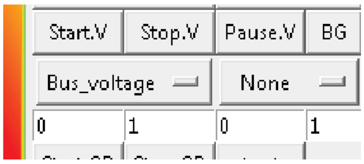


Figure C.17: Visualization parameter limits

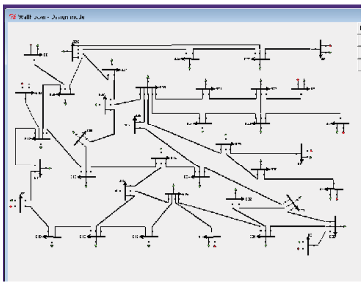


Figure C.18: Screen capture of complete system

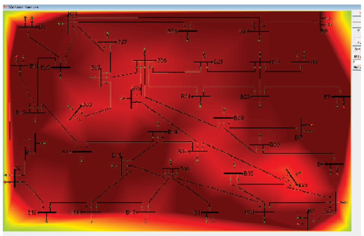


Figure C.19: Screen capture of model with visualization limits (0 to 1)

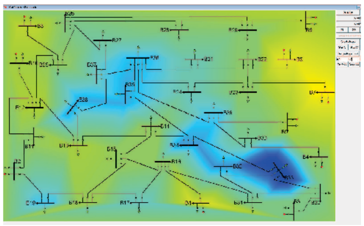


Figure C.20: Screen capture of model with visualization limits (0.9 to 1.1)

References

[1] U. Hager, C. Rehtanz, and N. Voropai, *Monitoring, Control and Protection of Interconnected Power Systems*. Berlin, Germany, Springer-Verlag, 2014.

[2] [Online]. Available: http://www.opal-rt.com/product/rt-lab-professional-real-time-digital-simulation-software.

[3] [Online]. Available: www.ni.com/pxi.

[4] T. Overbye, G. Gross, M. Laufenberg, and P. Sauer, "Visualizing power system operations in an open market," *IEEE Comput. Appl. Power*, vol. 10, no. 1, pp. 53–58, Jan. 1997.

[5] V. Terzija, G. Valverde, D. Cai, P. Regulski, V. Madani, J. Fitch, S. Skok, M. Begovic, and A. Phadke, "Wide-area monitoring, protection, and control of future electric power networks," *in Proc. IEEE*, vol. 99, no. 1, pp. 80–93, Jan. 2011.

[6] A. Phadke, "Synchronized phasor measurements in power systems," *IEEE Comput. Appl. Power*, vol. 6, no. 2, pp. 10–15, Apr. 1993.

[7] L. Vanfretti, M. Chenine, M. Almas, R. Leelaruji, L. Angquist, and L. Nordstrom, "SmarTS lab - a laboratory for developing applications for WAMPAC systems," in *Proc. IEEE/Power Eng. Soc. General Meeting*, Jul. 2012, pp. 1–8.

[8] S. Biswas, F. Shariatzadeh, R. Beckstrom, and A. Srivastava, "Real time testing and validation of smart grid devices and algorithms," in *Proc. IEEE/Power Eng. Soc. General Meeting*, Jul. 2013, pp. 1–5.

[9] S. Biswas, J. H. Kim, and A. Srivastava, “Development of a smart grid test bed and applications in PMU and PDC testing,” in *Proc. North American Power Symp. (NAPS)*, Sep. 2012, pp. 1–6.

[10] M. Golshani, G. Taylor, I. Pisica, and P. Ashton, “Laboratory-based deployment and investigation of PMU and openPDC capabilities,” in *Proc. 10th IET Int. Conf. AC and DC Power Transmission (ACDC 2012)*, Dec. 2012, pp. 1–6.

[11] M. Larsson, P. Korba, and M. Zima, “Implementation and applications of wide-area monitoring systems,” in *in Proc. IEEE/Power Eng. Soc. General Meeting*, Jun. 2007, pp. 1–6.

[12] S. Muller, H. Georg, C. Rehtanz, and C. Wietfeld, “Hybrid simulation of power systems and ICT for real-time applications,” in *Proc. 3rd IEEE PES Int. Conf. Innovative Smart Grid Technologies (ISGT Europe)*, Oct. 2012, pp. 1–7.

[13] D. Babazadeh, M. Chenine, K. Zhu, L. Nordstrom, and A. Al-Hammouri, “A platform for wide area monitoring and control system ICT analysis and development,” in *Proc. IEEE PowerTech*, Jun. 2013, pp. 1–7.

[14] F. Kawano, P. Beaumont, A. Ishibashi, K. Hamamatsu, Y. Tada, and Y. Serizawa, “Development of prototype wide-area monitoring, protection and control (WAMPAC) systems based upon international standards,” in *Proc. IEEE PowerTech*, Jun. 2013, pp. 1–6.

[15] M. Stanovich, I. Leonard, K. Sanjeev, M. Steurer, T. Roth, S. Jackson, and M. Bruce, “Development of a smart-grid cyber-physical systems testbed,” in *Proc. IEEE PES Int. Conf. Innovative Smart Grid Technologies (ISGT)*, Feb. 2013, pp. 1–6.

[16] H. Georg, S. Muller, N. Dorsch, C. Rehtanz, and C. Wietfeld, “INSPIRE: Integrated co-simulation of power and ICT systems for real-time evaluation,” in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Oct. 2013, pp. 576–581.

[17] [Online]. Available: http://www.opal-rt.com/product/emegasim-powergrid-real-time-digital-hardware-in-the-loop-simulator.

[18] [Online]. Available: http://sine.ni.com/nips/cds/view/p/lang/en/nid/13862.

[19] [Online]. Available: http://sine.ni.com/nips/cds/view/p/lang/en/nid/14127.

[20] [Online]. Available: http://sine.ni.com/nips/cds/view/p/lang/en/nid/14044.

[21] N. Instruments.(2013). Buffered network-published shared variables: Components and architecture. [Online]. Available: http://www.ni.com/white-paper/12176/en/.

[22] P. W. Sauer and M. A. Pai, *Power System Dynamics and Stability*. New Delhi, India, Prentice-Hall of India Pvt. Ltd., 1998.

[23] M. A. Pai, *Energy Function Analysis for Power System Stability*. New York, Kluwer Academic Publishers, 1989.