Cape Peninsula
University of Technology

PERFORMANCE ANALYSIS OF A PROTECTION SCHEME BASED ON P-CLASS SYNCHROPHASOR MEASUREMENTS

By

EVERETT MONDLIWETHU MTHUNZI

Thesis submitted in fulfilment of the requirements for the degree

Master of Engineering : **Electrical Engineering**

In the Faculty of : Engineering

At the Cape Peninsula University of Technology

Supervisor : Prof R. Tzoneva

Co-supervisor : Mr. C. Kriger

Campus : Bellville campus

Date submitted : 15 June 2016

Declaration

This thesis is a presentation of my original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions. The work was done under the guidance of Professor R. Tzoneva, at the Cape Peninsula University of Technology in Cape Town.

Everett. M. Mthunzi		
Name	Signature	Date

In my capacity as supervisor of the candidate's thesis, I certify that the above statements are true to the best of my knowledge.

Professor R. Tzoneva		
Name	Signature	Date

Abstract

Power grid and system protection advancement greatly depend on technological advances. Advent technologies like digital microprocessor type protective relays facilitate paradigm shifts, providing inimitable beneficial engineering adaptations. Phasor measuring technology provides one such technological advance. The onset and rapid development of the Phasor Measuring Unit (PMU) provides an excellent platform for phasor-based, power system engineering.

Power transmission constitutes a critical section in the electric power system. The power system transmission lines are susceptible to faults which require instant isolation to establish and maintain consistent system stability. This research focuses on the study of transmission line protection based on P-Class synchrophasor measurements. The IEEE C37.238-2011 Precision Time Protocol (PTP) paradigm shift facilitates practical application of synchrophasors in protection schemes.

Synchrophasor procession and accurate data alignment over wide areas support the hypothesis of a phasor-based transmission line differential protection. This research aims to directly implement P-Class synchrophasors in transmission line differential protection, employing synchrophasors to determine fault conditions and administer corresponding protective actions in wide area transmission lines. The research also aims to evaluate the operational characteristics of the synchrophasor-based transmission line differential protection scheme.

The research deliverables include a laboratory scale Test-bench that implements the PMU-based transmission line differential protection scheme, and a differential protection utility software solution that follows guidelines specified by the C37.118-2011 standard for synchrophasors.

The findings stand to evaluate performance of the PMU-based line differential protection scheme, verifying the protection model as an alternate, practical and feasible backup protection solution. The research deliverables include a synchrophasor-based current differential algorithm, software utility for implementing the PMU-based protection scheme and a Test-bench for concept and feasibility validation.

Keywords: P-Class synchrophasor measurements, IEEE C37.238-2011 precision time protocol, C37.118 guidelines, line differential protection.

Acknowledgements

I would like to thank God for overseeing my wellbeing throughout all the years of my study. I also wish to thank:

Professor R. Tzoneva, (Head of the Centre for Substation Automation and Energy Management System – CSAEMS) (my supervisor) for her unconditional support all these years.

Mr. C. Y. Adewole, for his motivation, inspiration and consistent guidance.

Dedication

For my loving parents, Lucky and Ann Mthunzi, who have sacrificed everything to give me an opportunity to peruse my aspirations and dreams.

Table of Contents

Table of Figures

List of Tables

Glossary

Key Terms	**Description**
Economy	Maximum protection at minimum cost
Reliability	Consistent function in the occurrence of fault conditions
Simplicity	Minimization of protection circuitry and equipment
Selectivity	Avoiding unwarranted, false trips
Speed	Efficient and quick functionality reducing equipment damage and fault duration, with only very precise intentional time delays
Protection Scheme	An automatic protection system designed to detect abnormal or predetermined system conditions, and take corrective actions other than and/or in addition to the isolation of faulted components to maintain system reliability

Nomenclatures	Description
PMU	Phasor Measuring Unit
GPS	Global Positioning System
PC	Personal Computer
IED	Intelligent Electronic Device
1PPS	1 Pulse Per Second
NTP	Network Time Protocol
SNTP	Simple Network Time Protocol
PTP	Peer to Peer
SAS	Substation Automation System
SCL	Substation Configuration description Language
VLAN	Virtual Local Area Network
GOOSE	Generic Object Oriented Substation Event
MMS	Manufacturing Message Specification
TCP	Transmission Control protocol
UDP	User Datagram Protocol
SV	Sample Values
SONET	Synchronous Optical Networking
SDH	Synchronous Digital Hierarchy
MW	Mega Watts
MVAR	Megavars
SCADA	Supervisory Control and Data Acquisition
WAMS	Wide Area Monitoring System
WECC	Western Electricity Coordination Council
PDC	Phasor Data Concentrator
TVE	Total Vector Error
ROCOF	Rate of Change of Frequency
CRC	Cyclic Redundancy Check
87L	Current Differential Protection Relays
PCDAA	Phasor Current Differential Action Adapter
FDD	Feature Driven Development

Symbols	**Description**
$\bar{\iota}$	Phasor magnitude
θ	Phasor angle
i_1	Current phasor from PMU_1
i_2	Current phasor from PMU_2
$\bar{\iota}_1$	PMU_1 current phasor magnitude
$\bar{\iota}_1$	PMU_2 current phasor magnitude
$\bar{\iota}_3$	Current phasor differential magnitude
θ_1	PMU_1 phasor angle
θ_2	PMU_2 phasor angle
θ_3	Current phasor differential angle
PV	Pick up Value
I_{RES}	Restraint Current
i_{DIFF}	Current Differential Signal
I_{DIFF}	Filtered Differential Signal
T_{wait}	Wait time
$T_{first\ PMU}$	Arrival of first PMU frame
T_{total}	Total latency of the PDC
T_{output}	PDC output capture time
$T_{process}$	Time stamp of the processing time
$T_{last\ input}$	Time of capture of the last PMU frame
$T_{PDC\ Process}$	Time stamp for the processing time
$T_{delayed\ PMU}$	Last acceptable delayed PMU frame input

Chapter 1 :
Introduction

1.1 Introduction

Power system protection and communication are crucial to ensure reliable and consistent generation, transmission and distribution of power. The motivation for synchrophasor technology development and application, together with an overview of the C37.238-2011 standard for synchrophasors are introduced in this Chapter, culminating awareness of utilizing synchrophasor in power system protection applications.

Chapter One describes the background, motivating advent Phasor Measuring Unit (PMU) technology. The background objectively highlights the shortcomings of Supervisory Control and Data Acquisition (SCADA) technology, emphasizing inadequate transmission rates of system state information employed in system control and system protection engineering. Detailed investigation into the development of Phasor Measuring Unit (PMU) technology emerges awareness of the problem, where utility and advancement of PMU technology leads its industrial application which is narrowly limited to wide-area monitoring. A broader investigation encompassing guidelines, standards and protocols governing development and deployment of PMU technology leads to identification of the IEEE C37.238-2011 Precision Time Protocol (PTP) which presents a platform that suggests utilizing synchrophasors directly for power system control and system protection culminating the problem statement from which the hypothesis is formulated. With a hypothesis the research aims and objectives are clearly defined, identifying assumptions and recognizing both the delimitations and the limitations of the research.

With a problem statement and clearly defined; aims, objectives, hypothesis and research considerations, a research methodology and method are selected, formulating a framework of thesis Chapters.

1.1.1 Chapter Organization

Section 1.2 describes the research motivation.

Section 1.3 presents the problem statement and identifies essential sub-problems.

Section 1.4 defines the research aims and objectives.

Section 1.5 presents the research hypotheses.

Section 1.6 outlines limitations and delimitations of the research.

Section 1.7 reviews all research assumptions considered.

Section 1.8 establishes the research methodology and methods.

Section 1.9 presents the framework of thesis chapters.

Section 1.10 concludes the chapter.

1.2 Research motivation

1.2.1 Motivation behind synchrophasor technology

Figure 1.1 illustrates a power system monitor and control paradigm based on long established SCADA technology. The accuracy and precision of monitor and control functions depend on transmission rates of system state information. In worst case scenarios, SCADA report rates are limited between four to six samples per second (4-6) / sec. These low report rates propagate poor system response during fast transients and disturbance scenarios.

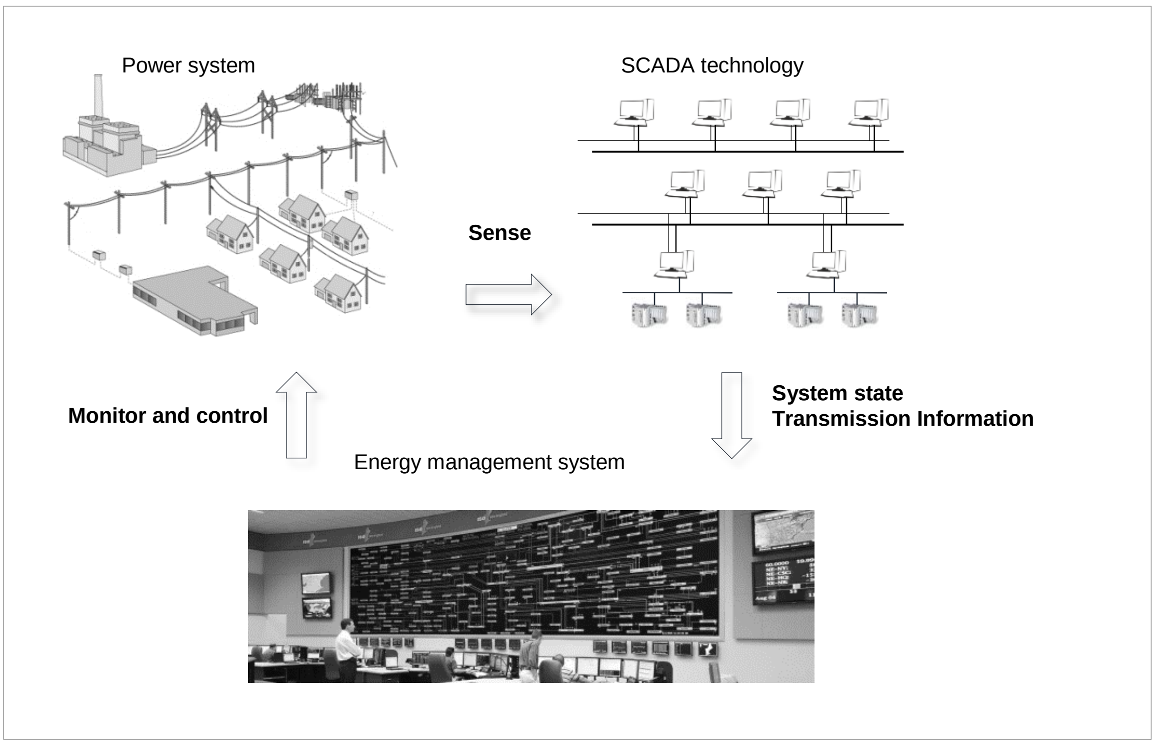


Figure 1.1: Power system control and monitor paradigm

SCADA technology measurements are also limited to the current and voltage magnitudes. They neglect the angular measurement component which provides crucial real-time indication of system stress in the transmission corridors.

PMUs provide voltage and current magnitude measurements together with magnitude phase angles typically at a rate of 48 samples per second. A global time source enables coherent time synchronization between real-time phasor measurements. Table 1.1 shows a general comparison between SCADA and PMU technology.

Table 1.1: Comparison between SCADA and PMU technology

Specification	SCADA technology	PMU technology
Report rate	4 – 6 samples / seconds	30 / 60Hz
Indication of system stress	NO	YES
Synchronized wide-area monitoring	NO	YES

In contrast to the conventional SCADA, advent PMU technology provides advanced engineering utility that facilitates more accurate system monitoring, protection and system control (Chen et al., 2008). The research focuses on application of PMU technology in transmission line differential protection.

1.2.2 IEEE C37.238-2011 Precision Time Protocol in power system applications

Power system applications require high precision time synchronization. The IEC/IEEE 61850-9-3 and the IEEE C37.238-2011 standards specify Precision Time Protocol (PTP) parameters and specification guidelines that provide global time availability, device interoperability, and failure management (Wang, 2010). The PTP parameters and specification guidelines enable IEEE 1588-based time synchronization in power system monitoring, protection, and control applications that emphasize critical data communication. The IEEE C37.238-2011 standard extends profile for the use of IEEE standard 1588-2008 Precision Clock Synchronization Protocol for Networked Measurement and Control in power system monitoring, protection and control applications that make use of the Ethernet media communications architecture (IEEE Power and Energy Society, 2011a).

Most power systems rely on conventional IRIG-B and one pulse per second (1PPS) signals to establish synchronization between system Intelligent Electronic Devices (IEDs).

The transmission of the signals requires dedicated wiring which is always coupled with high maintenance costs and stringent scalability options. Advent Ethernet communication technology provides a paradigm shift with new concepts of time synchronization based on network protocols like the Network Time Protocol (NTP) and Simple Network Time Protocol (SNTP) (Mills, 2014).

Substation systems require sub-microsecond precision for all substation applications governed by the IEC 62850 and IEEE C37.118-2011 standard performance requirements. The IEEE C37.238-2011 standard facilitates the IEEE 1588 sub-microsecond network-based time synchronization protocol with a PTP profile specific to power system applications. The PTP profile maintains separation of functions and establishes an overall steady-state synchronization requirement of one microsecond worst-case time error over 16 networks hops (IEEE Power and Energy Society, 2011a).

1.2.3 Awareness of the problem

The onset IEEE C37.238-2011 precision time protocol and the rapid development of PMU technology provides an excellent platform for phasor-based, power system engineering with the IEEE C37.238-2011 facilitating practical application of synchrophasors in wide-area monitoring, control and novel protection schemes. PMUs provide adept and advanced engineering concepts which include wide-area;

a. State estimation:

   Power system State Estimation (SE) facilitates real-time power system status for monitoring, protection and control (Hongga, 2007), (Chen et al., 2008), (Jain, 2009) and (Xie et al., 2012).

b. Resource integration:

   Real-time phasor measurement systems provide advanced monitoring, managing and integrating power sub-systems plants into a single power system (Nexans & Nerc, 2012).

c. Power oscillation monitoring:

   The exceptionally high PMU data rates enable detection of power system oscillations and ambient grid damping (Ghaisari et al., 2005), (Korba, 2008), (Farantatos et al., 2009), (Keyvani et al., 2014) and (Andersson, 1994).

d. Voltage monitoring and trend:

Phasor measuring utilities facilitate monitoring, control and protection functions that manage power system frequency and voltage.

The utilities also provide trend analysis for trending system voltages (Nexans & Nerc, 2012).

e. Frequency stability monitoring and trend:

System frequency provides an excellent indication of system integrity throughout system events involving separation or islanding (Park, 2010). System frequency also indicates load-resource balance correlating the frequency deviation to generation loss.

f. Dynamic line ratings and congestion management:

Phasor data can be used to monitor transmission line loadings and recalculate line ratings in real-time (Nexans & Nerc, 2012).

Consistent development of PMU technology supports the necessity to broaden the application scope of the technology. Acknowledgement and acceptance of the IEEE C37.238-2011 precision guideline make it possible to consider application of phasor technology in power system protection applications.

Establishing a phasor technology-based protection solution would facilitate alternate or back-up protection models capable of fully exploiting the inimitable benefits of phasor technology.

1.3 Problem statement and sub-problems

1.3.1 Problem statement

Design, development and implementation of a PMU-based line differential protection scheme to ascertain concept feasibility and provide alternative back-up protection for transmission lines based on phasor measuring technology.

1.3.2 Sub-problem one

Model a PMU-based line differential protection scheme considering prevailing PMU applications and principles for line differential protection.

1.3.3 Sub-problem two

Develop a phasor-current differential algorithm for fault detection and isolation to be administered by the phasor concentration software solution.

1.3.4 Sub-problem three

Develop a software utility capable of phasor data concentration, and specific to transmission line current differential protection.

1.3.5 Sub-problem four

Design, set-up, and configuration of a virtual, digital real-time Test-bench, to simulate a real-time transmission line system and its PMU-based differential protection scheme.

1.3.6 Sub-problem five:

Determine the performance and response of the PMU-based line differential protection scheme, to evaluate concept feasibility and establish quantified metrics for the protection model.

1.3.7 Sub-problem six:

Establish applicability of PMU measurements in transmission line differential protection.

1.4 Research aims and objectives

1.4.1 Research aims

The aim of the research is to prove and validate the concept of PMUs in transmission line differential protection to provide alternate or back-up transmission line protection that employs phasor measurement technology. This research also aims to provide a platform for further research and development in protection applications based on phasor measurement technology. Figure 1.2 presents the theoretic concept of the PMU-based line differential protection scheme.

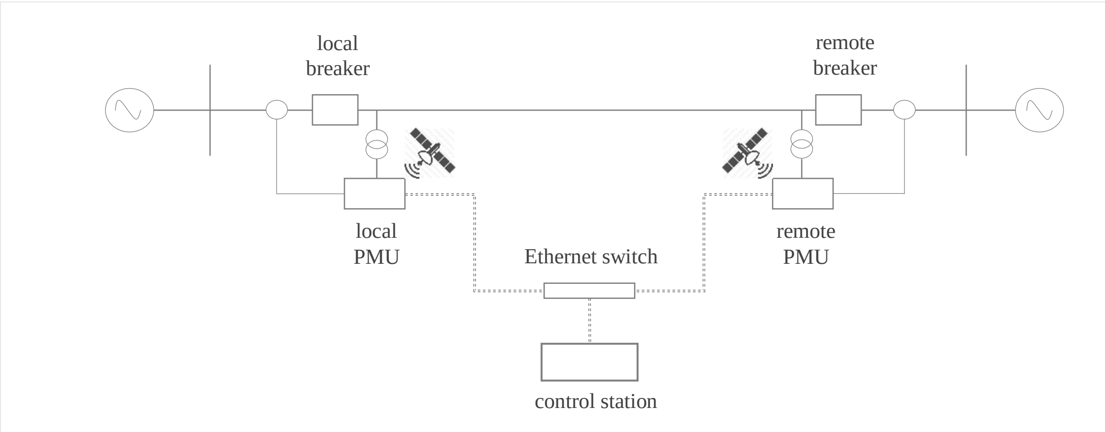


Figure 1.2: PMU-based, line differential protection concept

1.4.2 Research objectives

The objectives of the research include;

i. modelling a transmission line differential protection scheme based on PMU synchrophasors
ii. engineering a software utility that employs real-time phasor data concentration and administers differential protection and control algorithms
iii. modelling differential protection and control algorithms to be administered by the engineered software utility
iv. establishing a Test-bench to validate the hypothesis of the PMU-based line differential protection scheme
v. critically analyze performance of the PMU-based line deferential protection scheme, evaluating;
    - response of P-Class PMUs to short circuit (fault) fast transients
    - PMU response to conventional instrument transformers
    - impact of loss of PMU synchronization time
vi. report conclusive deductions of applicability of PMUs in transmission line differential protection based on the quantitative results obtained

1.5 Hypothesis

Change in approach to concepts and underlying assumptions specified by the IEEE C37.238-2011 precision time protocol, makes possible and facilitates implementation of synchrophasors in differential protection schemes for transmission lines.

1.6 Limitations and delimitations of the research

1.6.1 Delimitations

Application of the phasor-based line differential protection scheme is limited to a real-time simulation. RSCAD interface software is utilized to run a real-time transmission line system which is used to carry out the simulation study for the PMU-based line differential protection scheme.

No tests will be carried out on a real transmission line system. PMU application in the real-time simulation is limited to the Real-Time Digital Simulator (RTDS). No physical PMUs will be utilized in the real-time simulation. The research is restricted to the IEEE standards and protocols listed in Table 1.2.

Table 1.2: IEEE standard, protocol and guideline description

Standard / Protocol	Description
IEC 61850	Standard for communication in substation systems
C37.238-2011	Precision Time Protocol in power system applications
C37.118-2011	Standard for synchrophasors
C37.244-2013	Guide for phasor data concentrator requirements

1.6.2 Limitations

Performance specifications of the workbench desktop personal computer (PC) and Ethernet data transmission media encompass the physical limitations that determine overall quality of the simulation.

1.7 Assumptions

The following assumptions are made;

- Interoperable PMUs can be interfaced with a transmission line.
- A non-proprietary software can be developed to implement synchrophasor-based transmission line current differential protection and control functions.
- Synchrophasors from multiple PMUs in the line can be used to facilitate line protection.
- Both PB5 and GTNET RSCAD components are available and fully functional.
- The PB5 component includes analogue output channels with digital to analog converters and is equipped with transceivers to facilitate connection of peripheral input-output (I/O) devices at high speed.

1.8 Significance of research

This section emphasizes the significance of the research, describing contribution in the sub-field of transmission line system protection while recognizing a gap in knowledge with regard to expanding development of PMU applications to fully utilize all features facilitated by PMU utility.

1.8.1 Contribution and Impact on smart grid and power system engineering

Conventional back-up techniques are not easily implemented and rely too heavily on main protection. This research identifies established concepts and proposes an innovative PMU-based protection scheme. The PMU-based hypothesis puts forward a potential transmission line back-up protection solution that uniquely compensates for the shortcomings of conventional back-up protection schemes.

A critical inclusive contribution is engineering of utility software to concentrate real-time phasor data and process it, administering differential protection and control algorithms in real-time.

1.8.2 Academic contribution

The laboratory scale Test-bench created to validate the hypothesis provides opportunity to conduct related PMU based-future work. The simulation guideline will help on future PMU simulation study.

1.9 Research methodology and methods

1.9.1 Research methodology

This research employs qualitative and quantitative applied research methodology, administering contemporary concepts and accepted theoretic principles to establish an alternative back-up protection model with immediate potential application.

1.9.2 Research methods

The research methods consist of five main parts;

1. data collection and literature review
2. software development
3. algorithm development
4. laboratory simulation study
5. documentation of research findings and analysis of the results

1.9.2.1 Data collection and literature review

A systematic search and review of published research on; transmission line differential protection, existing PMU applications and PMU applications in transmission line differential protection is conducted to establish techniques, methods and new concepts regarding using PMU technology for protection. The established theoretical framework will also identify knowledge gaps and provide a method pool from which working methods can be adopted and improved on.

1.9.2.2 Software development

The software development method outlines a framework that is used to structure and detail engineering of the software utility for processing real-time phasor data from multiple streaming PMUs. The method follows stages of software development which include;

1. analyzing software objectives and theoretic expectations
2. devising a plan or design for the software-based solution
3. implementation (coding) of the software
4. testing the software
5. deployment

1.9.2.3 Algorithm development

The Algorithm development method is used to create a mathematical process for establishing precise system differential protection and control for the PMU-based line differential protection scheme. The method follows steps which include;

1. development of a model
2. checking the correctness of Algorithm
3. analysis of Algorithm
4. implementation of Algorithm

1.9.2.4 Laboratory research

Laboratory studies are carried out to establish laboratory scale, transmission line system protection tests. A test bench is modelled and constructed to evaluate and analyze the performance of the PMU-based line differential protection scheme to ascertain its feasibility. Figure 1.3 presents the RTDS simulator and the RSCAD interface model components which will be utilized in the simulation studies to model the real-time transmission line model, the PMU current measurement and the data acquisition.

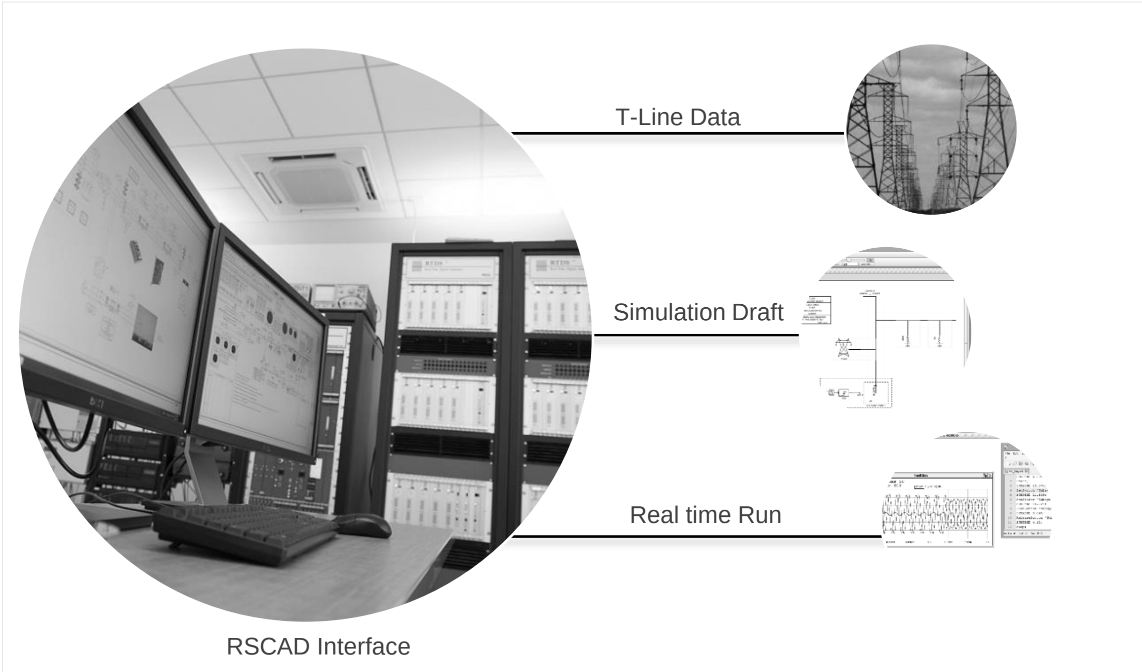


Figure 1.3: RTDS simulator and simulation study components

1.9.2.5 Documentation of research findings

All findings are documented and presented in this thesis, covering in great detail all design and real-time simulation test procedures carried out in the research. Documentation also includes a publication regarding obtained results.

1.10 Thesis Chapters

This Thesis is sub-divided into seven Chapters and two appendices. Table 1.3 provides a general overview of the Chapters and appendices coupled with content descriptions.

Table 1.3: Thesis Chapters

Chapter	Content / Description
Chapter Two	Literature review
Chapter Three	PMU-based Line differential Protection
Chapter Four	Phasor Current Differential Action Adapter (PCDAA) Software engineering
Chapter Five	Test-bench design
Chapter Six	Real-time simulation studies and experimental results
Chapter Seven	Conclusion, research findings and recommendation
Appendix A	Phasor Current Differential Action Adapter software source code
Appendix B	Configuration and parameters

1.10.1 Chapter Two: Literature Review

Chapter Two establishes a conclusive theoretic framework that provides specific validated concepts and solutions in; transmission line differential protection, the IEC 61850 communication standard, advent PMU technology and prevailing research and development in phasor measurement technology. Abstraction of the literature review enables identification and evaluation of critical concepts and relational methodologies.

1.10.2 Chapter Three: Feature and analysis of the PMU-based line differential protection scheme

Chapter Three presents the concept of the PMU-based line differential protection scheme. The collated engineering concepts from the literature review are utilized to model the PMU-based protection scheme.

1.10.4 Chapter Four: Engineering of the PCDAA utility software

This Chapter provides engineering outline and design specification of the Phasor Current Differential Action Adapter, a critical and core contribution in the research and development of the PMU-based transmission line protection scheme.

1.10.5 Chapter Five: The simulation study Test-bench

Chapter Four reviews contemporary or conventional IEC 61850 line differential protection scheme performance evaluation metrics. This Chapter also outlines development of a testing workbench for testing the PMU-based line differential protection scheme and its set up in the Center for Substation Automation and Energy Management Systems (CSAEMS) laboratory.

1.10.6 Chapter Six: Real-time simulation studies and experimental results

Chapter Six succinctly describes the real-time simulation procedure and experimental results. Utilizing schematics, illustrations, flow charts, tables and graphs, the chapter articulates engineering of the PMU-based line differential protection scheme and analyzes its performance under specific conditions.

1.10.7 Chapter Seven: Conclusion, research findings and recommendation

Thesis deliverables are presented. This Chapter also discusses recommendations for future work and possible extension for the research project.

1.10.8 Appendix A: Phasor Current Differential Action Adapter source code

Appendix A presents the final PDAA software, source code.

1.10.9 Appendix B: Guideline for configuration of RSCAD and RTDS parameters

Appendix B outlines network mapping and all configuration settings for RSCAD protection scheme test bench simulations.

1.11 Conclusion

This Chapter introduces PMUs providing a background and problem statement which outlines the knowledge gap in application of PMU-based technology and defines the specific problem to be addressed. The Chapter presents a hypothesis that describes a testable prediction highlighting delimitations and limitations of the theoretic concept of utilizing PMUs in transmission line protection. The significance and purpose of the study is also emphasized relating to research contribution.

Chapter Two presents a critical discussion that forms a methodological framework based on relevant concepts and technical solutions from existing research and prevailing development in transmission line protection and phasor measurement technology.

Chapter 2 :
Literature review

2.1 Introduction

Transmission line systems form an integral part of the electrical distribution system, providing power transfer between generation and load. Transmission lines systems also provide extensive operation for respective line utilities. In the occurrence of a fault, if undetected and not isolated timely, the disturbance may cause widespread outages throughout the transmission system network. Distribution and consumer power quality therefore depends on the transmission system utility response to disturbances, emphasizing identification and isolation of faults.

Protection schemes offer a stability solution to mitigate disturbance in power systems through providing a pragmatic approach to fault identification and isolation. Protection schemes establish selective, sensitive and efficient fault clearing to facilitate dependable and reliable power quality. Various schemes based on PMU technology have already been developed (Li et al., 1997), (Jiang et al., 2000), (Y. H. Lin et al., 2004), (Xu et al., 2007) and (Dalcastagnê et al., 2008).

The literature review described in this Chapter focuses on the IEC 61850, Phasor Measuring Unit (PMU) applications, and transmission line system protection. The IEC 61850 communication standard for electrical substation automation systems is reviewed to outline theoretic design considerations for modelling a modern transmission line differential protection scheme with regard to communication. The communication standard identifies prevailing criteria for modelling electrical substation communication architecture that emphasizes accuracy, dependability and efficiency.

A critical consideration is the migration from traditional protection utilities to modern and new high-performance technology protection utilities that provide functions that support advanced protection engineering. With clearly outlined criteria for modelling an IEC 61850 compliant transmission line protection scheme, a conceptual PMU-based transmission line differential protection scheme is modelled taking into account principles in transmission line differential protection and PMU application protocol guidelines.

This Chapter establishes a theoretic framework that systematically outlines principles, validated concepts and current methodologies used to draft a substandard guideline for designing and modelling the PMU-based transmission line differential protection scheme.

2.1 Chapter organization

Section 2.2 introduces the IEC 61850 communication standard for electrical substation automation systems, describing its background and presenting an analysis of the standards' implication on the electric smart grid.

Section 2.3 revises transmission lines and transmission line protection. In this section principles of line current differential protection are covered comprehensively. A synopsis of existing papers and research and development in line differential protection is also presented.

Section 2.4 introduces phasors describing the background of phasor measurement technology. PMU communication and application principles are studied focusing on evaluating standard impact of PMU technology on the electric grid.

Section 2.5 provides a critical discussion of collated literature. Subject matters from different authors, identified from the problem statement are comparatively studied analyzing contradictions and inconsistencies. The discussion realizes gaps and evaluates logical flaws in the literature reviews' body of knowledge, recognizes un-stated assumptions in arguments and distinguish facts from opinions.

Section 2.6 concludes the Chapter detailing an objective motivated approach to the research based on the assembled theoretic framework.

2.2 IEC 61850 Communication standard for electrical substation automation systems

2.2.1 Introduction

This section aims to provide an overview of the IEC 61850 Communication Standard, including as part of its scope; a background, the abstract communication model, function integration and implications of the IEC 61850 on the Smart Grid. It is necessary to consider the IEC 61850 to provide a contemporary PMU technology-based protection solution substantiated by inter-changeability and inter-operability requirements of the IEC 61850 communication standard for electrical substation automation systems.

2.2.2 IEC 61850 Background

Until recently, substation communication protocols have been limited to inceptive communication and wide area networking technologies. Advancement of communication technologies has enabled sustainable application of switched Ethernet and high speed networking technology into substation automation systems (SAS) which rely on communication to establish monitoring, control and protection functions (Goraj, 2010).

Between 2003 and 2005, the International Electro-technical Commission (IEC) developed and released the IEC 61850 Communication Standard (IEEE Power Engineering Society, 2009) to facilitate time-critical services and resolve long term interoperability and interchangeability between IEDs from different vendors (Liang et al., 2008). Table 2.1 illustrates the 14 parts of the IEC 61850 Communication Standard (IEEE Power Engineering Society, 2009).

Table 2.1: IEC 61850 Part Description (IEEE Power Engineering Society, 2009)

IEC 61850 Part:	Part Description
1	Introduction and Review
2	Glossary
3	General requirements
4	System protection and management
5	Communication requirements for functions and device models
6	Substation automation system configuration description language
7-1	Basic communication structure for substation and feeder equipment – ACSI
7-2	Basic communication structure for substation and feeder equipment – Principles and Models
7-3	Basic communication structure for substation and feeder equipment – Common data classes
7-4	Basic communication structure for substation and feeder equipment – Compatible logical node classes and data classes
8	Specific communication service mapping (SCSM) – Mapping to MMS (ISO/IEC 9506 part 1 – 2)
9-1	Specific communication service mapping (SCSM) – Serial unidirectional multi-drop point to point link
9-2	Specific communication service mapping (SCSM) – Mapping on a IEEE 802.3 base process bus
10	Conformance testing

2.2.3 IEC 61850 Communication

The IEC 61850 communication standard adopts and employs an object orientated hierarchal data model which focuses on the abstraction and definition of data items available from different primary equipment and substation function (Lei et al., 2014). The abstract concept is illustrated in Figure 2.1 (Electric Light and Power, 2015). Abstraction of definitions facilitates mapping of data objects and services to multiple protocols, which in turn provides adequate communication that adheres to data and service requirements of the IEC 61850 standard (Mackiewicz, 2004).

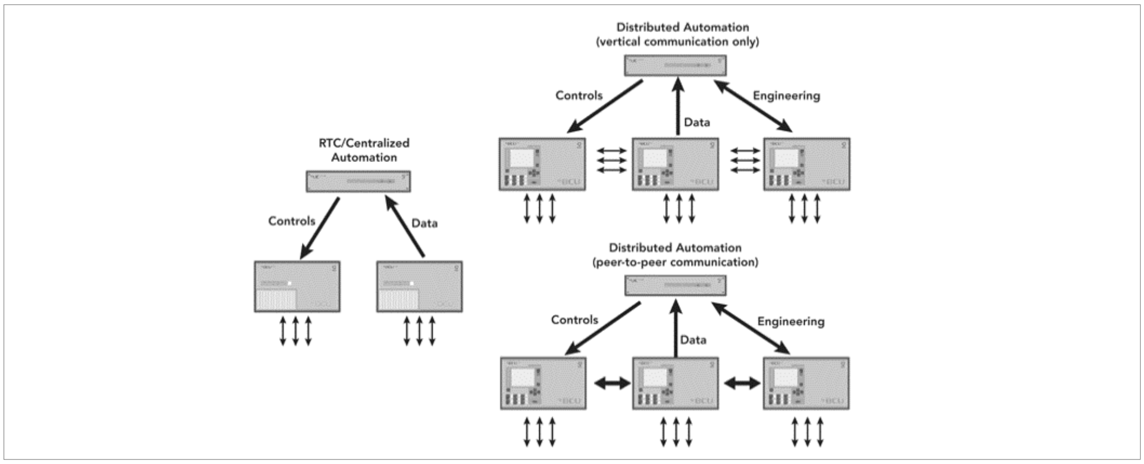


Figure 2.1: Object Orientated Communication Architecture (Electric Light and Power, 2015)

2.2.3.1 Configuration Language

The IEC 61850 compliant substation uses the Substation Configuration Language (SCL) to exchange configuration between system configuration tools responsible for substation system configuration (Brunner, 2008).

2.2.3.2 Communication services

For remote monitoring and control operation, the communication standard employs Ethernet client server communication which is based on Manufacturing Message Specification (MMS) over Transmission Control Protocol (TCP/IP). The communication architecture is utilized extensively to establish and maintain time critical information exchange between substation Intelligent Electronic Devices (IED) (Goraj, 2010). The client server architecture consists of two communication services:

- Sample Values (SV)
- Generic Object Oriented System Event (GOOSE)

Within substation traffic, both SV and GOOSE require real-time intranet performance supported by Ethertypes to identify GOOSE and SV data packets exclusively. Restricting SV and GOOSE Ethertype data to specific ports and network segments creates logical sub networks or Virtual Local Area Networks (VLAN) which efficiently manage the network load. Applying priority tagging, fast track forwarding of high priority packets enable further optimization of network communication efficiency (Andersson et al., 2003) and (Clavel et al., 2015).

2.2.3.3 Communication between IEDs

The IEC 61850 Substation Automation System(SAS) functions are sub-divided into Logical Nodes (LN) or sub-functions that exchange data with other logical entities (Kostic et al., 2005). LNs are considered a virtual representation of physical IEDs (IEEE Power Engineering Society, 2009).

Abstract definition of the IEC 61850 enables communication independent modelling, which facilitates use of different protocols. Protection, monitoring, and control functions in the SAS result from instances of different LNs (Apostolov et al., 2003). Figure 2.2 illustrates an object model for a distance protection relay (Brunner, 2008). LNs represent objects with attributes and operations. Objects represent class instances which describe respective properties and characteristics of the object. Part 7-2 of the IEC 61850 communication standard specifies definition of these class models (IEEE Power Engineering Society, 2009).

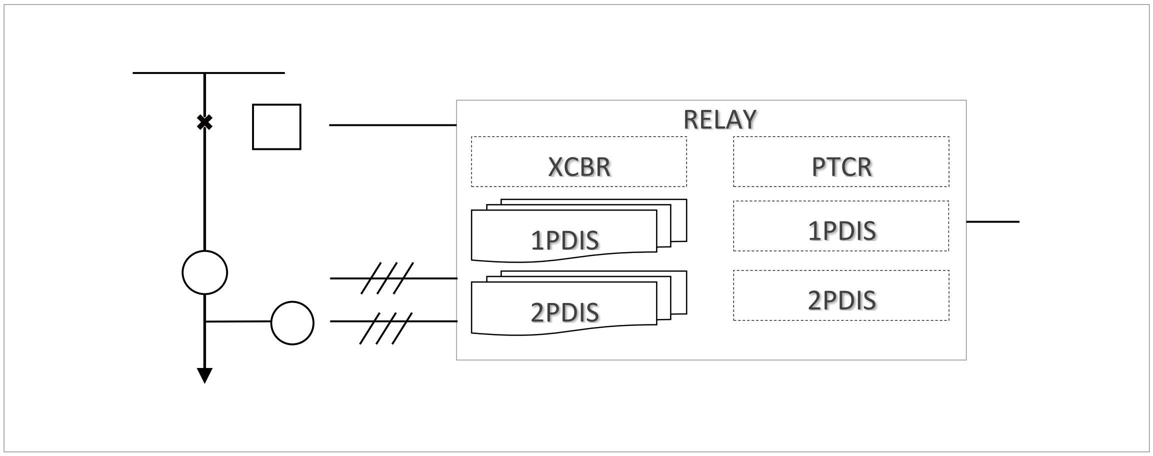


Figure 2.2: Distance protection (PDIS) object model (Brunner, 2008)

Figure 2.2 illustrates a distance protection (PDIS) LN for a relay based on the IEC 61850 data model. 1PDIS and 2PDIS represent instances of two separate zones where the Protection Trip Conditioning (PTRC) LN administers logical tripping of the circuit breaker LN (XCBR) based on the PDIS protection functions. LNs have two framework forms defined by the IEC 61850:

1. Logical node zero (LLN0)
2. Physical device logical node (LPDLN)

LLN0 facilitates hardware information of the Logical Device (LD) while the LPDLN facilitates data access of the LD (Kostic et al., 2005). LLN0 attributes and operations also include;

- GOOSE-Control-Blocks (GoCBs)
- Setting-Group-Control-Block (SGCB)
- Unicast-Sampled-Value-Control-Blocks (USVCBs)
- Multicast-Sampled-Value-Control-Blocks (MSVCBs)
- Generic Substation State Events (GSSE)-Control-Blocks (GsCBs)

Object references are formed through linking all object names that consist of instance whole-path names, identifying each instance uniquely (IEEE Power Engineering Society, 2009). Object operations characterize object services and are therefore considered as class interfaces (Walter & Kenrick, 2005). Figure 2.3 illustrates logical nodes participating in a distance protection function (Andersson et al., 2003). In Figure 2.3 the current transformer LN (TCTR) and voltage transformer (TVTR) interface measurement of current and voltages values.

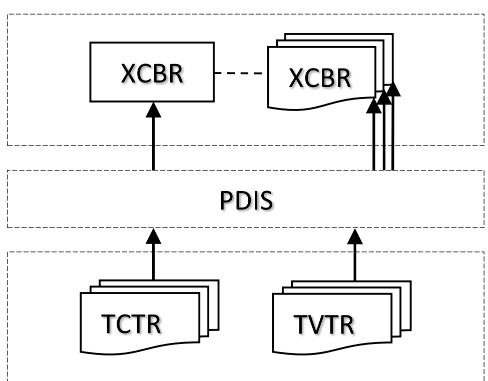


Figure 2.3: PDIS Logical node interaction (Andersson et al., 2003)

The IEC 61850 object models and services are mapped onto a communication stack that provides usable data models and services (Reilly, 2004).

2.2.3.4 IEC 61850 process bus

Figure 2.4 illustrates how the IEC 61850 process bus effectively divides the substation into a station level, bay level and process level completely decoupling the bay level equipment from the process level. In the process bus communication interface, SV transmission facilitates the replacement of analog Current Transformer (CT) and Voltage Transformer (VT) signals with serial communication.

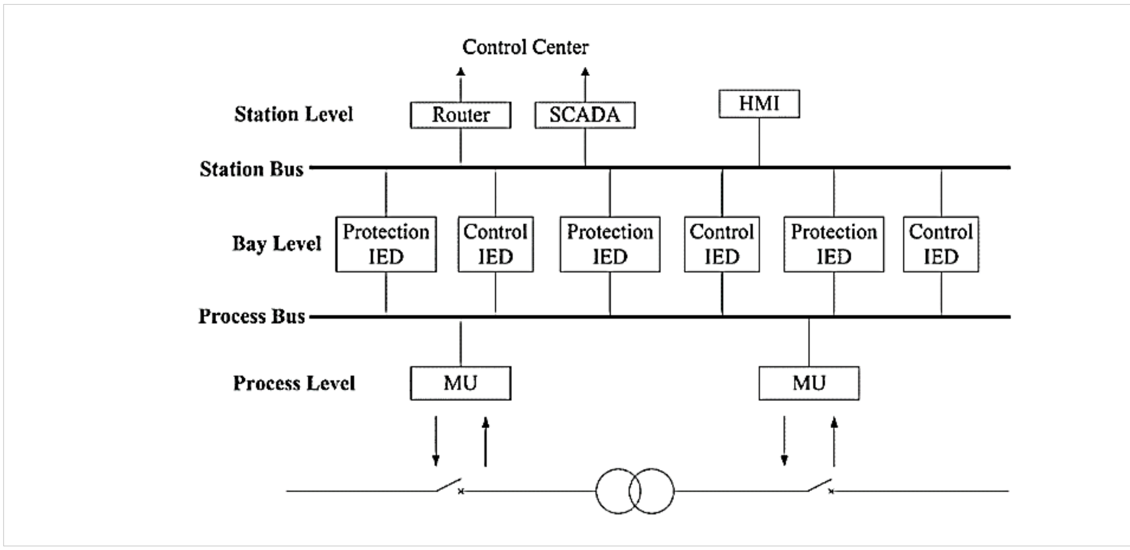


Figure 2.4: IEC 61850 Process Bus (IEEE Power Engineering Society, 2009)

2.2.4 IEC 61850 impact on Smart Grid

The characteristics of the IEC 61850 communication standard have positive and direct implications (Mackiewicz, 2004) on the Smart Grid, providing immense benefits that include;

Virtualized model:

The virtualized model enables definition of services, data and device functionality that characterize the protocols used to define data transmission over a network.

Devices Self-Description:

Client applications that communicate with IEC 61850 devices are able to download the description of all the data supported by the device from the device without any manual configuration of data objects or names.

Lower installation cost:

IEC 61850 enables data exchange over the substation network, eliminating the need for hard wire links for each relay, significantly reducing wiring costs. Utilizing network bandwidth for signals also eliminates construction costs, reducing requirements for trenching, ducts and construction of other communication medium support.

Lower commissioning cost:

Configuration and commission of IEC 61850 devices do not require extensive manual configuration in contrast to conventional and legacy devices. Client Server architecture and SCL utilize mapping to retrieve points lists directly from devices or importing via SCL files. Commissioning IEC 61850 devices primarily consists of setting up network addresses to establish communication, eliminating remarkable configuration and commissioning costs.

Lower equipment migration costs:

Interoperability and interchangeability completely eliminates proprietary prioritization. The IEC 61850 acutely defines externally aspects of physical devices in the substation minimizing cost for equipment migrations.

Lower integration and extension costs:

Low configuration, commission and migration costs of the IEC 61850 substation imply remarkable versatility. Through utilizing the same network, extending applications and adding devices into the substation system can be achieved with absolutely no ramifications substantially lowering integration and extension costs.

High-level services:

ACSI in the IEC 61850 enables support for wide variety of services that include GOOSE, GSSE, SMV, and logs.

VLANs and priority flags:

VLANs and priority flags for GOOSE and SV facilitate more efficient use of Ethernet utilizing switches.

Explicit procurement:

SCL can be used to accurately define substation device requirements through providing device specification for suppliers.

Table 2.2 presents a tabulated literature review synopsis of the IEC 61850 communication standard for electrical substation automation systems and its application.

Figure 2.5 provides an overview of the transmission line and differential protection literature reviewed.

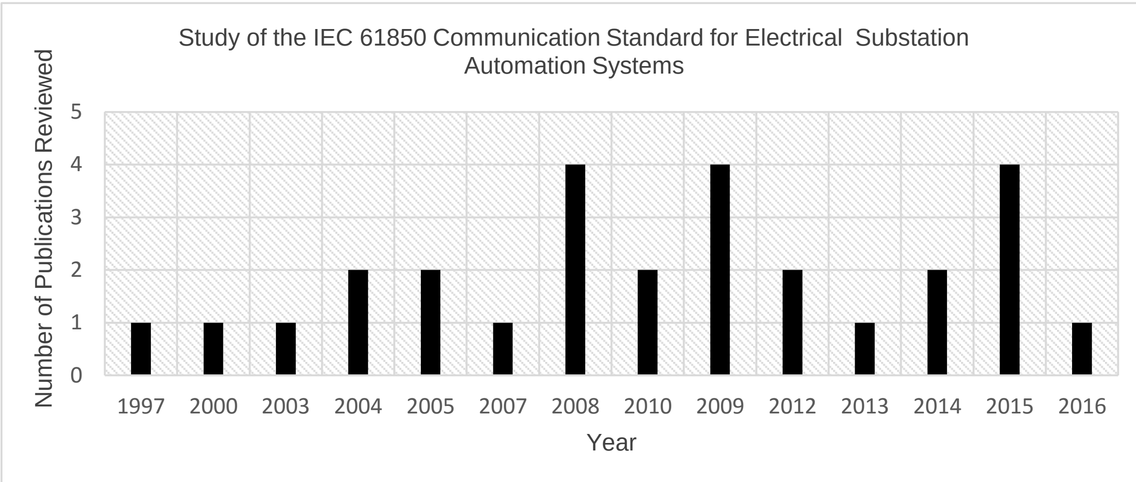


Figure 2.5: Study of the IEC 61850 communication standard for electrical substation automation systems

Twenty seven IEEE publications between 1997 and 2016 introducing the IEC 61850 communication standard for electrical substation automation systems are reviewed. All ten of the IEC 61850 standard documents are also reviewed comprehensively.

IEC 61850 introduces a substation structure that mitigates communication migration from the analog to digital paradigm, standardizing the communication architecture and systematically specifying services. The migration from analog to digital communication models presents a significant cost reduction substantiating the economic impact of the standard. The real-time communication Ethernet media also provides inimitable benefits that increase the standards overall reliability.

The IEC 61850 protocol forms groundwork intended for the communication evolution for electrical substation automation systems. To facilitate a future proof solution, the IEC 61850 standard is used to outline structural guidelines for developing the PMU-based transmission line current differential back-up protection scheme capable of inter-operability and interchangeability.

2.2.5 Review and discussion on IEC 61850 communication standard

Table 2.2: Literature review on the IEC 61850 communication standard for electrical substation automation systems and its application

Paper	Aim Of Paper	Methodology	Benefits / Drawbacks	Achievements
(Hanno Georg, Nils Dorsch, 2013)	Presents a real-time high performance evaluation model.	Focusing on substation automation at bay level to establish a simulation model and evaluate an analytical approach on basis of Network Calculus to identify worst case boundaries for intra-substation communication.	Real-time communication delays less that 10ms are established for both simulation an analytical models verifying adherence to IEC 61850 performance requirement specification.	Results show Network Calculus is capable of determining upper delay bounds.
(Arnold et al., 2014)	The paper presents performance evaluation of devices based on the IEC 61850 with respect: • speed • security • dependability	Implementation of multivendor Intelligent Electronic Devices (IDEs) for Permissive Overreaching Transfer Trip (POTT) communication scheme with IEC 61850 Generic Object Oriented Substation Events (GOOSE) messages.	The paper provides industrial reference for adopting IEC 61850 and IEC 61850 compliant IEDs devices into existing networks.	Experimental results demonstrate dependability and security characteristics' of a POTT communication scheme based on the IEC 61850 standard. In contrast to conventional POTT communication schemes, faster operating time is established.
(Elgargouri & Elmusrati, 2015)	This paper discusses security requirements of the IEC 61850 communication standard.	Literature review of factors pertaining to threats from the internet considering mapping to Transmission Control Protocol TCP/IP stack and the duration of handshake process.	Establishing address protection for smart grid cyber operations from unauthorized access though mapping the IEC 61850 over Ethernet stack and its' MAC.	Validating smart grid cyber requirements fulfilled by the IEC 61850.

Paper	Aim Of Paper	Methodology	Benefits / Drawbacks	Achievements
(Khavnekar et al., 2015)	The paper review and outlines development of communication development.	Comparative analysis between the Edition I and Edition II of the IEC 61850 communication standard for electrical substation automation systems.	Edition-II of the IEC 61850 introduces network redundancy, boosting communication reliability. It also introduces security mechanisms extending data models to expand the scope of the standard.	Verifying development in substation communication and advancement in security mechanisms.
(Yang & Vyatkin, 2015)	The paper proposes enabling automatic generation of IEC 61499 control systems from IEC 61850 specifications.	The proposed method is used to represent an IEC 61850 protection scheme graphically, integrating it with the IEC 61850 ontology developed previously.	IEC 61850 ontology development with LN interaction and message passing.	The proposed method converts protection scheme message passing information to a graph which is converted to an ontology.
(Yoo et al., 2016)	The paper proposes a (DER) network interface for improving the data transfer performance in IEC 61850 servers.	Analyses data acquisition, processing and transfer performance between conventional architecture and the proposed architecture.	Simple and reliable data transfer performance improvement of legacy DER network communication interface	Overall data transfer performance is been improved 63.5% on average

2.3 Transmission line Protection

2.3.1 Transmission lines

In electrical distribution networks, transmission lines play an essential and extremely important role. Typical transmission lines transmit three-phase high-voltage alternating current (AC) which is normally above 115 kV to compensate and reduce energy losses over long distance transmission. There are two principal types of transmission line systems:

1. Higher voltage transmission line systems which range above 230kV and
2. Lower voltage transmission line systems which typically start from 115kV

High-voltage direct-current (HVDC) technology is utilized to establish greater efficiency over longer distances which may be well over hundreds of kilometers. HVDC is also used to connect power between grids that are not mutually synchronized to increase stability over large distribution networks and avert cascading failures (Buldyrev et al., 2010). Regional and national inter connection of electric transmission networks facilitates redundancy which enables alternative routes for power flow increasing the grids overall dependability.

Figure 2.6 illustrates the electric power transmission service, providing power transmission pathways from generation to substation and consumer loads

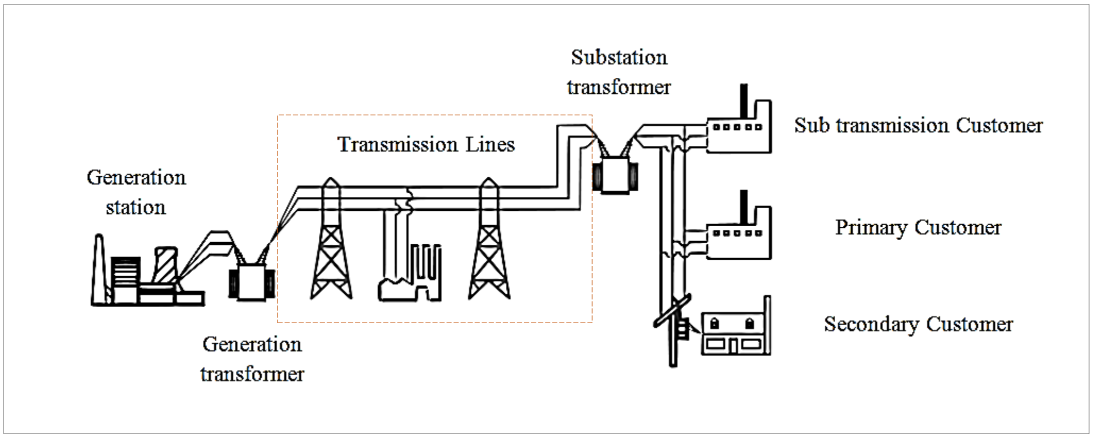


Figure 2.6: Electrical Distribution Network

Eskom's electric power distribution encompasses both lower and higher voltage transmission line systems (Eskom, 2007).

Power is typically transmitted using overhead power lines. Specification for transmission structures are determined by the transmission routes and terrain characteristics. In general, transmission line structures consist of steel lattice towers, wooden H-Frames or single-pole steels.

2.3.1.1 Transmission line Protection

In the occurrence of a fault, system disturbance may cause widespread outages within the interconnected transmission network. Considering potential culminating cascades, line protection in transmission system networks is necessary to identify, locate, isolate and clear faults within the transmission network. The higher level factors that directly influence line protection include (General Electric, 2007);

- Line Load
- Line Length
- Line Configuration
- Line Communication
- The System Feeding the Line
- Protection equipment and respective operation
- Fault clearing (time requirements for system stability)

The conventional transmission line protection methods are listed below (Idaho Power, 2011).

a. Overcurrent
b. Distance (Impedance)
c. Directional Overcurrent
d. Line Current Differential
e. Pilot:
    1) DCB (Directional Comparison Blocking
    2) POTT (Permissive Overreaching Transfer Trip)

Review of transmission line protection is limited to current differential protection as it is at the core of the research and is covered in great detail in the next sub-section.

2.3.2 Principles and classification of differential protection

Differential balance protection is based on Kirchhoff's current law which compares the sum of the incoming and outgoing currents.

Although differential protection concepts have essentially been limited to power transformers and generators recent development has extended differential balance protection concepts to transmission lines (J. Skea, D. Anderson, T. Green, R. Gross, 2007), (Mahanty & Gupta, 2004), (Kasztenny & Finney, 2005) and (Nengling et al., 2006). Differential balance protection systems can be broadly classified as unit and non-unit protection systems where unit systems are bound by Current Transformer (CT) location. Differential protection is classified under two main types;

2.3.2.1 Voltage balance

Figure 2.7 Illustrates the principle of differential protection based on balanced voltage principle.

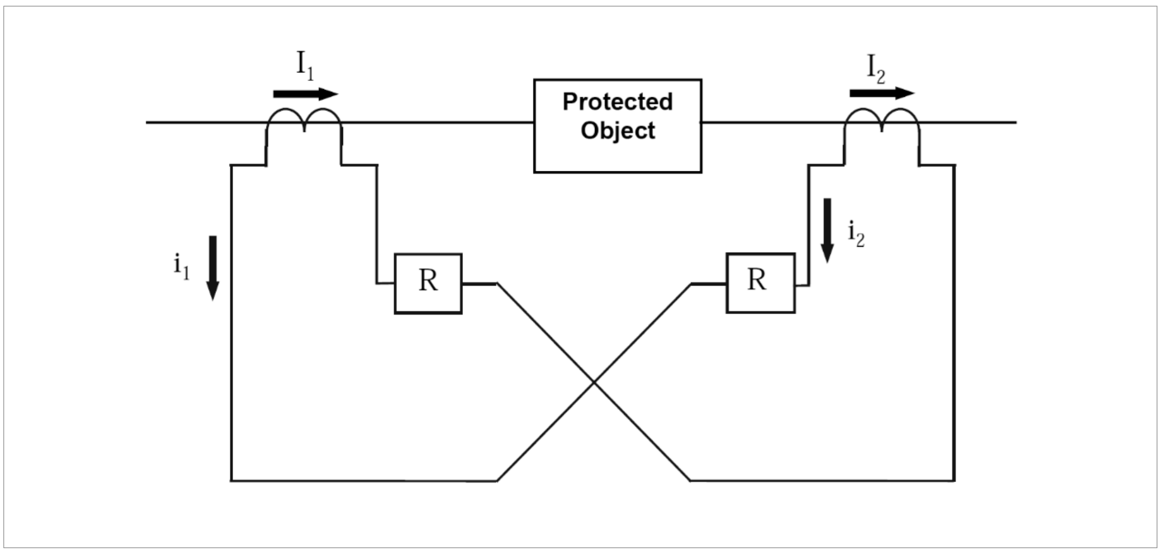


Figure 2.7: Differential protection based on balance voltage protection

As illustrated in Figure 2.7 the CT secondaries are connected to facilitate a mode under which for normal conditions and through fault conditions, the secondary currents on the opposite ends cancel out creating a voltage balance. The CTs utilize an air gap core to prevent saturation and over-voltages that may occur during zero secondary currents.

Differential relay coils (R) in Figure 2.7 are inserted in the loop. When an internal fault occurs in the equipment under protection, the voltages are no longer balanced causing current flowing through the relay coil that subsequently trips circuit breaker.

2.3.2.2 Current balance

Leading development in digital relays with advancement in communication and networking technologies facilitates application of the current balance differential concept onto transmission lines (Wheatley, 1985), (Albrecht et al., 1992) and (Xu et al., 2007). Current differential protection provides an outstanding solution for unit protection, providing selectivity within and between the protection schemes which limits fault effects within the network while providing immunity against power swings.

Digital line current differential relays are installed at ends of a transmission line. In a two-ended transmission line, the relays are placed at the local and remote current entry ports of the line. Both relays continuously exchange data over a digital communication channel with frequency and data type governed by the implemented algorithm. In the two-ended transmission line, the relays are typically identical, with each relay connected to a local CT, implementing the same differential algorithm. Figure 2.8 illustrates single line representation of the two-ended transmission with specific location of the differential elements including the communication channel.

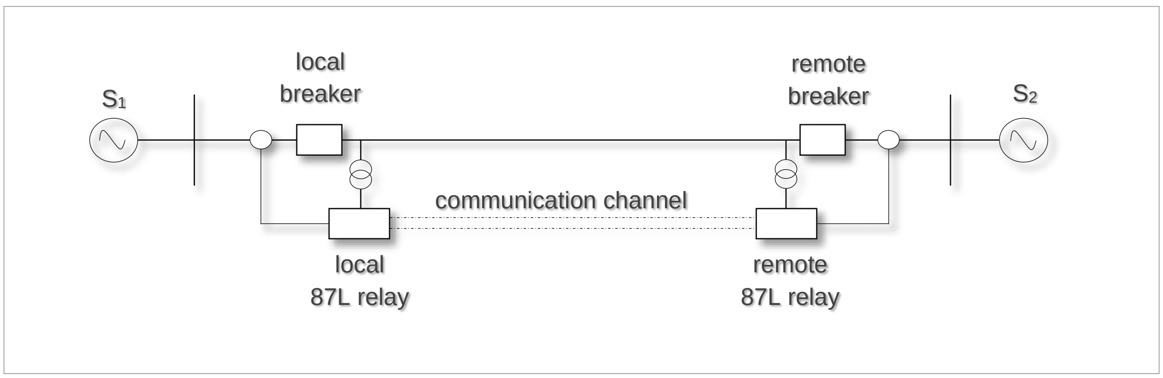


Figure 2.8: Two terminal transmission line schematic

2.3.3 Differential relaying

With reference to Figure 2.9 (Kasztenny & Fischer, 2010), a line current differential protection system consists of identical local and remote end relays (depicted Relay 1 and Relay 2 respectively) that operate independently, connected over a digital communication channel. Each of the digital relays sample analog input currents utilizing an analog to digital (A/D) converter to obtain respective digital representation.

The sampling rates depend on the algorithm design, varying from a few kilohertz to less than twenty samples per cycle. Although line differential current functions utilize a range of low sample rates, high resolution and rate is maintained. This is because the same digital data facilitates to utilize other local functions which may include;

- metering
- fault recording
- breaker failure protection
- other protection schemes

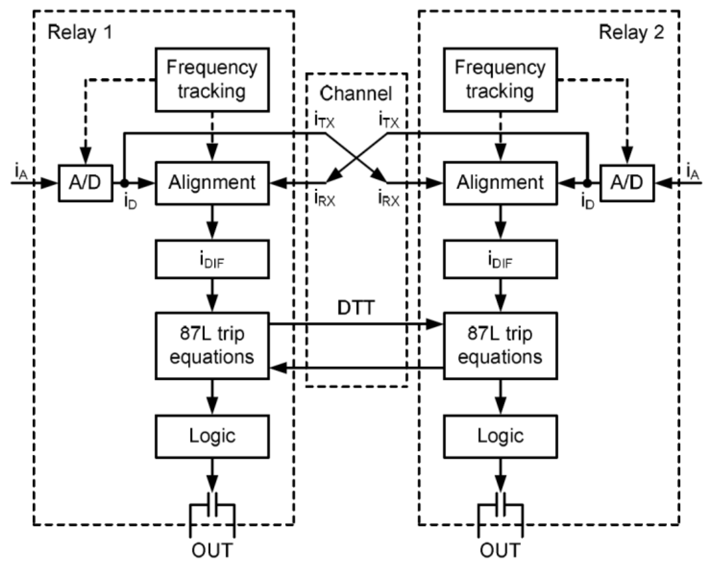


Figure 2.9: Transmission line differential relay, Direct Trip Transfer (DTT) (Kasztenny & Fischer, 2010)

The samples are obtained synchronously, typically maintaining a constant sampling rate, and tracking the power system frequency.

2.3.4 Differential line communication

Communication between relays in a two-ended transmission line facilitates the current differential function. When a full data set is communicated, the receiving relay aligns the data (Miller et al., 2010a). The differential trip equations are then subsequently run based on the dataset received.

The receiving relay is typically configured to operate in master mode while the relay that strictly publishes the dataset is typically configured to operate in slave mode. Both relays operate autonomously using the DTT to enable issuing of trip commands to breakers local to slave mode relays (Tholomier et al., 2008). Figure 2.10 illustrates a communication-based protection scheme whose decisive protection function is based on synchronous sampling of the analog current values acquired at the line ends.

Figure 2.10: IEC 61850 based channel communication (IEEE Power Engineering Society, 2009)

Channel specifications and bandwidth:

The two conventional communication channel designs in line current differential protection include the multiplexed connection and point to point fiber.

2.3.4.1 Multiplexed connection

This channel design is more common for line current differential protection applications, utilizing a bandwidth typically limited to 64 kbps. Multiplexed channels utilize SONET/SDH infrastructure. The channel design is inapt for long haul commission, necessitating integration of third party devices between differential communicating elements in the line. This introduces complexity and third party device failure modes which need to be accounted for in the protection design (Kasztenny & Fischer, 2010). To mitigate potential failure, the line current differential relays are further designed to incorporate third party device failure modes.

2.3.4.2 Direct point to point fiber

Direct point to point fiber is superior to multiplexed connections for critical or long haul commission applications.

The channels' design supports longer distances, eliminating need for amplifiers and amplifier related infrastructure. Direct point to point does not utilize any third party devices between communicating relays (Kasztenny & Fischer, 2010), facilitating a communication channel that is inherently symmetrical, confining communication impairments strictly to the relays and the channel medium itself.

Conventional relays provide redundancy, with channels dedicated for both fiber connection, and multiplexed connection, typically using the multiplexed connection as a standby channel. Communication channel designs vary considerably but maintain a 64kbps bandwidth that restrict larger amounts of data exchange.

Sampling:

Typical quantities relays exchange over the channel are:

i. Current samples

Precise current sampling and transmission of sample data, utilize interpolation rates enough for accurate interpolation. This enables data alignment through measuring data latency between the remote and local relays, interpolating the remote current samples to align them with the local samples different phase angles with transient errors (Wang, 2010). Interpolating also enables frequency tracking. With current sampling, relay dedicated clocks do not require any synchronization.

ii. Phasors

Phasors require more bandwidth to transmit real and imaginary parts of the currents, subsequently, phasor exchange rates cannot facilitate interpolation limiting data alignment to sampling clocks. Phasors also require control of the relay sampling clocks to track system frequency(Wang, 2010).

Synchronization:

Precise operation of the differential function depends on synchronous sampling from both relays in the line. Data alignment or synchronization is critical in line current differential protection. The conventional data alignment methods are outlined here.

a. Ping-Pong

The Ping-Pong algorithm employs estimation of clock offset between two line relays working over a communication channel (Liu et al., 2011). The algorithm measures travel time, time-stamping transmissions and exchanging line information. Assuming channel symmetry, the total channel offset time is utilized to align the data.

b. Clock offset control:

The clock offset control method utilizes the measured clock offset to control the local sampling time at both ends. This ensures synchronization by compensating for and cancelling out the time offset (Liu et al., 2011).

c. Local time time-stamp:

Time-stamping transmitted data with local time enables calculation of clock differences between the two relays, which in turn facilitates re-sampling of the data in order to align time instances.

Although the GPS presents a practical solution for central timing across wide spread areas, dependence on satellite systems and other external timing devices has been less preferred.

Channel symmetry and synchronization are indeed critical as both dependability and security are absolutely dependent on alignment. Misalignment causes inaccuracy, and different phase angles with transient errors (Wang, 2010). Immunity to shifts is therefore necessary to achieve accurate function.

2.3.5 Current differential protection algorithm:

Power quality on transmission lines is subject to accurate high speed fault clearing that improves transient stability. Fault clearing is a crucial protection function that has driven realization of efficient and effective protection algorithms. Differential algorithms utilize line terminal fault current, comparatively analyzing current between line-end relays of a transmission line. Figure 2.11 illustrates subsequent operations to evaluate differential current and trigger trip logic.

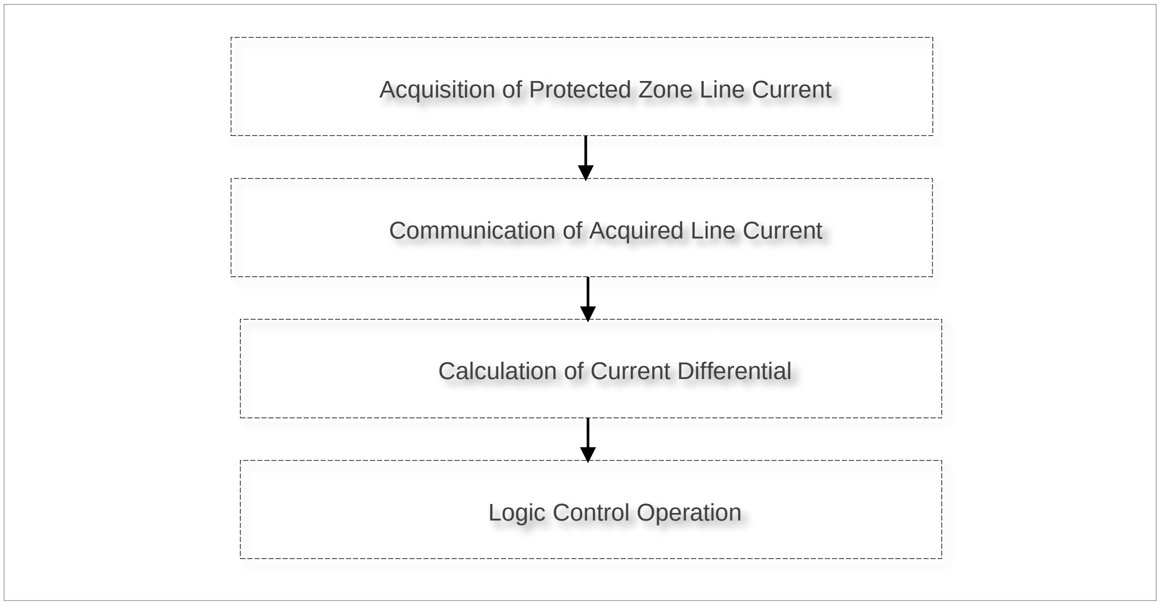


Figure 2.11: Current differential modular process

The calculated differential current within a protected zone is theoretically zero. This is based on the premise that current flow in and out of a protected zone is equal. However in practice, the calculated differential current is comparatively monitored against a pre-set threshold, having the protection function operate when the difference surpasses marginal value. This characteristic is discussed further below.

2.3.6 Restraint and stability characteristic

The differential algorithm calculates the current differential from the currents measured at local and remote current entry ports of the protected zone as shown in Figure 2.12 (Liu et al., 2011). The current differential is proportional to the fault current and theoretically nears zero under no fault conditions. However, practical application of differential relaying differs from theoretic principles as non-zero differentials occur. Non-zero differentials that occur are factored in the algorithm model to prevent incorrect system response and erroneous protection operation.

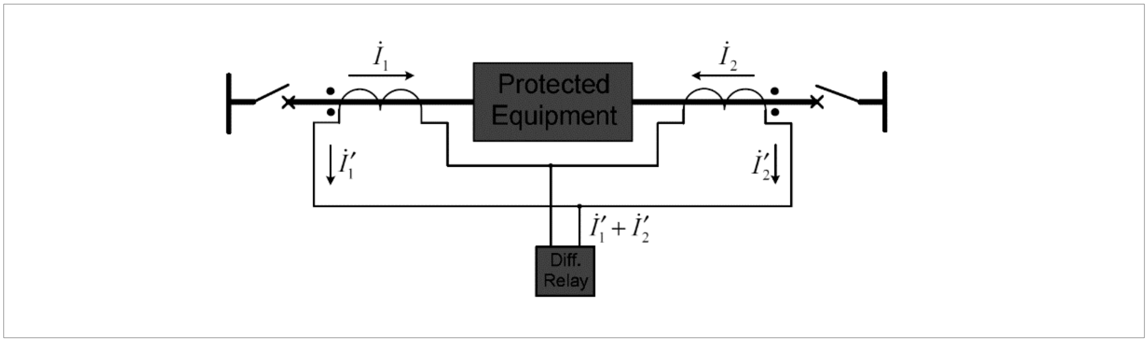


Figure 2.12: Differential relaying (Liu et al., 2011)

The operating current is progressively correlated with a preset restraint current.

$$I_{\text{DIFF}} = |\,\dot{I_1}' + \dot{I_2}'| \tag{2.1}$$

$$I_{\text{RES}} = (|\dot{I_1}'| + |\dot{I_2}'|)/2 \tag{2.2}$$

The current differential and restraint current are given by Equations 2.1 and 2.2 respectively. Figure 2.13 (Jorge et al., 2006) illustrates the restraint and stability characteristic.

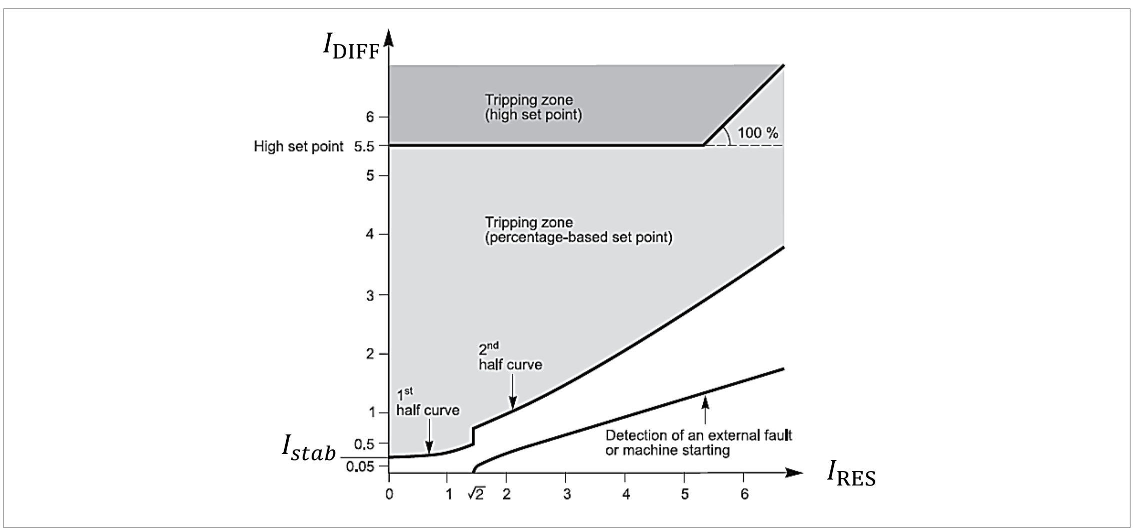


Figure 2.13: Restraint / Stability Characteristic plot (Jorge et al., 2006)

$$I_{\text{BIAS}} = I_{\text{RES}} = I_{\text{STAB}} \tag{2.3}$$

Where;

I_{BIAS} : Bias current
I_{RES} : Restraint current
I_{STAB} : Stability current
I_{DIFF} : Differential current

While the protection scheme employs fundamental algebra, protective control functions remain susceptible to false positives with regard to currents that may occur during external fault conditions. Under no fault conditions, current into the protected zone is equal to the current proceeding out of the protected zone. Under fault conditions, the differential current is used to prompt the protective system response.

To mitigate inaccurate system response that may be caused by contrived current differentials, line terminal restrain is applied to the algorithms' operating current. Comparatively monitoring the differential current against a pre-set threshold would restrict the protective control functions to strictly operate when the difference surpasses marginal value. This safe guards and prevents inaccurate differential operation, offering a rational approach and compensating for non-zero current differential values.

Employing a restraint current and raising the relay setting in proportion to the through fault current is an excellent way to avert erroneous protection functions. The CT currents in the protected zone are summed. The summation represents a measure of the loading of the primary system. It is worth mentioning that precise values and formulas for the summation are proprietary. For internal fault circuit breaker trip, the summed system load current and the calculated differential current are utilized to distinguish between internal and external faults (Rintamaki & Ylinen, 2008). Table 2.3 presents a tabulated literature review synopsis of transmission line system protection and differential principles.

2.3.7 Review and discussion on existing papers on line differential protection

Table 2.3: Line differential protection literature review

Paper	Aim Of Paper	Methodology	Hardware / Software Utilized	Benefits / Drawbacks	Achievements
(Tholomier et al., 2008)	Comparison of : Line differential protection to Directional comparison protection outlining the advantages and disadvantages of each	Comparative performance analysis of both differential and directional comparison functions	Software: (Virtual Hardware) • Fault model • Breaker model • Network model	Highlighting protection function key benefits and deficiencies for both directional comparison and line differential protection schemes	Outlining the (delta) directional comparison scheme having more benefits in contrast to the line differential protection scheme
(Pires & Guerreiro, 2008)	Presents a new approach on current differential protection for transmission lines	Park transformation is utilized to transform the three phase quantities into a synchronous rotating reference frame	Software: Mat Lab Simulink	Applying the concept of transforming three phase line currents into DC components removes sampling misalignment and improving overall time delay	The new proposed approach was analyzed and the preliminary results evidence effectiveness and notable performance improvements
(N. Zhang et al., 2008)	Presents a new integrated protection scheme for transmission lines based on current differential protection techniques	Applying specially designed relays which implement current differential algorithms with multiple settings to cover all protection line settings	Software: Virtual Protection Scheme	Establishing grounds for further research and development on a future proof protection scheme solution	The proposed protection scheme offers a number of significant advantages over conventional approaches including cost effectiveness
(Dambhare et al., 2009)	Proposes a methodology for adaptive control of the restraining region in a current differential plane	Extending the methodology for protection of series compensated transmission lines		Proposed methodology enhances sensitivity and relaying speed without compromising the security of the protection system	The proposed adaptive control of restrain region together with phasorlet algorithm for phasor estimation provides the best solution for current differential protection of transmission lines

Paper	Aim Of Paper	Methodology	Benefits / Drawbacks	Achievements
(Miller et al., 2010a)	Reviewing technical solutions to the line current differential design and application	Highlighting and addressing common design constraints and utility driven needs.	Outlining a procedural tutorial for application of protection principles together with communication and signal processing	Presenting design directions for line current differential protection
(Liu et al., 2011)	Reviewing traditional technical solutions	Highlighting and addressing technical solutions to the transmission line current differential protection	New technologies are introduced into the proposed protection solution such as the IEEE 1588 time synchronization algorithm	The paper outlines the architecture with ratio differential criterion while presenting a novel, next generation protection scheme based on the IEC61850 standard
(Unde & Dambhare, 2011)	Exploring a current differential protection scheme for mutually coupled lines considering line charging current	Utilizing the GPS for synchronized current and voltage measurements, developing the differential function using the equivalent-π model for transmission line and phase co-ordinate approach	The scheme works for double fault and does no mal-operate for external disturbance like in line switching	The proposed scheme is immune to the mutual coupling effect and discriminates the external and internal faults with high fault resistance sensitivity
(Gartia et al., 2013)	Presenting a differential protection scheme for transmission lines	Focusing on the design of a differential protection scheme that is microcontroller based, using fiber optic communication	The developed method would be a useful tool for wide area monitoring and control even more so with growing research and development in PMUs	Providing a fail-safe method of differential protection that may replace conventional methods
(Yuan et al., 2015)	Presenting a solution for the deterioration of protection of transmission lines using differential current protection as the voltage levels and the line length increase	The method primarily utilizes single Phase-Locked Loop (PLL) to track phase of the corresponding voltages then transforming to rotating coordinates through transformation matrix	Proposed method provides high sensitivity and strong capacity of enduring transition resistance	Provided method has merits of effectiveness, robustness and rapidity

Paper	Aim Of Paper	Methodology	Benefits / Drawbacks	Achievements
(J. Zhang et al., 2008)	To study and solve temperature excursion and operating stability problems	Application of the Adaptive Optical Transducing Principle and advanced study of the Solenoid Collecting Magnetic Field Optical Path	The research utilizes a Bergeron model-based algorithm to resolve current capacitive influence on differential protection	Simulation results validate the reliability and accuracy of designed algorithm
(Darwish et al., 2009)	The paper presents a study on evaluation of a power differential relay	Local computation of relay input signals, active and reactive powers from complex product of the current and voltage phasors at each end on a sample by sample basis to examine relay settings under fault conditions	Tests and performance evaluations are done on a physical model	Formulating a refined methodology to outline relay parameters and relay ratings
(Rintamaki & Ylinen, 2008)	The paper presents an alternate protection scheme as a solution for varying fault current conditions, magnitudes and contributions from several fault current sources	Application of unit type protection schemes that utilize differential protection	Native IEC 61850 relays in line differential protection schemes provide enhanced scheme speed, flexibility and reliability	Establishing that IEC 61850 based line differential protection schemes are future proof
(Pires et al., 2010)	The paper presents a new approach for current differential transmission lines	Application of the Clarke-Concordia transformation and principal component analysis	The Clarke-Concordia transformation yields typical patterns for each fault type making it possible to define the operational condition of different relay operations	The simulation results validate the proposed approach

Paper	Aim Of Paper	Methodology	Benefits / Drawbacks	Achievements
(dos Santos et al., 2010)	The paper presents a novel solution for protection of high voltage power cables	Implementing differential principles based on Rogowski Coil current sensor measurements	Field tests are performed in an industrial power system that serves a steel production plant	Test results validate high dependability and security of the solution protecting parallel cables that provide power to a 90 MV A electric arc furnace transformer
(Kasztenny & Fischer, 2010)	The paper reviews prevailing technical solutions to line current differential design and application, addressing common design constraints together with application requirements	Review of prevailing and existing methodologies in line current differential protection schemes	The reviewed solution include Alpha Plane differential trip equations that enhance the original concept to multi-terminal lines with in-line transformers and line-charging current compensation	Formulating a comprehensive outline of typical design procedures for next generation line current differential protection
(Sivanagaraju et al., 2014)	The paper presents a method to estimate transmission line parameters using phasor measurements from PMUs together with uncertainties considering the measurement inaccuracies	Utilizing synchrophasors from PMUs to estimate uncertainties in differential line transmission systems	Conventional measurements are presented along with the PMU measurements	A new current differential protection scheme utilizing phasor measurements at both line ends is realized
(Xingguo Wang et al., 2015)	The paper proposes current differential protection based on virtual restraint current	Utilizing a superior criterion for internal faults and external faults to optimize reliability	The protection criterion proposed, accommodates current inversion of series compensated transmission	Critical faults introduced in the 500 Kv RTDS power system simulation are cleared

Figure 2.14 provides an overview of the transmission line and differential protection literature reviewed.

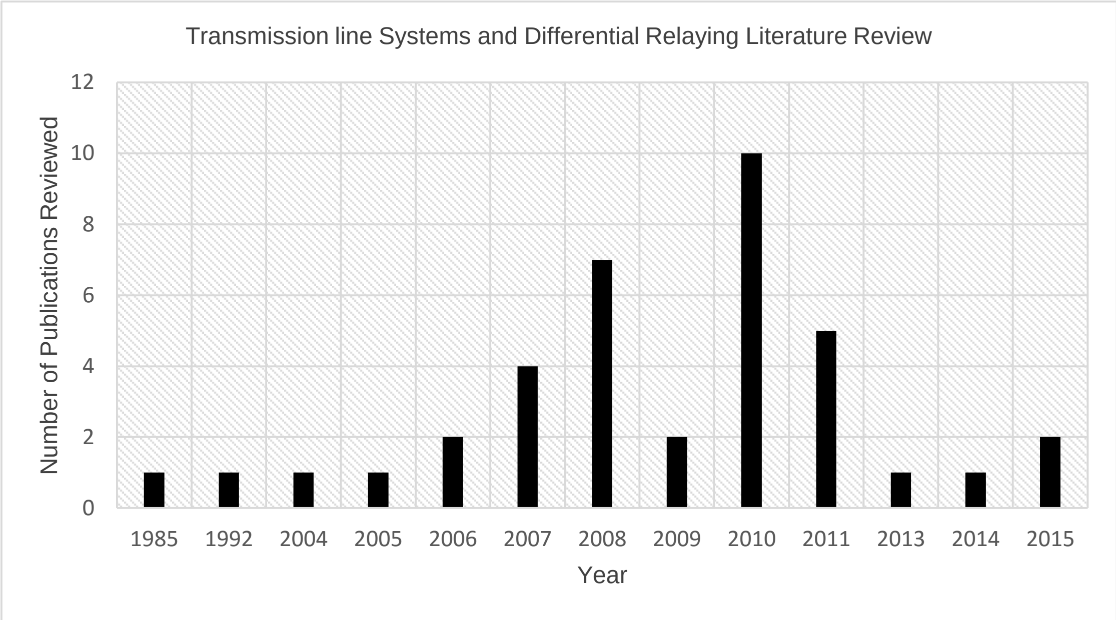


Figure 2.14: Transmission line systems and differential Relaying Literature Review

Between 1985 and 2015 thirty eight IEEE publications focusing on transmission line systems and line protection based on differential relaying, are reviewed. A general distribution model of the flagship journals reviewed is presented.

The literature review outlines prevailing transmission line protection techniques, objectively emphasizing relative benefits and drawbacks for each of the protection methods. The review directs its main focus to differential protection principles which provide precise and selective unit protection.

Review of differential relaying concepts and prevailing methods provides awareness of a knowledge gap in using PMUs for differential transmission line protection. The voltage balance differential protection technique requires a multi-tap transformer to accurately balance the current between transformer pairs. Although this system is suitable for protection of relatively short length transmission lines it has the critical disadvantage of ineffectiveness over long transmission lines as the charging current operates the protective relay regardless of voltage balance.

Considering this disadvantage, the current differential method is selected over the voltage balance protection method in order to optimally utilize the benefits it holds over the voltage balance protection method. Some of the more essential benefits of a current balance protection scheme include (Tholomier et al., 2008);

- protection system ignores mutual coupling on double circuit lines
- protection system is no affected by current reversal on double circuit lines
- increased simplicity over large network with no need to process voltage data

Change in approach to concepts and underlying assumptions specified by the IEEE C37.238-2011 precision time protocol, qualify the hypothesis of implementing synchrophasors in differential protection schemes for transmission lines with a contribution that mitigates conventional drawbacks experienced in current differential relaying. Some of these drawbacks include;

- vulnerability to CT saturation
- vulnerability to charging current
- transmission propagation delays
- requirement for phase shift compensation
- sensitivity issues with high impedance faults
- vulnerability to communications channel impairments

2.4 Phasor measurement technology and PMU applications

This section presents a review of research and development in synchrophasor technology. The sub-sections include; a background and necessity for phasor measurements, an introduction to phasors, PMU principles, PMU communication and finally, impact of PMUs on Smart Grid protection applications.

2.4.1 Background

In real-time power systems, state estimation data provides estimations of power system; angles, currents, voltages, breaker status and Mega Watts (MW) / Megavars (MVAR) flows. The accuracy and precision of state estimation data depends on the measurement technology and framework of the power system model. Figure 2.15 (Wu & Giri, 2006) depicts the role of state estimation (Wood et al., 2013) illustrating the application and use of state estimation data which include:

- security enhancement
- optimization applications
- dynamic security analysis
- network contingency analysis

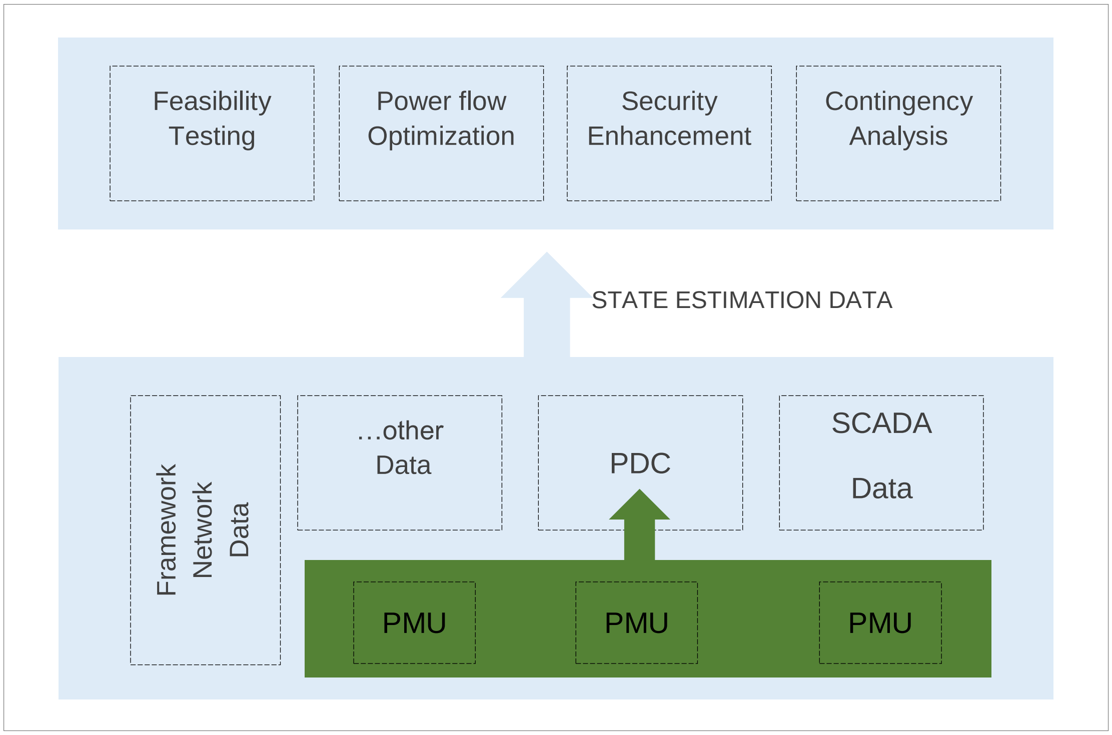


Figure 2.15: Role and importance of state estimation adopted from (Wu & Giri, 2006)

State estimation emphasizes critical data acquisition technology to optimize accuracy and precision of state estimation data. Until recently, supervisory control and data acquisition (SCADA) technology has been the prevailing state estimation technology. With SCADA, current and voltage transformers connected to the system provide state estimation data at fixed intervals (between two and six seconds). The data is then utilized for monitoring, protection and control of the system.

SCADA standards have proven insufficient for providing accurate state estimation critical for analyzing disturbance and post disturbance scenarios (Wu & Giri, 2006). The SCADA report rates thwart the energy management systems' ability to facilitate precise and accurate system response, unable to reflect real-time, dynamic characteristics of the system which remains susceptible to fast transients and unpredictable disturbances.

SCADA neglects synchronous sampling between substations which results in time skew errors. The measurement system only employs magnitude components, neglecting necessary substation bus angular components. This is yet another critical shortcoming of the SCADA system as angle separation provides excellent indication of system stress. The shortcomings of SCADA technology and call for superior data acquisition standards have led to the development of phasor measurement technology. Phasor measurements have become critical for Wide Area Measurement Systems (WAMS) that facilitate power system monitoring, protection, and control applications. Phasor Measuring units (PMUs) are intelligent electronic devices that provide synchronized measurements of real-time phasors that represent voltages or currents. Synchronization of the phasors is achieved through coherently time aligning samples using timing signals from the Global Positioning System (GPS) Satellite. Synchronized phasor measurements are better suited for power system monitoring, control, and protection (Hauer, 1996). Table 2.4 presents a comparison between the characteristics of the SCADA and PMU systems.

Table 2.4: Comparison between SCADA and PMUs systems

Attribute	SCADA	PMU
Measurement	Analog	Digital
Resolution	2 – 4 samples per cycle	Up to 60 samples per cycle
Observability	Steady State	Dynamic and Transient
Monitoring	Local	Wide Area
Phasor Angle Measurement	No	Yes

2.4.2 Phasors

Phasors are a complex number representation of sinusoids. In Figure 2.16, both the Cartesian co-ordinates (X, Y) in the complex plane and polar forms $X\,sin\,\emptyset$ and $X\,cos\,\emptyset$ are shown.

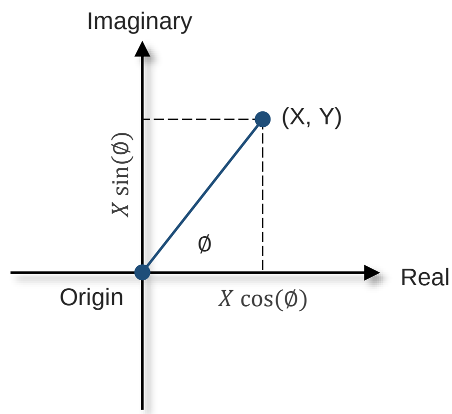


Figure 2.16: Two dimensional representation of a complex number

With phasors, only the real part $X\ cos\ \emptyset$ is desired and considered. The alternating current (AC) waveform presented in Equation 2.4, is commonly represented as the phasor shown in Equation 2.5 and can be mathematically represented in phasor notation as show in Equation 2.6 (IEEE Power Engineering Society, 2005).

$$x(t) = X_m\ cos(\ \omega t\ +\ \phi) \tag{2.4}$$

$$= X_r + jX_i \tag{2.5}$$

$$\bar{X} = X_m \angle\phi \tag{2.6}$$

Where;

X_m : Magnitude of the sinusoidal waveform

ω : $2.\pi.f$ where f is the instantaneous frequency

ϕ : Angular starting point for the waveform (phase angle)

$\bar{X}$: Phasor representation of $x(t)$

X_i : Imaginary part of rectangular component

X_r : Real part of rectangular component

2.4.2.1 Synchrophasor definition

The sinusoidal signal given in Equation 2.4 can be represented as a synchrophasor sinusoid as shown Equation 2.7. The corresponding sinusoid waveform is illustrated in Figure 2.17, with the magnitude and angle emphasized in Figure 2.18 (IEEE Power and Energy Society, 2011b).

$$x(t) = X_m \, cos(\omega t + \phi) = X_m \, cos(2\pi f_0 + \phi) \tag{2.7}$$

Where;

f_0 : Nominal angular system frequency (50 Hz)

ϕ : Phase angle relative to the cosine function Coordinated Universal Time (UTC) system frequency f_0

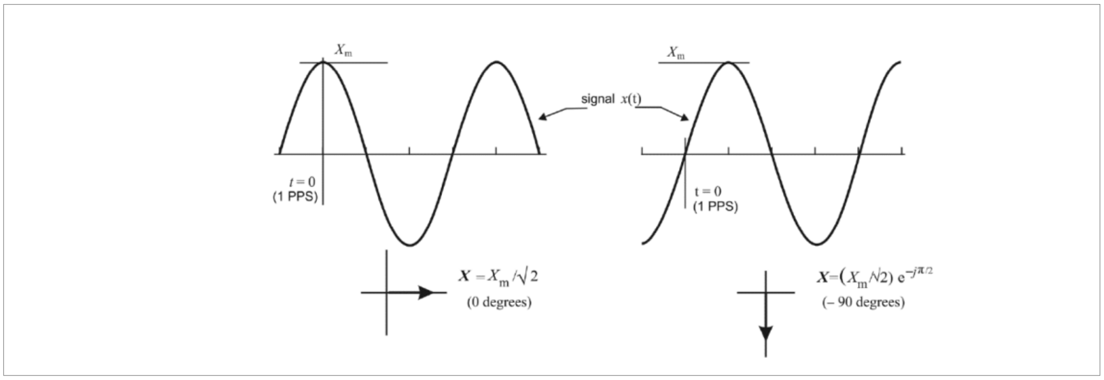


Figure 2.17: Convention for synchrophasor representation

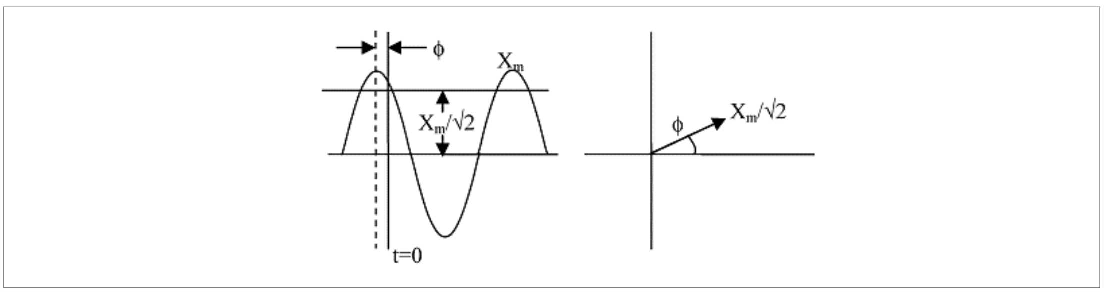


Figure 2.18: Phasor representation of an AC waveform (Singh et al., 2011)

For the general case where;

$X_m(t)$ = amplitude in the time domain

$f(t)$ = sinusoid frequency in time domain

g = difference between the nominal and actual frequency in time domain and considering a representation of difference between the nominal and actual frequency given in Equation 2.8 (IEEE Power and Energy Society, 2011b),

$$g(t) = f(t) - f_0 \tag{2.8}$$

The synchrophasor sinusoid is given by Equation 2.9

$$x(t) = X_m(t)\, cos(2\pi \int f\ (t)\, dt + \phi) \tag{2.9}$$

$$= X_m(t)\, cos(2\pi \int (f_0 + g(t))\, dt + \phi)$$

$$= X_m(t)\, cos(2\pi\, f_0(t) + (2\pi \int g(t) dt + \phi))$$

and can be represented as shown in Equation 2.10 (IEEE Power and Energy Society, 2011b),

$$X(t) = \left(X_m(t)/\sqrt{2}\right) e^{j(2\pi \int gt + \phi)} \tag{2.10}$$

2.4.2.2 Frequency measurement and measurement evaluation

For a synchrophasor sinusoid signal given by Equation 2.11 (IEEE Power and Energy Society, 2011b).

$$x(t) = X_m\, cos[\varphi(t)] \tag{2.11}$$

The PMU establishes a reporting or measuring frequency defined by Equation 2.12 (IEEE Power and Energy Society, 2011b).

$$f(t) = \frac{1 d\varphi}{2\pi\, dt}\ (t) \tag{2.12}$$

Where frequency error is denoted (FE), frequency measurement evaluation for absolute deviation between the theoretic values and estimated values is given in Hz and is shown in Equation 2.13 (IEEE Power and Energy Society, 2011b).

$$\mathrm{FE} = = |f_{true} - f_{measured}| = |\Delta f_{true} - \Delta f_{measured}| \tag{2.13}$$

2.4.2.3 Rate of Change of frequency measurement and measurement evaluation

For a synchrophasor sinusoid signal given by Equation 2.11.The PMU establishes the derivative of the reporting frequency according to the time defined by Equation 2.14 (IEEE Power and Energy Society, 2011b).

$$\mathrm{ROCOF}(t) = \frac{df}{dt}(t) \tag{2.14}$$

Where ROCOF error is denoted (RFE), frequency measurement evaluation for absolute deviation between the theoretic values and estimated values is given in Hz and Hz/s respectively and is shown in Equation 2.15 (IEEE Power and Energy Society, 2011b).

$$\mathrm{RFE} = = |(df/dt)_{true} - (df/dt)_{measured}| \tag{2.15}$$

For non-sinusoidal waveforms, the phasor is assumed to represent the respective frequency component. Synchronized measurements of adjacent bus voltage phasors facilitate computation of instantaneous power angles as illustrated in Figure 2.19 and Equation 2.8. The power angle is then used for analysis of the degree of stress between two substations.

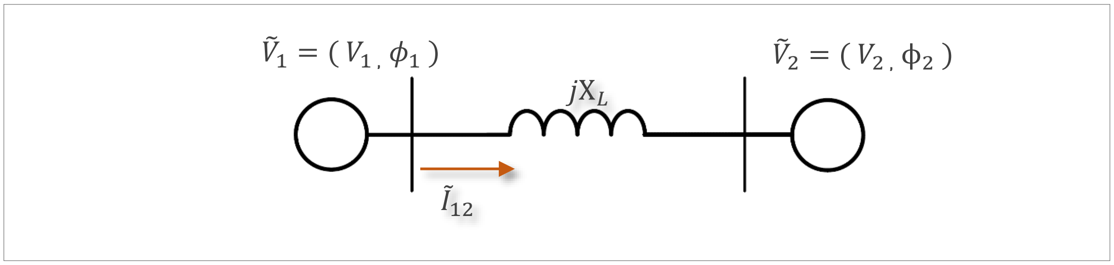


Figure 2.19: Adjacent Substations

$$\tilde{I}_{12} = \frac{\tilde{V}_1 - \tilde{V}_2}{jX_L} \tag{2.16}$$

$$P_{12} = \frac{V_1.V_2\, sin(\phi_1 - \phi_2)}{X_L} \tag{2.17}$$

Where;

P_{12} : Power transfer

V_1 : Voltage magnitude of substation one

V_2 : Voltage magnitude of substation two

ϕ_1 : Sub-station one phase angle component

ϕ_2 : Sub-station two phase angle component

2.4.3 PMU Principles

PMUs are capable of carrying out highly accurate and precise grid measurements at very high sampling rates, maintaining synchronous sampling based on GPS time signals to facilitate time-stamping of the sampled system values.

The GPS time signal is intercepted by a local satellite receiver which then provides an exact timing pulse utilized to tag sampled voltage and current inputs. Once sampled, positive sequence voltages and currents are calculated (IEEE Power Engineering Society, 2005) and there after tagged to establish absolute synchronicity with microsecond precision. Based on the IEEE 1344 standard (Huang et al., 2007) a message frame is then assembled and formatted from the time stamped phasor, which is then transmitted over a communication link. These synchronous positive sequence phasors are collected and may be exchanged between local protection and control IEDs. Figure 2.20 illustrates PMU function block diagram representation (IEEE Power and Energy Society, 2011b).

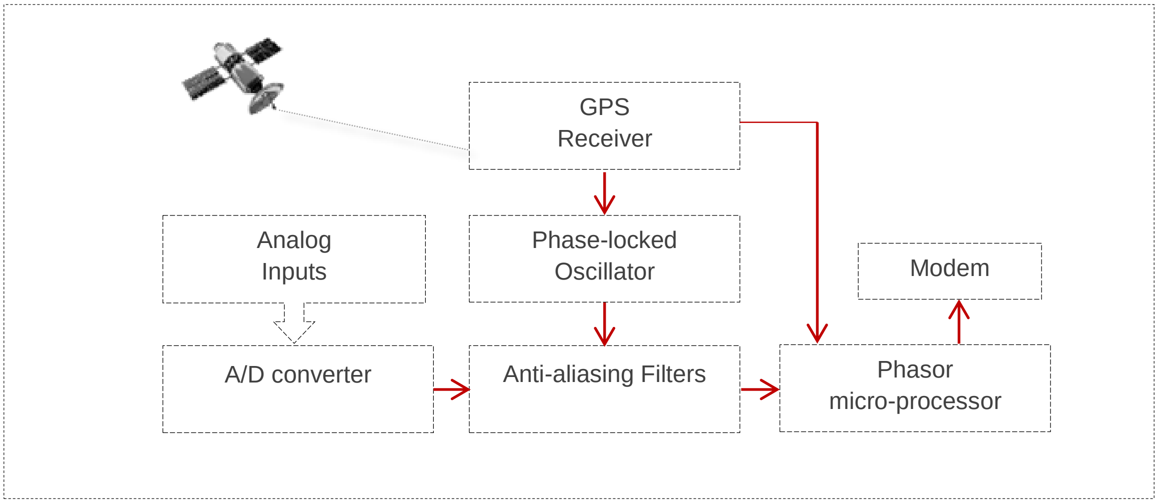


Figure 2.20: PMU modular design

2.4.4 PMU Communication

Measured PMU phasor data is transmitted at report rates presented in Table 2.5. Together with the phase-lock oscillator, the GPS reference provides high-speed synchronized sampling within microsecond accuracy, facilitating time tagged phasors which are transmitted between PMUs and Phasor Data Concentrators (PDCs) at report rates up to 60 samples per second.

Table 2.5: Phasor data report rates

System Frequency:	**50 Hz**		**60 Hz**				
Reporting Rates: (Frames per sec)	10	25	10	12	15	20	30

Phasor data is reported to PDCs, which concentrate phasors from multiple PMUs. A PDC is a computer software utility that utilizes dedicated server architecture to accept phasor data from multiple PMUs instantaneously. Figure 2.21 illustrates a PMU-PDC network.

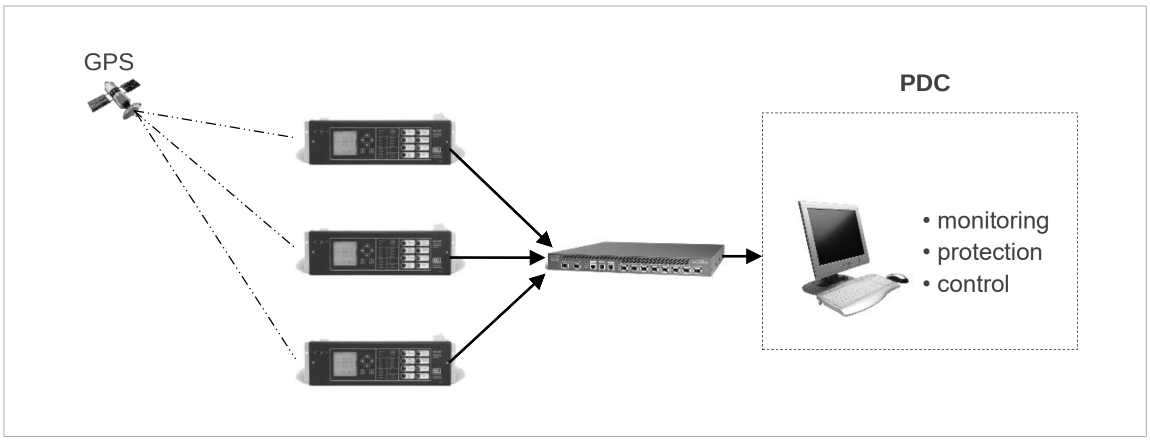


Figure 2.21: Typical PMU-PDC network

PMUs and PDCs are interconnected over the Ethernet communication media. Phasor data is collected from each PMU using the C37.118-2011 communication protocol. The data collected is then sent to a PDC database server for real time monitoring, protection, and control functions. The phasor data communication protocols and guidelines include;

a) IEC TR 61850-90-5:

   This protocol is a joint IEEE-IEC project for synchrophasor data communication that focuses on using the IEC 61850 models and processes while in-cooperating communication methods as per requirement. The protocol outlines transmission of digital state and time synchronized power measurements over wide-area networks, facilitating implementation of wide-area monitoring, protection and control (WAMPAC) systems.

b) C37.242-2013:

The C37.242-2013 outlines guidelines for installation, calibration and testing of PMUs.

c) C37.244-2013:

C37.244-2013 outlines guidelines for PDC communication requirements, stipulating the Total Vector Error (TVE) compliance levels as defined in the C37.118 standard.

d) C37.118 Standard for synchrophasors :

The C37.118 defines a method for synchrophasor data transmission between PMUs and PDCs. The standard stipulates the communication framework and application components to facilitate transmission and exchange of synchrophasor data. The C37.118 standard is split into two parts;

1. C37.118.1-2011:

   The first part of the standard stipulates synchrophasor frequency, and Rate of Change of Frequency (ROCOF) measurements under operating conditions. The standard outlines evaluation of the measurements specifying relative requirements for compliance for dynamic and steady-state conditions. Time tagging and alignment specifications are also outlined.

2. C37.118.2-2011:

   The second part of the standard defines real-time exchange of synchrophasor measurements. The standard presents a universal message framework suitable for most protocols with real-time communication between PMUs and PDCs.

Table 2.6 presents comparison between the C37.118-2011 standard for synchrophasors and the IEC 61850 communication standard.

Table 2.6: Comparative analysis between the C37.118 and the IEC 61850

Function	C37.118	IEC 61850 GOOSE & SV
Streaming Protocol	yes	sampled values
Report Rate	10-30 samples / sec	4800-15360 samples / sec
Native IP Routing	yes	no
Application	situational	control
Security	no	yes
Measurement Specification	yes	yes
Communication Specification	no	yes
Event Driven	no	GOOSE
Standard Communication Language	no	yes

2.4.4.2 GPS synchronization

Synchronization and alignment of instantaneous phasor data from multiple PMUs at different locations facilitate detection and analyses of phase error which correlates to the systems' stress. Figure 2.22 presents elaboration of phase error and analysis.

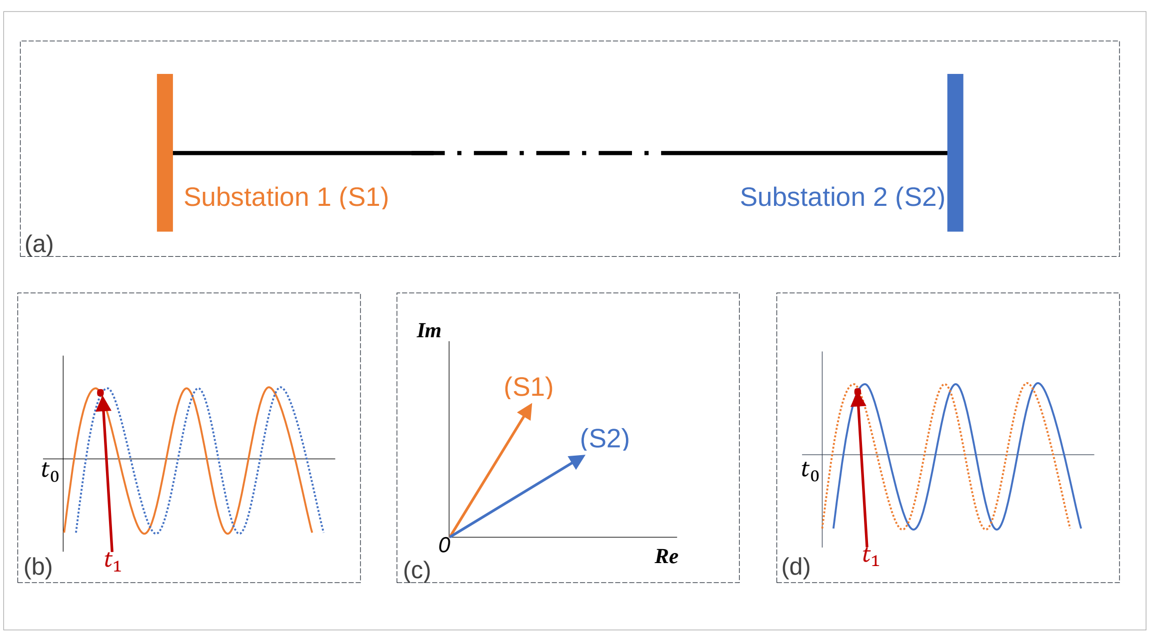


Figure 2.22: Phase angle analysis between substation A and substation B

In Figure 2.22, synchronized phasors or synchrophasor signals from S1 and S2 (which may be separated over a relatively long distance) facilitate evaluation of real-time system dynamics through correlation or graphical superimposing the signals at a specified instance as indicated by t_2 in the figure.

As outlined by the IEEE Std. C37.118.1-2011, PMUs utilize a GPS to establish a common timing reference. The global timing reference enables alignment of PMU phasors at different locations over widespread areas.

2.4.4.3 C37.118.2-2011 Message Framework

The transmitted phasor data message body consists of a message frames. The four message types for the message frames include (IEEE Power Engineering Society, 2005):

1. Data Messages:

   The Data Messages are measurements made by a PMU.

2. Command Frame:

   The Command Frame contains configuration and control code transmitted between the PMU and PDC

3. Header Frame:

   The Header Frame contains user defined information transmitted between the PMU and PDC.

4. Configuration Frame:

   The Header Frame contains machine-readable message describing the data types, calibration factors, and meta-data for both PMU and PDC communication

Table 2.7 provides description for the communication references defining representative formats.

Table 2.7: C37.118 Communication references

Communication References	Description
Command Frame (structured-binary format)	Start – Stop command to from host (PDC)
Header Frame (unstructured – ASCII text)	Up to 80 characters comment or any other information
Configuration Frame1 (structured – binary format)	Constant part of the PMU configuration
Configuration Frame 2 (structured – binary format)	Variable PMU phasor configuration
Data Frame (structured – binary format)	Real-time PMU phasor data – magnitude, phase angle, frequency, analog, digital data

The C37.118 is not limited to a unique communication system. Data transmission can be applied over any media, adhering to the specified message framework presented in Figure 2.23.

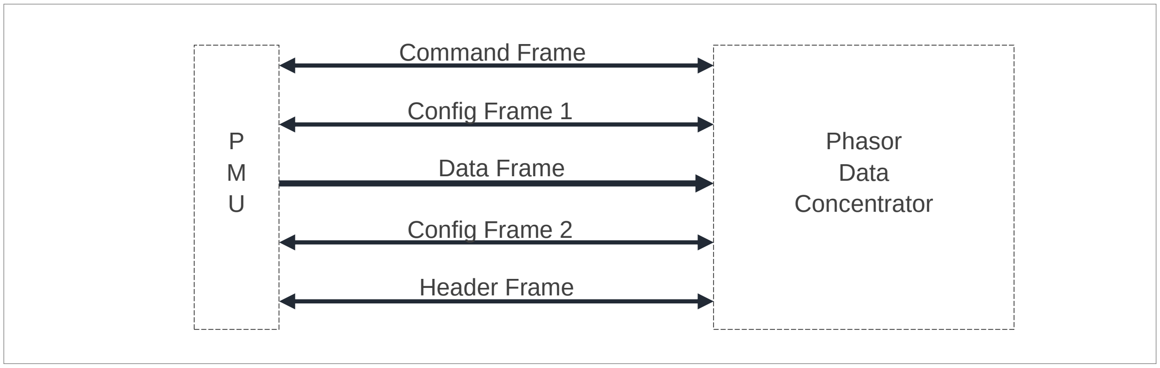


Figure 2.23: Message Framework data transmission

Considering MMS and the IP, the entire frame is written onto and read from the application layer of the stacked protocols. The entire frame is also transmitted over Ethernet and serial communication assuring data integrity in all cases. Figure 2.24 presents the message frame format.

Header information (14 bytes)	Status Field	Phasor (magnitude, phase angle)	Frequency	Rate of Change of Frequency	Analog data	Digital data	CRC

Figure 2.24: Message frame

2.4.4.3 Ethernet communication protocol dependencies

Synchrophasor use both the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). Table 2.7 presents a comparison between the TCP and UDP communication protocols.

Table 2.8: Comparison of Message frame transmission over TCP and UDP

TCP / IP	UDP
Secure protocol	Non – secure
Increased latency	Faster in comparison to TCP
Inefficient for real time operation	Efficient for real time application
No Multicast addressing	Multicast addressing

2.4.5 Impact of PMU technology on Smart Grid applications

Real-time measurement of synchrophasors (Singh et al., 2011) facilitates applications that can be grouped into three main categories;

1. Monitoring
2. Advanced protection
3. Advanced control

Some of the existing applications include post disturbance analysis, adaptive protection, system protection schemes and state estimation.

2.4.5.1 Impact of PMUs on power system operation

The impact of PMU integration into the power systems network is comparatively assessed against old SCADA technology. The large scale impacts include (Wu & Giri, 2006);

- Accurate estimates of the power system state obtained at frequent intervals.
- Timely execution of relative control functions, ensuring higher quality of power supply to the consumers.

- Improved and advanced post-disturbance analyses.
- Precise synchronous wide-area snapshots of the system states facilitated by GPS synchronization.
- Progressive security valuation of power system networks.
- Improved and advanced overall protection and response to disturbances.
- Advanced control utilizing improved controller performances.

2.4.5.2 Smart Grid applications

Real-time PMU applications with synchronized phasor measurements are more commonly utilized in power systems including dynamic power flow data visualization and problem identification of;

1. real-time frequency,
2. rate of change of frequency and
3. dynamic phase angle separation

(Adamiak et al., 2006) presents a real-time phasor plot Figure 2.25. Surface contours indicate instantaneous magnitude and the color schemes present phase angle relative to a reference angle.

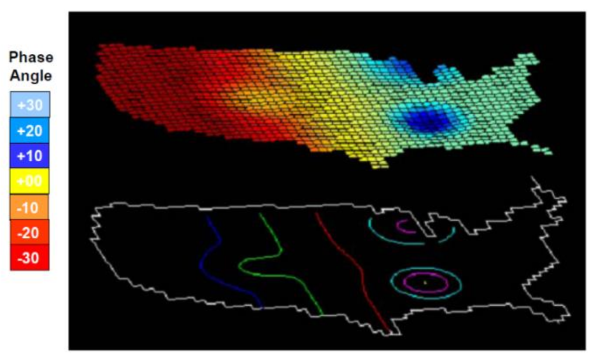


Figure 2.25: Voltage phasor, real time visualization (Querol et al., 2001)

A Wide-Area Monitoring System (WAMS), in Western Electricity Coordinating Council (WECC) is an example of an existing PMU network project (Hauer, 1997). The WAMS data is analysed for multiple purposes, including model validation (Pereira, 2003), modal analysis (Hauer, 1991), major blackout analysis (Kosterev et al., 1999), and special control actions (Kosterev et al., 1998). Figure 2.26 illustrates a WAMS with real-time application of synchrophasors (Nexans & Nerc, 2012).

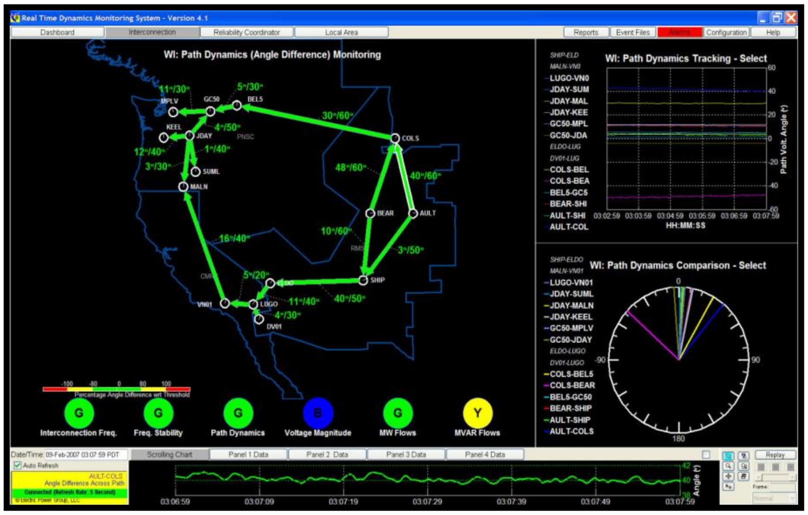


Figure 2.26: Real-time application of synchrophasors (Nexans & Nerc, 2012)

PMU projects that have been successfully completed (Wu et al., 2004) are presented in Table 2.9

Table 2.9: PMU Projects in North America (Wu et al., 2004)

Location	Project name	Status
North America	WAMS	successfully completed
North America	EIPP	successfully completed

In the projects multiple PMUs were integrated into the SASs, providing influx of synchrophasor data available for all network applications and system monitoring. Collective results verify the PMU phasor data being both instantaneous and accurate (Slutsker & Mokhtari, 1995) and optimizing the overall state estimation. Table 2.10 presents a tabulated literature review synopsis of research and development in PMU technology and PMU application.

2.4.6 Review of existing papers on phasor measurement technology and PMU applications

Table 2.10: PMU Research and development literature review

Paper	Aim Of Paper	Methodology	Hardware / Software Utilized	Advantages / Drawbacks	Achievements
(Wu & Giri, 2006)	Review PMU Impact on state estimation reliability for improved grid security	Comparatively analyzing performance of PMU-based systems and SCADA-based systems	Hardware : PMU	Beneficial impacts of PMU data on state estimation depend on PMU measurement accuracy and calibration, the number of PMUs and PMU locations.	Preliminary research verifies that PMU data provides instantaneous information that directly improves the reliability and robustness of state estimation
(Huang et al., 2007)	Evaluation of PMU Dynamic Performance in laboratory and field conditions	Capturing and analyzing PMU-based system dynamics in both laboratory and field conditions. Evaluating data from inconsistencies	Hardware : PMU	Providing a model for evaluating PMU dynamic performance	Outlining a general model for evaluating PMU dynamic performance via three complimentary methods • PMU model study, • PMU laboratory testing, • field evaluation
(Komarnicki et al., 2008)	Outlining the need to future proof PMU test standards to ensure optimal operation of PMUs from different vendors under different repeatable scenarios and test stand set-up conditions	Analysis of PMU test reviews carried out in Europe and in the United States of America	Hardware : PMU Signal generator Power Amplifier Software: Lab View	Aspects outlined in the paper can be further utilized to improve PMU operation in further research and development	Proposing a validated set up test stand and new various test procedures for PMU accuracy
(Khederzadeh, 2010b)	Proposing a new protection scheme for transmission lines equipped with Unified Power Flow Controller (UPFC) based on the synchronized phasor measurements at both ends of the line	Deriving and using location indices together with fault detection to determine internal and external fault events accurately	Hardware : PMU	The proposed protection scheme model remains unaffected by power system variations caused by the UPFC during faults	The protection scheme responds well and fast with regard to dependability and security, protecting the full line length

Paper	Aim Of Paper	Methodology	Hardware / Software Utilized	Advantages / Drawbacks	Achievements
(Kaminsky et al., 2011)	To present an open conception of PMU testing under both static and dynamic properties on a virtual instrumentation platform	Verification of PMU properties utilizing sets of software tools intended for designing the PMU.	Hardware : PMU Software: Lab view (National Instruments)	Financial constraints limiting further development of the test system	Realization of a flexible PMU Tester Based on Virtual Instrumentation
(Lira et al., 2011)	To present findings in the practical design and implementation of PMU based systems	Analyzing the required response of PMUs under various types of system disturbances while testing for frequency variations and presence of harmonics within the system	Hardware : PMU	Findings recommend the quality of PMU data from phasor data concentrator applications (PDC)	Identifying the realized integrated simulation environment used to test applications, to be an invaluable tool for understanding the PMU data together with power system behavior
(H. Liu et al., 2012)	To present PMU dynamic performance evaluation test system and discuss the results	Modelling and utilizing a PMU test system based on the Omicron which can generate high accuracy signals with time synchronization	Hardware : Omicron CMC IRIG-B PMU	Findings recommend that a lot more work needs to be done to improve the dynamic performance of the PMU	The tests brings into focus the importance of the dynamic mathematical model of the phasor and calculation window
(Zhang et al., 2012)	Analysis of PMU data on the robustness and precision of the Maximum Normal Measurement Rate (MNMR) State Estimator	Introducing synchrophasor current and voltage data, analyzing the performance response of the MNMR estimator		Introducing PMU data improves both robustness and precision of the MNMR estimator	The simulation results verify that the MNMR estimation precision is greatly improved by removing the zero-current injection constraints
(Lee et al., 2013)	Outlining PMU interface using the IEC 61850 standards	Meticulous review of the IEC- 61850-90-5 PMU interface in relation with the PMU communication standard C37.118.2		Development of applications for micro grid monitoring , load pattern analysis and modelling using PMU high bandwidth characteristics and synchronized measurements	Developing and verifying an IEC 61850 based PMU interface for performance analysis of a monitoring system

Paper	Aim Of Paper	Methodology	Hardware / Software Utilized	Advantages / Drawbacks	Achievements
(Hongga, 2007)	The paper presents a new model for measured and calculated state variables from a PMU state variable node	To utilize a PMU in the model to measure state variables while; • reducing estimation scale • improving calculation time • improving convergence speed	Hardware : PMU Software: Virtual State estimation simulation model	The research expands the state estimation model to the double state estimation model, improving state estimation equations' convergence speed and estimation precision utilizing PMU measurements	The simulation validates room to improve state estimation equations numerical stability and convergence speed
(Moraes et al., 2012)	The paper presents interoperability, interchangeability, steady state and dynamic performance analysis of PMUs	Carrying out comprehensive performance tests for 8 PMUs specific to the IEEE C37.118-2005 standard	Hardware : 8 - PMUs NIST Synchro Metrology Laboratory	Tests were established against C37.118.2005 criteria with a consistent methodology and test system	The results of the comprehensive tests and performance analysis of the PMUs provided important information for development of the equipment and verification of adherence to the standard specification requirements
(Bi et al., 2012)	The paper presents interoperability, interchange ability, and dynamic performance analysis of PMUs	Carrying out comprehensive performance tests for 3 PMUs specific to the IEEE C37.118.1 standard	Hardware : 3 - PMUs Omicron based test system	The dynamic performance of the PMUs does not meet standard criterion	Establishing that more work and revision need to be carried out to develop the dynamic frequency measurement of the dynamic phasor model

Paper	Aim Of Paper	Methodology	Hardware / Software Utilized	Advantages / Drawbacks	Achievements
(Khan et al., 2014)	The paper presents a parallel detrended fluctuation analysis approach for fast detection of transient events on large scale PMU measurements	Comparative performance analysis of ; • speed • accuracy • scalability between parallel detrended fluctuation analysis and standalone detrended fluctuation analysis	Hardware : WAMS architecture MapReduce computer cluster	The MapReduce Hadoop framework has over 180 configuration parameters enabling performance analysis tests over varied configuration of the parameters	Experimental results how the speedup of parallel detrended fluctuation analysis
(Cheng et al., 2014)	The paper presents complex number equations as a solution for PMU-only state estimation	Analysis of the error characteristics of PMU measurements in the complex domain		The complex number weighted least squares algorithm provides an effective method to estimate system states	Establishing complex measurement equations for PMU-only state estimation
(Kolosok et al., 2014)	The paper proposes test equation techniques for linear state estimation of power system facilities based on PMU phasor measurements	Utilizing algorithms based on the exclusion of unmeasured variables and state vector components from rectangular component equations		The approach provides a simple implementation, with high speed problem solving	The developed algorithms for local state estimation can be very easily integrated into the automated control system
(Castello et al., 2015)	The paper presents a multichannel operative mode together with awareness of synchronization conditions for PMUs	Modelling a distributed algorithm that is able to obtain an estimation of the systems time quality	Software: Mat Lab	The developed distributed time synchronization system shows restriction or a limit to the time synchronization performance	Establishing tradeoffs between bandwidth and accurate synchronization

Advent PMU technology has motivated research into augmenting synchrophasor application in protection systems, advancing engineering in power system protection. Until recently, PMU application research has been limited to wide area monitoring, creating a development imbalance with regard to the rate of PMU development versus industrial utility application.

There is no active research on extensive utilization PMU technology in line differential back-protection schemes. To date, no research has been dedicated to practical application of synchrophasors in differential line back-up protection. Figure 2.27 provides an overview of the transmission line and differential protection literature reviewed.

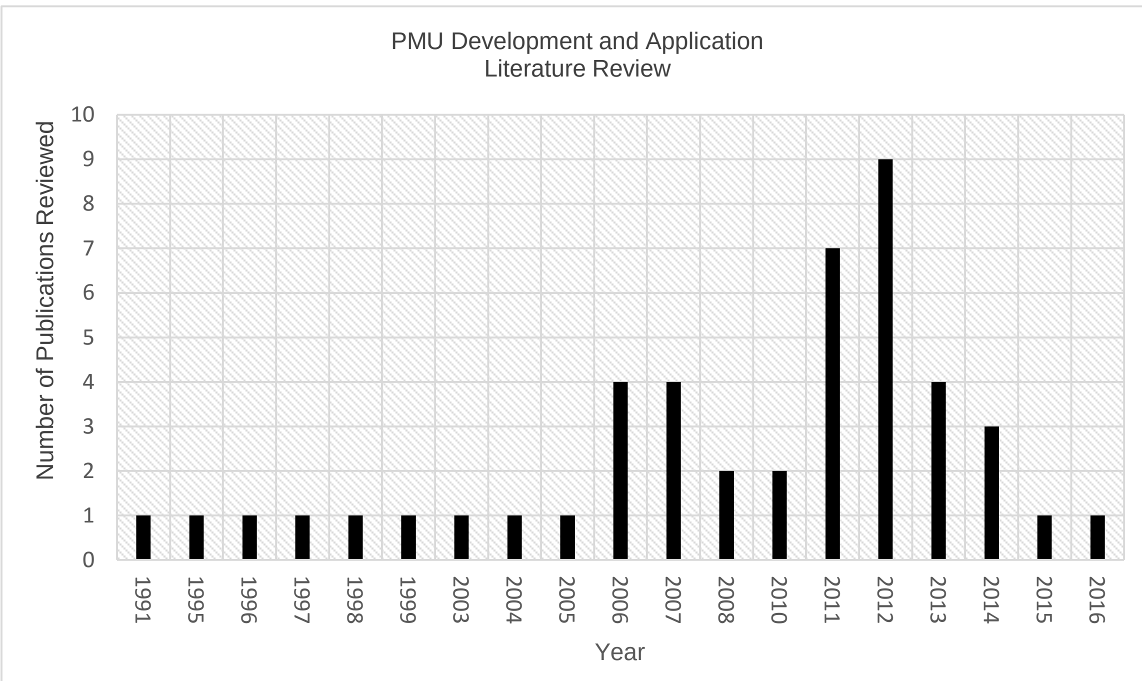


Figure 2.27: Transmission line systems and differential Relaying Literature Review

Between 1991 and 2016 forty six IEEE publications centering PMU development and PMU-based applications are reviewed. A general distribution model of the literature review is presented, with PMU research and development peaking in 2011 and 2012.

Research and development established by (Dalcastagnê et al., 2008), (Khederzadeh, 2010a), (Jiang et al., 2000) and (Y.-H. Lin et al., 2004) is discussed here, critically analyzing achievements to be considered and adopted in the development of the PMU-based line differential protection scheme. Excessively narrow focus from (Dalcastagnê et al., 2008) and detailed assessment from (Khederzadeh, 2010a) resulting in hypothesis validation are described here.

- Excessively narrow subject focus:
  (Dalcastagnê et al., 2008) neglects comparatively analyzing the impact of the synchrophasors against unsynchronized phasors in the two-terminal fault-location method. The subject tightly concentrates on application of the voltage magnitudes neglecting the current magnitudes and phase angles which suggests incomplete, unnecessarily narrow and possibly biased results, which may lead the proposed approach to converge to incorrect conclusions.

- Detailed assessment and hypothesis validation leading to reliable conclusions:
  (Khederzadeh, 2010a) provides sufficient detail to validate the hypothesis and reliability of the conclusions. The approach recognizes synchronous current and voltage phasor measurements (synchrophasors) from the line ends to mitigate and overcome inherent drawbacks in conventional protection.

2.4.6.1 Final thoughts

(Dalcastagnê et al., 2008), (Khederzadeh, 2010a), (Jiang et al., 2000) and (Y.-H. Lin et al., 2004) fully utilize phasor technology, effectively detailing novel protection concepts based on unsynchronized phasors and synchrophasors. A critical point to emphasize in the synopsis of the reviewed papers is the lack of use of more current guidelines and prevailing standards which specifically outline; coherent time-alignment, communication and concentration of phasors. Phasor-based transmission line protection concepts are adopted from (Dalcastagnê et al., 2008) expanding narrow concentration on voltage magnitude phasors to include current magnitude phasors and phasor angle as in (Khederzadeh, 2010a), the phasor angle being utilized as an excellent indicator for system stress. (Jiang et al., 2000) and (Y.-H. Lin et al., 2004) identify pragmatic hypothesis validation methodologies which are also adopted and used in further research and development of the PMU-based transmission line current differential protection scheme.

2.5 Conclusion

Chapter Two identifies relevant and essential concepts necessary to validate the research hypothesis to provide a relevant contribution towards development of PMU applications, realizing a PMU-based backup protection scheme. The Chapter establishes a literature review framework and presents critical discussion on essential subject matters relating to development, analysis and validation of the hypothesis.

Three subject matters which encompass the IEC 61850 communication standard, transmission line system protection and PMU application development are reviewed. Each of the subject matters are exclusively reviewed and discussed, highlighting adoptable methods, unstated assumptions and a knowledge gap in utilizing phasor measuring technology for transmission line protection.

Chapter Three introduces the concept of the PMU-based line current differential protection scheme. The Chapter details and analyses the features of the protection scheme, emphasizing application of design principles and methods established in the literature review frame-work. Chapter Three also highlights groundwork standards and guidelines considered in modelling the PMU-based line differential protection scheme.

Chapter 3 :

Feature and analysis of the PMU-based line differential protection scheme

3.1 Introduction

Chapter Three presents feature and analysis of the potential Phasor Measurement Unit (PMU)-based transmission line differential backup protection scheme. This Chapter applies methods and concepts adopted from papers reviewed in Chapter Two. Considerations for modelling conventional transmission line differential protection schemes are reviewed, highlighting benefits and drawbacks of existing protection methods.

This Chapter describes considerations in modelling conventional transmission line differential protection schemes, emphasizing challenges associated with typical differential relays which may be mitigated by the potential PMU-based transmission line differential protection scheme. Considering outlined pitfalls from conventional differential protection schemes and established PMU-based protection methods adopted from (Dalcastagnê et al., 2008), (Khederzadeh, 2010a), (Jiang et al., 2000) and (Y.-H. Lin et al., 2004) the proposed PMU-based transmission line differential protection scheme is presented, outlining all assumptions and theoretic expectations.

A detailed study on real-time phasor data streaming and phasor data transmission is also presented in this Chapter. It is necessary to highlight that PMU-data transmission is critical as it forms the basis of the hypothesis which is realized from further development of the IEEE C37.238-2011 Precision Time Protocol (PTP) which qualifies considering synchrophasors for protection.

Once the feature of the PMU-based protection scheme is presented, and real-time phasor data transmission analyzed a general comparison between conventional transmission line differential protection schemes and the novel PMU-based line differential protection scheme is presented.

3.1.1 Chapter organization

Section 3.2 outlines considerations for transmission line current differential protection, highlighting challenges associated with differential relays in conventional transmission line protection.

Section 3.3 describes implementation of the current differential algorithm in conventional line current differential protection schemes.

Section 3.4 presents the PMU-based line current differential protection scheme. The concept and features of the potential back-up protection model are described outlining all assumptions and theoretic expectations.

Section 3.5 describes phasor measurement data transmission and phasor data concentration.

Section 3.6 presents a comparative analysis between conventional line current differential protection and detailed PMU-based protection, emphasizing a knowledge gap with regard to utilizing phasor measurement technology in line current differential protection applications.

3.2 Transmission line current differential protection

Chapter Two introduces and discusses principles behind line current differential protection, where current differential protection functions respond to currents within a protected zone with the sum of all currents ideally nearing zero under no internal fault conditions. Pragmatic protection scheme design considerations account for measurement errors and fictitious differential signals through implementation of adequate counter-measures. The conventional countermeasures against fictitious current differential signals include (Zhang et al., 2009), (Miao et al., 2010)and (Lopez et al., 2015);

- the harmonic restraint method
- the percentage restraint method
- introducing intentional time delays
- using adaptive restraining techniques

Relays in transmission line current differential protection schemes operate in one of two modes;

Master Mode:

In master mode, the master relay receives current information from peer relays on the line, administering relative differential trip equations that facilitate Direct Transfer Tripping (DTT) which consequently enables slave relays to issue trip commands to their breakers as illustrated in Figure 3.1 (Miller et al., 2010b).

Remote Mode:

In Remote mode, the remote relay transmits current data, but does not receive any current data and does not facilitate any current differential protection directives.

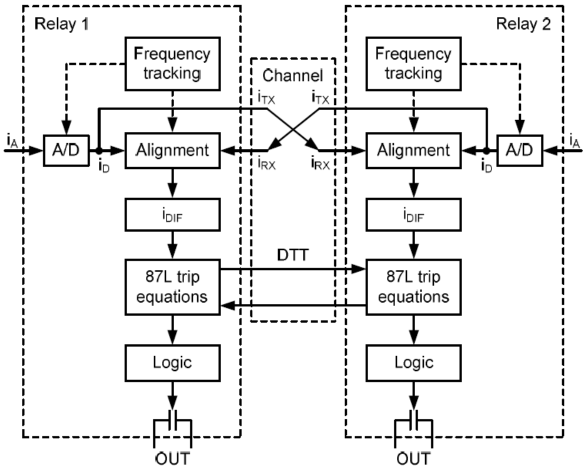


Figure 3.1: Direct Transfer Tripping (DTT) implementation in transmission line differential relaying (Miller et al., 2010b)

In a two terminal transmission line, a bi-directional communications path is essential. Figure 3.2 illustrates a conventional method used to improve overall dependability of the current differential line protection scheme, making use of redundant communications channels. In the scheme, data is continuously transmitted over both channels, but the relay uses only the data from a single channel.

If channel failure occurs on the primary channel in use (channel 1), the relays will automatically switch to the other channel (channel 2), and will continue to provide current differential protection (Adamiak et al., 1998).

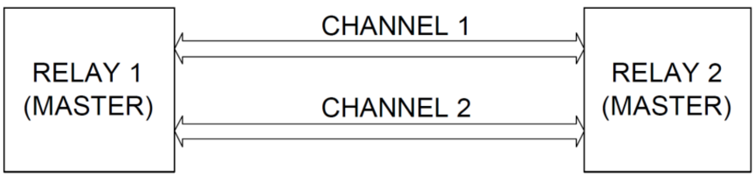


Figure 3.2: Redundant communication channels (Adamiak et al., 1998)

Phasors required twice as much more bandwidth to transmit both the real and imaginary parts of the currents which limits the phasor data exchange rate which prohibit phasor interpolation.

3.2.1 Design considerations for a current differential protection scheme

Design considerations for line current differential protection schemes are detailed here, outlining prevailing criteria for modelling modern transmission line differential protection schemes that are accurate, dependable and efficient.

a. Adoption of Contemporary protection utilities

One of the more critical considerations is the migration from traditional protection utilities to modern or new high-performance technology protection utilities that provide advantages and benefits that support advanced engineering of protection concepts that relate to;

- sensitivity
- selectivity
- dependability
- speed of operation

b. Security considerations

Security of conventional transmission line protection schemes is determined by three main factors (Kasztenny et al., 2011);

- robustness of the relay hardware and firmware
- sensitivity and response to channel impairments and measurement errors
- robustness of administered current differential algorithms and trip logic control

An advanced consideration regarding modern relays in design of protection schemes provides event recording to facilitate post-event or post disturbance analysis.

c. Auxiliary functions

Prevailing IEEE standards and guidelines necessitate relays capable of supporting emerging functions. The IEC 61850 standard for substation communication and the C37.118 standard for synchrophasors are perfect examples that emphasize the necessity for future proof relay models. In general relays must support;

- Digital Fault Recorder (DFR) requirements
- Sequential Events Recorder (SER) requirements
- Digital Disturbance Recorder (DDR) disturbance monitoring
- Critical Infrastructure Protection (CIP) requirements for password security

d. Reclosing

A modern line current differential relay should be capable of reclosing.

3.2.2 Drawbacks associated with current differential relays in transmissions line protection

Line current differential relays require design considerations that account for all failure modes including those caused by the active communications infrastructure between the relays. The drawbacks associated with the current differential protection relays are presented here.

Communications channel:

Line differential protection utilizes a communication channel between both ends of a two-ended transmission line. The relays are separated over long distances that can vary up to hundreds of kilometers. Regardless of the distance, relays need to share and exchange instantaneous current measurements utilizing a provided communication channel. However, because of the long distance across which the data exchange occurs, the communication channel is susceptible to impairments and limitations. Some of these impairments include bit errors, asymmetry, path switching, loopbacks, frame slips and unintentional cross-connections between separate differential protection schemes (Kasztenny et al., 2011). The channel bandwidth restriction is a particularly important consideration as it may directly hinder coherent communication and synchronous exchange of larger current values.

Alignment of digital current values:

Line differential protection depends on synchronous sampling between both local and remote relays. The differential principle employed by the differential protection schemes is based on the premise that current data from both line ends is sampled simultaneously.

The more conventional method for current data alignment employs the Ping-Pong algorithm. During data transmission, the Ping-Pong algorithm is susceptible to asymmetry. When the channel is asymmetric, the Ping-Pong algorithm fails which in turn generates phase error proportional to the amount of asymmetry.

Another drawback is the channel latency which may change in response to the communications path switching when using multiplexed channels (Voloh et al., 2005).

Sensitivity:

Transmission line faults may result from variable conditions which more often include tower grounding resistance. Consequently, power line faults are frequently accompanied by relatively high fault resistance evident in single line to ground faults (Pires & Guerreiro, 2008). The faults, if undetected timely cause cascading failures. A great deal of emphasis is therefore needed when considering line sensitivity and faster communication speeds.

Line charging current:

The length of a transmission line is proportional to its charging current (Kasztenny et al., 2011). Longer transmission lines in this regard are more susceptible to substantial amounts of charging current. The digital differential relays consider line charging current as part of the line feed. If the line charging current is unaccounted for, it is factored into the differential calculation. This misleading current input signal compromises sensitivity and security, causing system mal-operation.

Current transformer saturation from external faults:

With regard to accuracy and high sensitivity, current transformer errors need to be accounted for by applying fail safe countermeasures against current transformer saturation which is caused by external faults (Morais Pereira & Zanetta Jr., 2005), (Richards & Tan, 1983). Countermeasures are necessary and include increasing in the restraining action and introducing intentional time delays (Miller et al., 2010b).

Series compensated lines:

(Altuve et al., 2009) outline voltage and current inversion in series compensated lines resulting from compensated lines' capacitive reactance. Capacitor overvoltage protection creates non-linearity in the series capacitor circuit. This presents unequal bypass action between phases which alternately creates series unbalance (Kasztenny et al., 2011). This series unbalance couples the sequence networks that represent the protected line, challenging traditional protection assumptions and relationships between sequence currents and voltages during both internal and external faults. Although the differential principle remains independent from series compensation, current inversion and coupling between sequence networks create dependability challenges and may also delay the differential protection action for internal faults (Xue et al., 2013).

3.3 The line current differential algorithm

3.3.1 The differential signal

Figure 3.3 shows the current differential relaying principle based on Kirchhoff's Current Law. Protection function operates based on the sum of currents flowing in the Current Transformer (CT) secondaries.

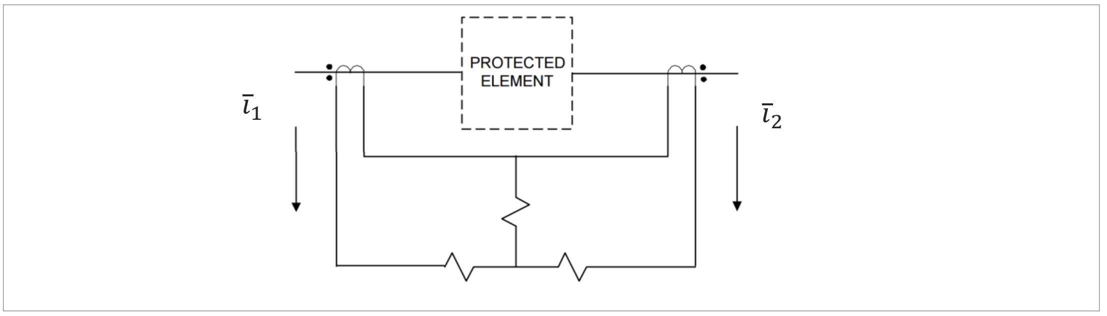


Figure 3.3: Differential relaying based on Kirchhoff's current law (Adamiak et al., 1998)

Equation 3.1 defines the differential signal i_{DIFF} for a line bounded by N currents, $\bar{\iota}_1$ through $\bar{\iota}_N$. The differential signal is typically filtered for better accuracy and used to qualify an internal fault.

$$i_{\mathrm{DIFF}} = \bar{\iota}_1 + \bar{\iota}_2 + \ldots + \bar{\iota}_N \tag{3.1}$$

Where $\bar{\iota}_1$ and $\bar{\iota}_2$ are current magnitudes at any given instance. Equation 3.2 illustrates representation of a filtered differential signal.

$$I_{\mathrm{DIFF}} = |i_{\mathrm{DIFF}}| \tag{3.2}$$

Where I_{DIFF} is the differential current utilized to evaluate fault conditions.

3.3.2 The restraint signal

Summing the currents flowing in the CT secondaries yields inaccurate differentials, where the differential signal differs from absolute zero values. Practical evaluation of the differential signal requires a percentage differential characteristic to pre-emptively mitigate the non-zero differentials that may occur (Altuve et al., 2004). The restraining signal is utilized to verify the differential signal, differentiating the internal faults from the external faults while reflecting the overall current level of the transmission line. The restraining signal is an arbitrary signal that provides security limited to CT saturation.

$$i_{\mathrm{RES}} = |\bar{\iota}_1| + |\bar{\iota}_2| + \ldots + |\bar{\iota}_{\mathrm{N}}| \tag{3.3}$$

The restraining signal $\mathrm{I}_{\mathrm{RES}}$ in a two ended transmission line is denoted by Equation 3.4 (Kasztenny et al., 2011).

$$I_{\mathrm{RES}} = K \bullet |I_{\mathrm{DIFF}}| \tag{3.4}$$

The function of the restraining signal is to reflect the overall current level for all the line currents, providing security limited to a specified level of CT saturation.

3.3.3 The operating characteristic

The differential trip logic is triggered when the differential signal exceeds a predetermined Pickup Value (PV) and a specified percentage of the restraining value. Equation 3.5 and 3.6 represent these conditions (Kasztenny et al., 2011). Equation 3.7 presents the combinational condition from Equation 3.5 and 3.6.

$$I_{\text{DIFF}} > PV \tag{3.5}$$

$$I_{\text{DIFF}} > K \bullet I_{\text{RES}} \tag{3.6}$$

$$I_{\text{DIFF}} > PV + K \bullet I_{\text{RES}} \tag{3.7}$$

Figures 3.4 (a) and (b) illustrate the differential-restraining plane where the characteristic is represented by a straight line with a slope K. The characteristic verifies CT fault current and alignment fault current proportional to the restraining signal.

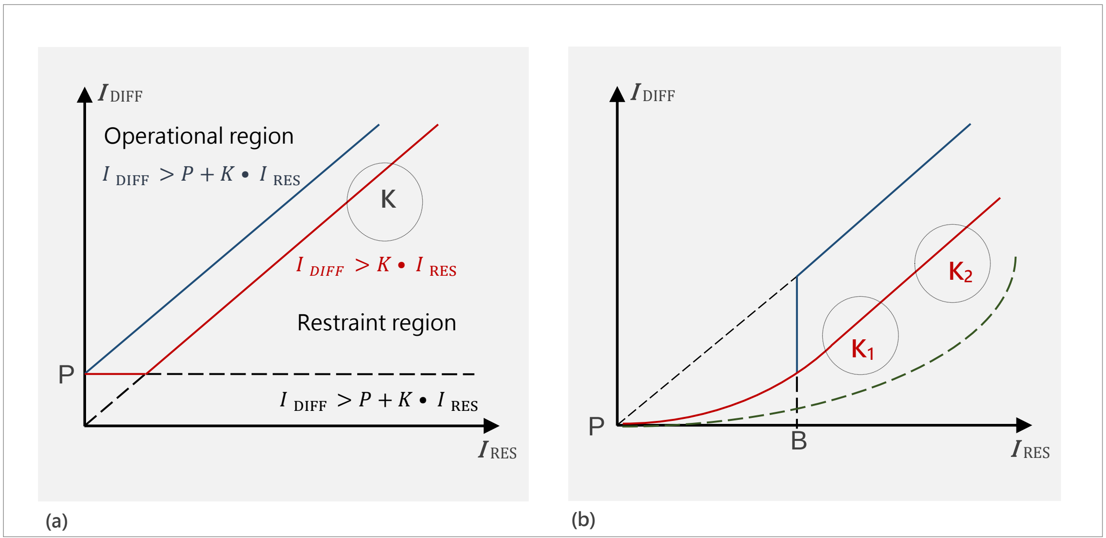


Figure 3.4: where K is percentage of a restraining signal
(a) single - slope percentage differential characteristics;
(b) fictitious differential signal and dual – slope percentage differential characteristics
(Kasztenny et al., 2011).

The occurrence of CT saturation relational to greater fault current causes an increase in the fictitious current leading to the application of the dual slope percentage differential application presented in Figure 3.4 (b).

The dual slope characteristic increases the security for high current external faults applying a greater restraint to compensate for the large CT fault current while moderating sensitivity to low current internal faults.

3.4 The PMU-based transmission line current differential protection scheme

Section 3.2.1 presents design considerations for modern conventional transmission line differential protection which give emphasis to adoption of contemporary high performance protection utilities to establish advanced protection engineering. Section 3.2.2 alternately emphasizes drawbacks that exist with prevailing transmission line differential protection schemes based on typical digital differential protection relays.

In this regard, a PMU-based protection solution relates to adopting a higher performance protection utility that establishes advanced protection engineering based on synchrophasors to mitigate drawbacks that exist with prevailing transmission line differential protection schemes based on typical digital differential protection relays including alignment of digital current values. This section introduces the concept and features of the PMU-based transmission line current differential protection scheme.

3.4.1 Theoretic outline

The PMU-based line differential protection scheme is presented here. The protection scheme is modeled taking into account the design considerations for modern line current differential protection schemes outlined in the second section of this Chapter. Utility criteria for the PMU-based line differential protection scheme incorporates;

- C37.118 synchrophasor standard adherence
- IEC 61850 communication standard compliance
- Providing a modern solution for back-up protection

3.4.2 Design concept and principle of the operation

The groundwork scope for the proposed PMU-based protection scheme is limited to a two-ended or two terminal transmission line. Figure 3.5 illustrates the application principle of the PMU-based protection scheme. A general description for each function is provided.

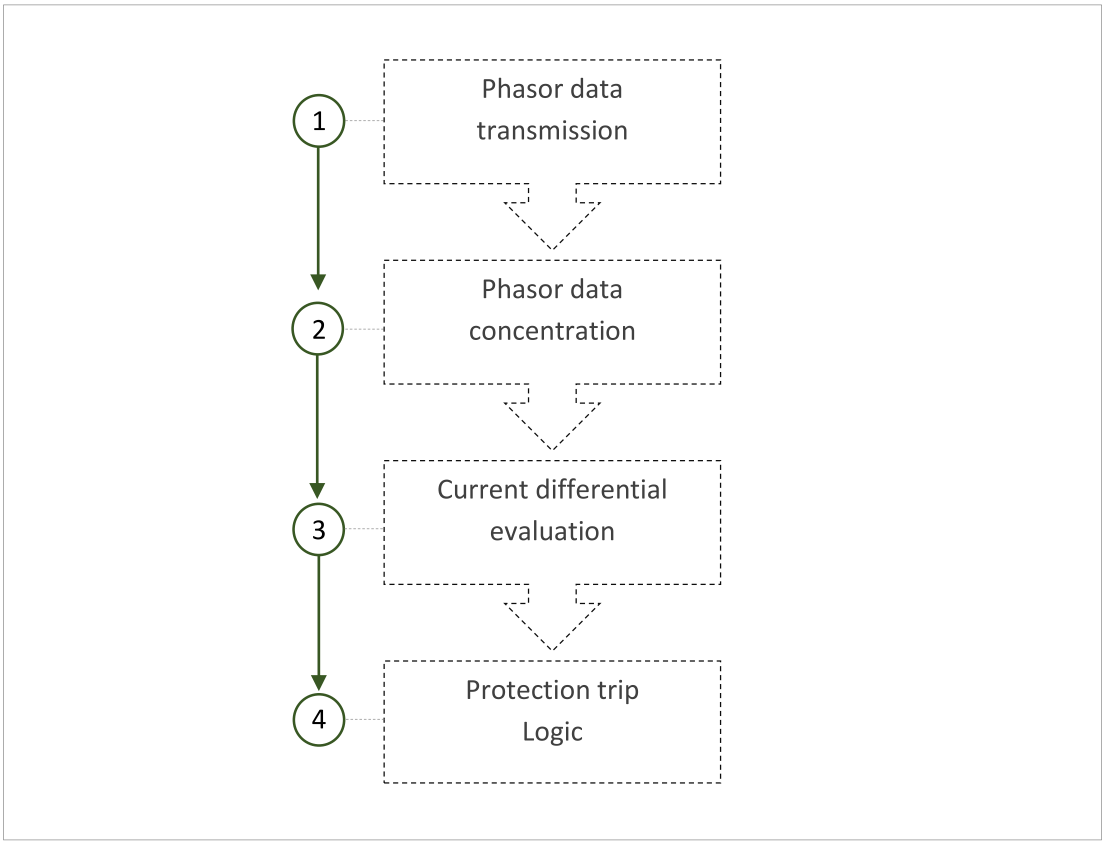


Figure 3.5: PMU-based protection application

1. Line-end or line terminal current phasor data related to the protection zone is transmitted to a station level PDC which is part of the energy management system that monitors the systems and administers system control and protection.

2. The transmitted phasor data is coherently concentrated and archived by application specific PDC software.

3. The concentrated phasor data facilitates real-time evaluation of the current differential and implementation of the current differential algorithm.

4. In the event of a fault or disturbance the control trip logic administers protective functions.

The phasor measurement technology in the PMU-based protection scheme recourses conventional design considerations for differential protection schemes. Figure 3.6 presents a proposed PMU based line differential protection scheme.

The coherently aligned phasor data or synchrophasors from both line-ends are utilized to evaluate current differentials and facilitate protection trip logic remotely.

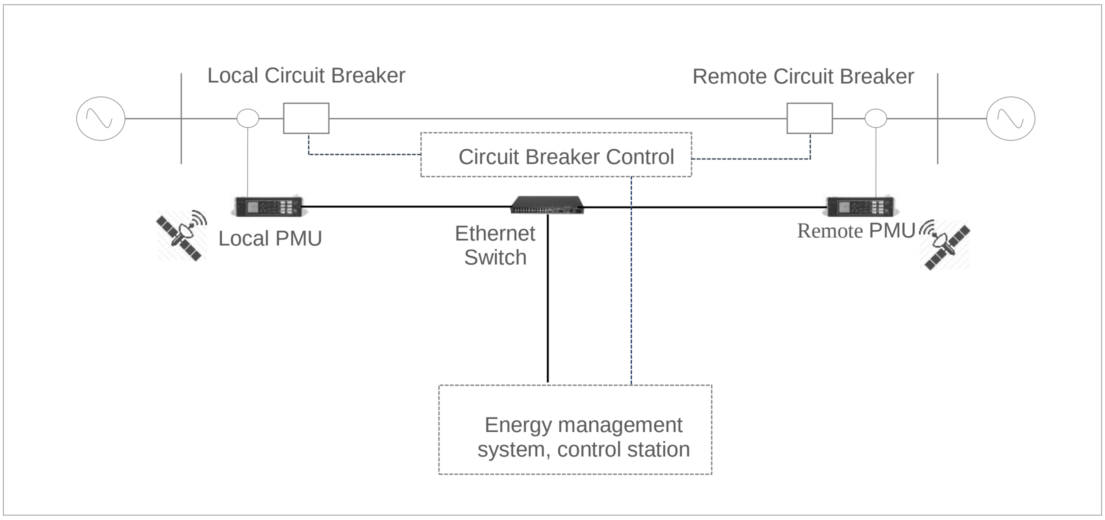


Figure 3.6: PMU-based line differential protection scheme

Table 3.1 contrasts the proposed protection model to a conventional 87L protection model. The proposed protection scheme requires an alternate platform to administer the differential algorithm, as opposed to the digital current differential relays which have the algorithm embedded within them (Wheatley, 1985), (Albrecht et al., 1992), (Xu et al., 2007).

Table 3.1: Comparison between a conventional 87L protection scheme and the PMU-based protection scheme

Function / Description	87L based Protection scheme	PMU-based Protection scheme
Communication	Ping-Pong	PC37.244
Algorithm Implementation	local - embedded in relay	remote - control station (EMS)
Wide Area Monitoring Capable	NO	YES

3.5 PMU data transmission and phasor data concentration

Phasor measurements from the PMUs are time tagged with Coordinated Universal Time (UTC) (IEEE Power and Energy Society, 2011). Utilizing high precision One Pulse per second (1PP) GPS receivers, the PMUs synchronize their sampling clocks accurate to microsecond precision. The coherently aligned phasors or synchrophasors are then sent to a PDC over a specified communication media.

3.5.1 Ethernet data transmission protocols

The line end PMUs are connected over Ethernet media, a link layer in the TCP/IP stack with speed up to a thousand megabits per second (1000 Mbit/s) (Cisco, 2016) . Phasor data transmission over Ethernet is protocol specific, using the User Datagram Protocol (UDP) or the Transmission Control Protocol (TCP).

Considering phasor data transmission, UDP holds a significant advantage to TCP as it enables relatively secure and faster transmission of phasor data to a destination IP address without establishing a communication link or predetermining a transmission channel. Despite no delivery acknowledgement and loss of packets, discrete packages of streaming phasor data remain unaffected. Figure 3.7 illustrates the communication differences between the two protocols.

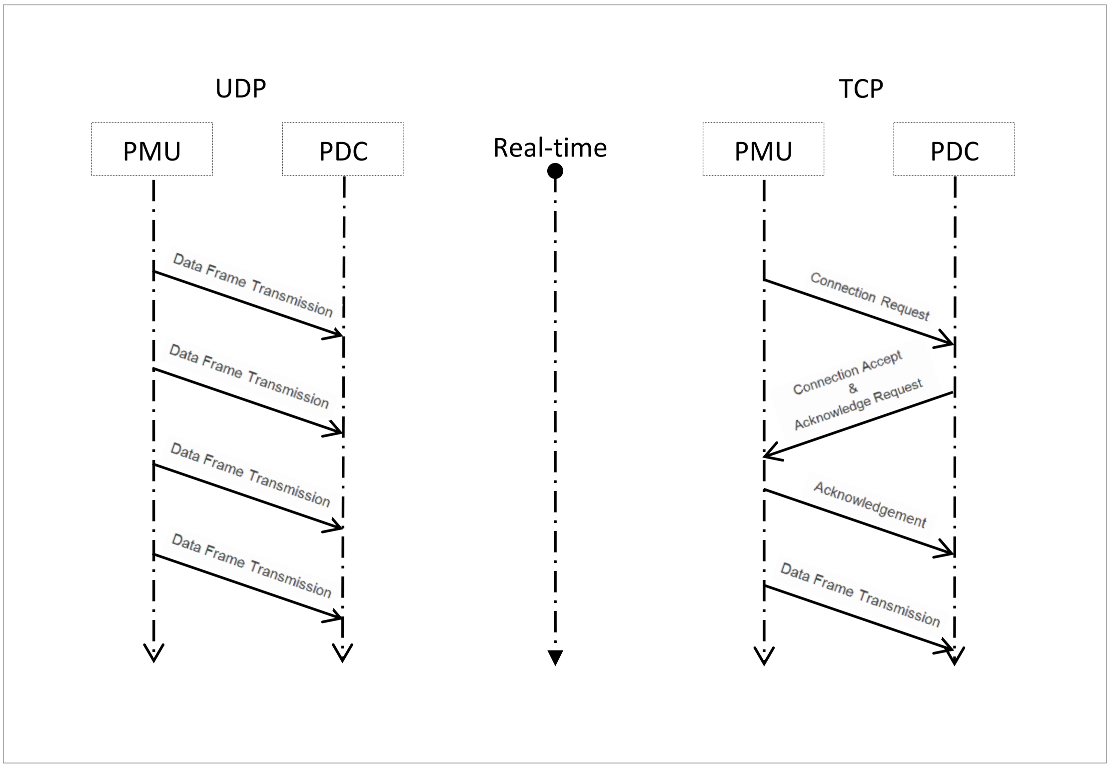


Figure 3.7: Comparison between TCP and UDP Data Stream Protocols

Characteristic comparison between the TCP and UDP protocols is presented in Table 3.2.

Table 3.2: Comparison between TCP and UDP Data Stream Protocols

Characteristic	UDP	TCP
Protocol Connection	Data is sent without establishing a connection link	A connection link is established before any data transmission
Data Interface	Message based discrete package transmission	Unstructured stream based transmission
Reliability and Acknowledgements	Unreliable best-effort transmission without acknowledgements	Reliable delivery of acknowledged messages
Data flow management	none	Sliding windows and congestion avoidance algorithms
Retransmission	none	Data delivery management, automatic retransmission
Transmission Speed	high	low

Considering the comparative analysis between the TCP and UDP protocols, the UDP protocol is selected as the administrative communication protocol in the PMU-based protection scheme, emphasizing its transmission speed advantage over TCP. It is worth mentioning that real-time coherent alignment of phasors negates the UDP transmission protocol pitfalls given in Table 3.2, which include no data flow management, no data retransmission and no successful data transmission acknowledgements.

Protection and control consider a further stage in substation evolution with shift of the system control and protection assets from hardware standards to interchangeable and interoperable future proof software architecture-based standards evident in advent PDC technology and the PC37.244-2013 PTP standard. The engineered software utility detailed in Chapter Four is a critical contribution in the research, providing an interoperability solution for implementing synchrophasors directly in transmission line differential protection, adhering to the IEC 61850 communication standard for electrical substation automation systems, performance requirements outlined by the C37.118-2011 standard for synchrophasors and protocol guidelines specified by the PC37.244-2013 guide for PDC requirements for power system protection, control, and monitoring.

Although the UDP protocol is selected, the TCP protocol is also considered to establish end point connection and socket data transmission between the line terminal PMUs, and the phasor data processing utility software that administer the transmission line systems' control and protection functions.

3.5.2 Phasor data concentration

Data transmitted by the PMUs will be synchronously concentrated by a PDC. The PDC is also time synchronized to limit errors that may result from the physical distance between the PMUs and the PDC itself.

Time alignment between the PDC and the multiple PMUs facilitates accurate communication and reliable timing information across the phasor data network. As part of its function, PDCs verify: time alignment, translation, errors, and data rates from the PMUs. When the measurement frames are processed they are then used for monitoring protection and control applications (Martin, 2006).

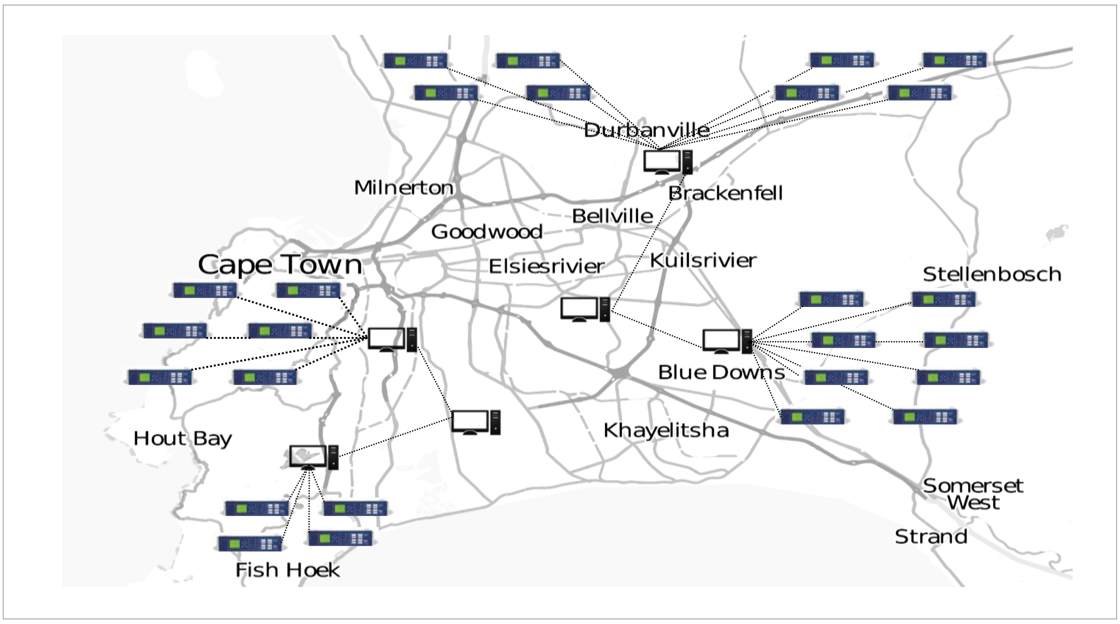


Figure 3.8: PDC-PMU Network

Figure 3.8 illustrates multiple wide area PMU-PDC networks that connect over long distances. It is important to note that distributed PDCs may also interact with each other on a peer-to-peer basis and the PMU-PDC network may connect to additional PDCs that may provide additional utility (IEEE Power and Energy Society, 2012).

3.5.3 IEEE PC37.244-2013 Standard guidelines

The PC37.244-2013 standard specifies requirement guidelines for PDC performance. All PDC applications administer specified guidelines to establish acceptable, adequate performance metrics that ensure accurate and optimal operation of PDC applications.

Essential PC37.244-2013 guidelines that outline crucial performance requirements for PDC applications are described here (IEEE Power and Energy Society, 2012).

a. Conformance:

   The PC37.244-2013 conforms to the C37.118.1 synchrophasor measuring standard and the C37.118.2 standard for phasor data transfer. The C37.118.1 defines measurements for all operating conditions including frequency and ROCOF(IEEE Power and Energy Society, 2011).

   The standard also outlines specific measurement evaluation methods to strictly ensure conformance to performance requirements. The C37.118.2 specifies a standard for transmission of synchronous phasor data between PMUs and PDCs in real-time, outlining formats, data types, message types and content (IEEE Power and Energy Society, 2011).

b. Performance:

   PC37.244-2013 PDC application performance requirements are not limited to the C37.118.1 and C37.118.2 standards but they are extended to other IEEE standards to model comprehensive performance requirements specific to PDC applications.

   The IEC 61850-90-5 is a perfect example of one of the IEC standards extended by the PC37.244-2013. Performance requirements also encompass application specific components which include the transmission networks frame-work and the number of PMUs transmitting to the PDC (IEEE Power and Energy Society, 2012).

c. Sampling:

   PDC report rates are configured to equal PMU report rates. In practice imprecise operating configuration may cause PDC to receive data at PMU report rates not equal to its own necessitating re-sampling. It is critical for the data to be re-sampled, to cover all up-sampling and down-sampling scenarios.

   PDC reporting rates are evaluated using the Fraction of Second (FRACSEC) values. The STAT word bit 9 is a flag bit that reflects any modification to measurements characteristics of the PMU. This bit indicates a re-sampling necessity and is set to 1 if the PDC is required to resample the data.

d. Data aggregation, synchronization and validation:

Before application specific processing, a PDC is required to synchronize time aligned phasors from multiple PMUs. Packet losses in real-time transmission networks cause frame alteration. It is therefore necessary to verify the accuracy of the phasor data being received by PDCs.

Verification of phasor data is achieved by evaluating the incoming and outgoing network traffic at the PDC node, identifying the message frame time stamp which consists of the three parts;

Part 1: Second-of-Century (SOC):

The SOC is a binary count of seconds from UTC midnight (00:00:00) of January 1, 1970 to the current second (IEEE Power and Energy Society, 2012).

Part 2: Fraction-of-Second (FRACSEC):

Each device has a TIME_BASE integer and the actual fraction of the second count is an integer representing the FRACSEC divided by the TIME_BASE of the device so that each frame is evenly spaced (IEEE Power and Energy Society, 2012).

Part 3: Time quality flag:

The SOC and FRACSEC components combine to form the actual time as shown in Equation 3.8 (IEEE Power and Energy Society, 2012).

$$\mathrm{Time} = \mathrm{SOC} + \mathrm{FRACSEC} \,/\, \mathrm{TIMEBASE} \tag{3.8}$$

Figure 3.9 illustrates how message frame measurement information between the PMU and PDC are compared, matching inputs based on similar time stamps, reinforcing correct transmission of packets (IEEE Power and Energy Society, 2012).

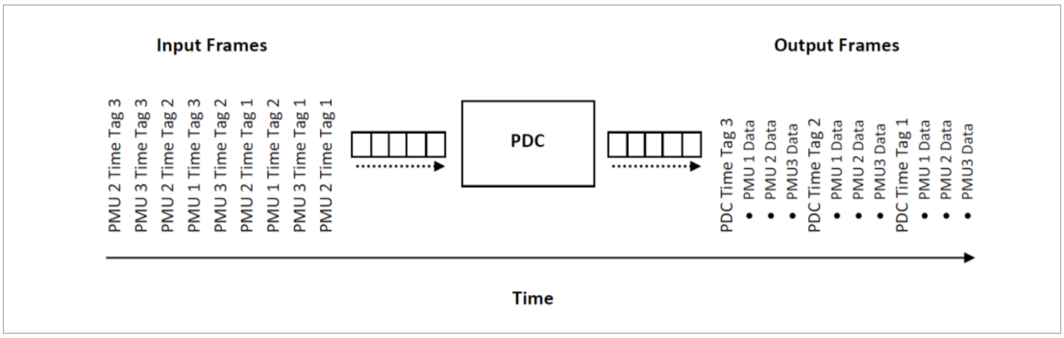


Figure 3.9: PDC input and output frames (IEEE Power and Energy Society, 2012)

For data validation, the data frames and status flags are checked. The PDC output measurement blocks display data quality and utilize these flags to establish data validation. The validation flags represent and validate;

- PMU error
- data arrangement
- time locked quality
- time synchronization

e. Data format and coordinate conversion:

Phasor measurement data frames can be represented by fixed integers and floating points or polar and rectangular coordinate form formats. Multiple PMUs can transmit unique format types to a single application specific PDC which maintains its output frame in a specified format. Once synchronized and validated the phasors data is converted into a single format.

f. Capacity and Buffering:

Capacity refers to the maximum data a PDC can process without exceeding or increasing specified latency. PDC latency relates to phasor data processing time for a stated number of PMU inputs. The buffer is the storage capacity of the PDC. Equation 3.9 illustrates the evaluation of the buffer limit. When the PDC processes the largest delayed input, the buffer size is calculated as follows (IEEE Power and Energy Society, 2012);

$$\text{Buffer Size} = T_{delayed\ PMU} - T_{first\ PMU} \tag{3.9}$$

Where;

$T_{first\ PMU}$: Arrival of the first PMU frame

$T_{delayed\ PMU}$: Largest acceptable delayed input

g. Latency:

PDC latency is critical and has more stringent performance requirements. The variables that affect phasor data latency include;

- data frame construction
- network data transmission
- packet processing, which include;
  - PDC wait time and
  - PDC process time

The data frame construction and network transmission of the data frame are both latency components which need to be accounted for uniquely as they are not specified by the PC37.244-2013 guideline.

Figures 3.10 and 3.11 illustrate the PDC wait and PDC process latency components specified by the PC37.244-2013 guideline (IEEE Power and Energy Society, 2012).

In the coinciding figures, both the *"no data loss"* and *"data loss"* scenarios are presented. The hashed PMU blocks signify where a data frame was expected to be received by the PDC. The figures emphasize a PDC data processing rate which is higher than the data arrival rate to ensure successful operation.

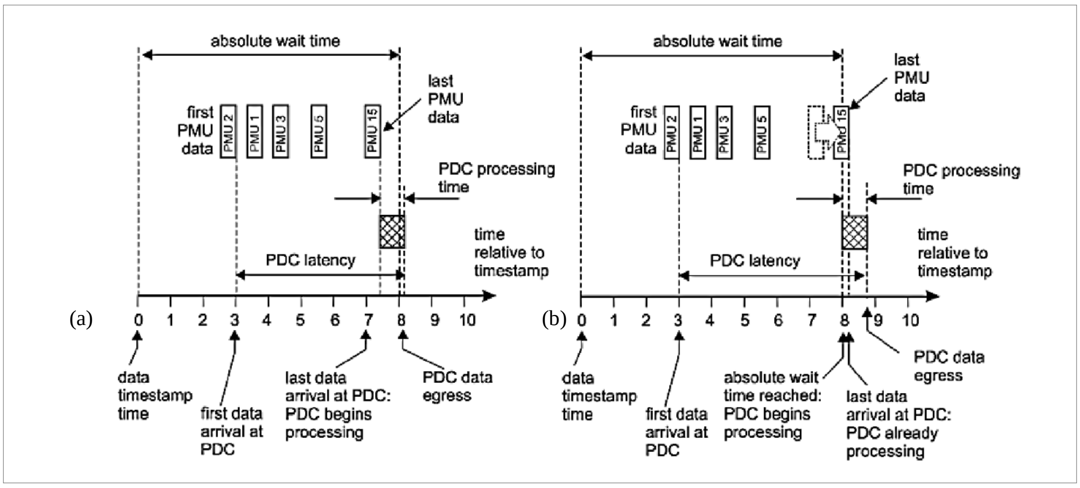


Figure 3.10: PDC latency for data aggregation with time alignment to absolute time (IEEE Power and Energy Society, 2012)

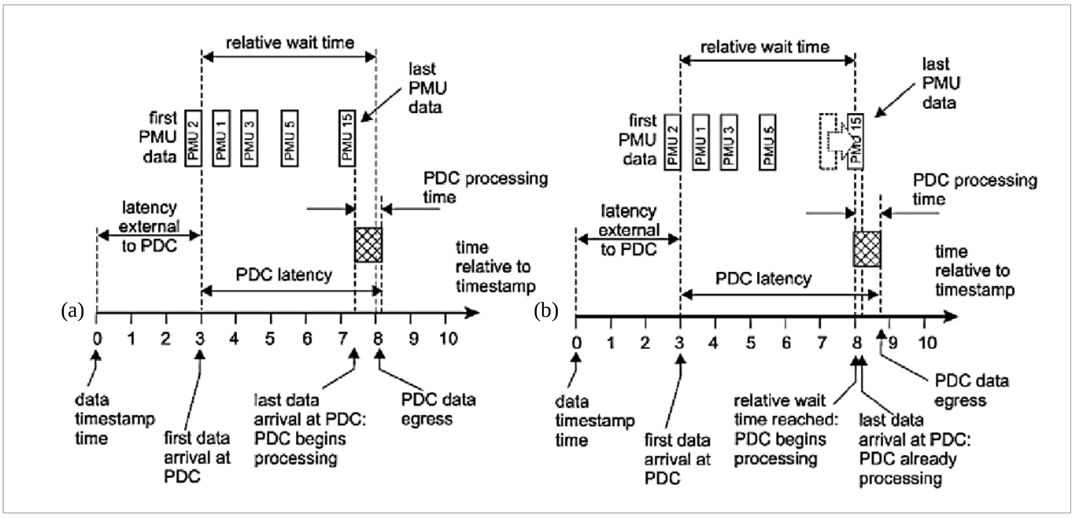


Figure 3.11: PDC latency for data aggregation with time alignment to relative time (IEEE Power and Energy Society, 2012)

The calculation of the processing time is given in Equation 3.10 as the time difference between the last arriving PMU inputs' specified time tag and the corresponding output frame established from matching SOC and FRACSEC time tags within both data frames.

$$T_{process} = T_{otuput} - T_{last\ input} \tag{3.10}$$

Where;

$T_{process}$: Time stamp for processing time

T_{output} : Time of capture of the PDC output

$T_{last\ input}$: Time of capture of the last PMU frame

The total latency of the PDC is given in Equation 3.11 and is calculated similarly to the processing latency, evaluating the time difference between the first arriving PMU inputs' specified time tag and the corresponding output frame established from matching SOC and FRACSEC time tags within both data frames.

$$T_{total} = T_{output} - T_{first\ input} \tag{3.11}$$

The wait time is then approximated as the time difference between the PDC processing time and the total latency of the PDC as shown in Equation 3.12

$$T_{wait} = T_{total} - T_{process} \tag{3.12}$$

Where;

T_{wait} : Wait time

T_{total} : Total latency of the PDC

h. Irregular order of packets:

In theory, PDCs' are expected to receive and process PMU phasor data packets in sequence or consecutively. In practice this is seldom the case as phasor data packets from a PMU can arrive in a non-consecutive order regardless of PMUs' time based sequential processing of the phasor data. Once published to the network, the data packets are susceptible to transposition as network packets prioritize the least congested data transmission channel between the transmitting PMU nodes and receiving PDC nodes.

The received PDC data requires verification and re-ordering of irregular packets based on the GPS UCT time tags. This is achieved by comparing PMU node outputs to respective PDC node inputs.

i. Message framework and configuration:
The phasor data message framework is outlined in great detail in chapter two. The PC37.244-2013 requires the location and size of each message frame field to be verified to ensure accurate translation of information in the data frame.

3.6 PMU-based transmission line current differential algorithm

The PMU-based line current differential protection scheme implements the current and differential trip algorithms from a remote energy management system control station. Once the synchrophasors from both the remote and local terminals are concentrated validated and aligned, the acquired synchrophasor data facilitates current differential evaluation and execution of subsequent current differential protection trip algorithm functions.

Figure 3.12 shows a generalized process flow viewpoint of implementation of the protection algorithms.

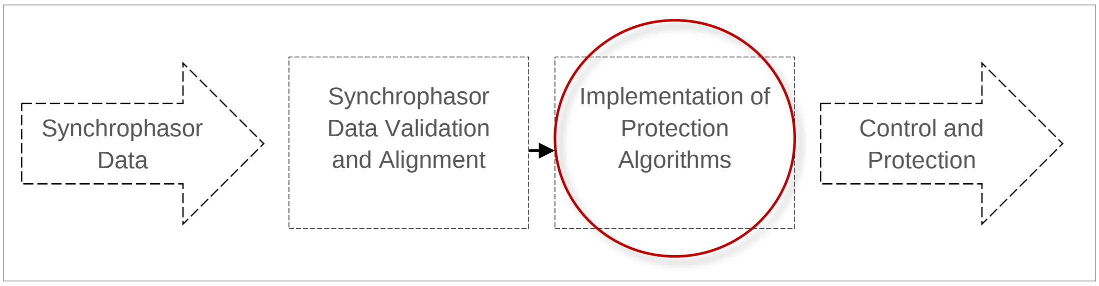


Figure 3.12: Protection algorithm implementation

A software utility is developed precisely for implementing protection and control algorithms in the PMU-based line differential protection scheme. Development of the software utility is analytically detailed in Chapter Four and presented as a key integral deliverable in Chapter Six.

The software utility solution adopts conventional protection and control algorithms utilizing them to establish protection in the PMU-based line differential protection scheme.

Subsections 3.6.1 through 3.6.3 present evaluation of the differential signal, restraint signal and operational characteristics adopted and implemented in a two terminal or two ended PMU-based line differential protection scheme.

3.6.1 PMU-based transmission line current differential signal

The PMU-based transmission line current differential signal is evaluated using an algorithm that employs real-time synchrophasor data from the line terminal PMUs. The C37.118.1-2011 specifies representation of the synchrophasor sinusoidal in the time domain using Equation 3.13

$$i(t) = \bar{\imath} cos(\omega t + \theta) \qquad (3.13)$$

Where;

$\omega = 2\pi f$

$\bar{\imath}$ Represents the phasor magnitude

θ Represents the phasor angle

Analytic representation of Equation 3.13 is given in Equation 3.14, applying Euler's formula and the complex exponent $e^{j\theta}$.

$$\bar{\imath} cos(\omega t + \theta) = \bar{\imath}\left(\frac{e^{j(\omega t+\theta)} + e^{-j(\omega t+\theta)}}{2}\right) \qquad (3.14)$$

Where;

$e^{j\theta} = cos(\theta) + jsin(\theta)$

Considering rectangular form, the phasor is represented by the real part "Re" of Equation 3.14, and is given in Equation 3.15.

$$Re\left\{\bar{\imath}e^{j(\omega t+\theta)}\right\} = \bar{\imath}_m e^{j\theta} e^{j\omega t} \qquad (3.15)$$

Where;

$\bar{\imath}e^{j\theta}e^{j\omega t}$ is the analytical representation of $\bar{\imath} cos(\omega t + \theta)$.

Compact phasor notation of Equation 3.15 is given by Equation 3.16

$$\bar{\imath} \angle\theta \qquad (3.16)$$

Figure 3.13 illustrates evaluation of $i_{\mathrm{DIFF}} = i_1 + i_2$ utilizing line terminal currents from streaming synchrophasor data where i_1 and i_2 represent current phasors from PMU_1 and PMU_2 respectively.

$$i_{\mathrm{DIFF}} = i_1 + i_2$$

$$= \bar{\imath}_1 \angle\theta_1 + \bar{\imath}_2 \angle\theta_2$$

$$= \bar{\imath}_3 \angle\theta_3 \tag{3.17}$$

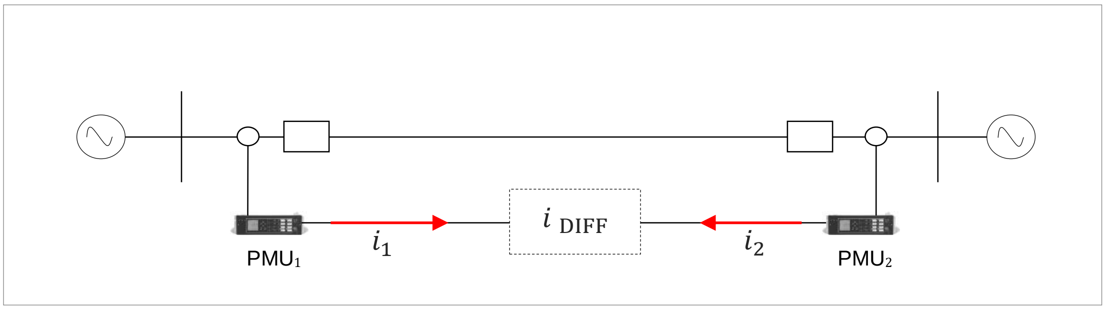


Figure 3.13: Evaluation of differential current in the PMU-based transmission line differential protection scheme

In the evaluation of synchrophasor-based differential current, the phasors i_1 and i_2 are considered instead of current magnitudes represented by $\bar{\imath}_1$ and $\bar{\imath}_2$. The analytic, rectangular representation of $i_{\mathrm{DIFF}} = i_1 + i_2$ follows Equation 3.18;

$$\bar{\imath}_1 \cos(\omega t + \theta_1) + \bar{\imath}_2 \cos(\omega t + \theta_2)$$

$$= Re\{\bar{\imath}_1 e^{j\theta_1} e^{j\omega t}\} + Re\{\bar{\imath}_2 e^{j\theta_2} e^{j\omega t}\}$$

$$= Re\{\bar{\imath}_1 e^{j\theta_1} e^{j\omega t} + \bar{\imath}_2 e^{j\theta_2} e^{j\omega t}\}$$

$$= Re\{(\bar{\imath}_1 e^{j\theta_1} + \bar{\imath}_2 e^{j\theta_2}) e^{j\omega t}\}$$

$$= Re\{(\bar{\imath}_3 e^{j\theta_3}) e^{j\omega t}\} \tag{3.18}$$

Where;

$i_1 = \bar{\iota}_1 \, cos(\omega t + \theta_1)$

$i_2 = \bar{\iota}_2 \, cos(\omega t + \theta_2)$

$Re\{(\bar{\iota}_3 e^{i\,\theta_3})e^{i\omega t}\} = \bar{\iota}_3 \, cos(\omega t + \theta_3) = i_{\mathrm{DIFF}}$

Equation 3.19 presents filtering and magnitude estimation of the differential signal used to qualify internal faults.

$$I_{\mathrm{DIFF}} = |i_{\mathrm{DIFF}}|$$

$$= |\bar{\iota}_3 \, cos(\omega t + \theta_3)| \qquad (3.19)$$

Evaluation of $\bar{\iota}_3$ and θ_3 is independent of ω and t (Lathi, 2004) and is given by equations Equation 3.20 and Equation 3.21 respectively;

$$\bar{\iota}_3{}^2 = (\bar{\iota}_1 \cos\theta_1 + \bar{\iota}_2 \cos\theta_2)^2 + (\bar{\iota}_1 \sin\theta_1 + \bar{\iota}_2 \sin\theta_2)^2 \qquad (3.20)$$

$$\theta_3{}^2 = \arctan\left(\frac{\bar{\iota}_1 \sin\theta_1 + \bar{\iota}_2 \sin\theta_2}{\bar{\iota}_1 \cos\theta_1 + \bar{\iota}_2 \cos\theta_2}\right) \qquad (3.21)$$

Equation 3.20 can be expressed as illustrated in Equation 3.22

$$\bar{\iota}_3{}^2 = \bar{\iota}_1{}^2 + \bar{\iota}_2{}^2 - 2\bar{\iota}_1\bar{\iota}_2 \cos(180 - \Delta\theta) \qquad (3.22)$$

For Equation 3.22 Two case studies are considered.

Case 1: $\Delta\theta = \theta_1 - \theta_2 = 0$

In case 1, there is no angular difference between the phasors from PMU_1 and PMU_2.

$$\bar{\iota}_3{}^2 = \bar{\iota}_1{}^2 + \bar{\iota}_2{}^2 - 2\bar{\iota}_1\bar{\iota}_2 \cos(180 - \Delta\theta)$$

$$= \bar{\iota}_1{}^2 + \bar{\iota}_2{}^2 - 2\bar{\iota}_1\bar{\iota}_2 \cos(\Delta\theta)$$

$$= (\bar{\iota}_1 + \bar{\iota}_2)^2, \text{ as } \cos(0) = 1$$

$$\bar{\iota}_3 = \sqrt{(\bar{\iota}_1 + \bar{\iota}_2)^2}$$

$$\bar{\iota}_3 = \bar{\iota}_1 + \bar{\iota}_2 \qquad (3.23)$$

In the case where $\Delta\theta = 0$, evaluation of the phasor-based differential current is similar to Equation 3.1 (Kasztenny et al., 2011), where only the current magnitudes are considered.

Case 2: $\Delta\theta = \theta_1 - \theta_2 \neq 0$

In case 2, there is an angular difference between the phasors from PMU_1 and PMU_2, which is factored in the evaluation of $\bar{\iota}_3$.

$$\bar{\iota}_3{}^2 = \bar{\iota}_1{}^2 + \bar{\iota}_2{}^2 - 2\bar{\iota}_1\bar{\iota}_2 \cos(180 - \Delta\theta)$$
$$= \bar{\iota}_1{}^2 + \bar{\iota}_2{}^2 - 2\bar{\iota}_1\bar{\iota}_2 \cos(\Delta\theta)$$

$$\bar{\iota}_3 = \pm\sqrt{\bar{\iota}_1{}^2 + \bar{\iota}_2{}^2 - 2\bar{\iota}_1\bar{\iota}_2 \cos(\Delta\theta)} \tag{3.24}$$

$\Delta\theta$ plays a very active role in the evaluation of the differential current utilizing synchrophasors. For the case where $\Delta\theta = 0$ evaluation of the current differential conforms to the conventional algorithm as proven in Equation 3.23. For the case where $\Delta\theta \neq 0$, the differential current is evaluated using Equation 3.24.

3.6.2 PMU-based line current restraint signal

A percentage differential characteristic is applied to pre-emptively mitigate the non-zero differentials that may occur (Altuve et al., 2004). The restraining signal I_{RES} in a two ended transmission line is denoted by Equation 3.25.

$$\begin{aligned} I_{\mathrm{RES}} &= |i_1| + |i_2| = |i_3| \\ &= |\bar{\iota}_1 \,cos(\omega t + \theta_1)| + |\bar{\iota}_2 \,cos(\omega t + \theta_2)| \\ &= |\bar{\iota}_3 \,cos(\omega t + \theta_3)| \qquad (3.25) \\ &= |\bar{\iota}_3 \angle\theta_3| \\ &= |\bar{\iota}_3||cos(\theta_3)| \end{aligned}$$

3.6.3 PMU-based line operating characteristic

The differential trip logic is triggered when the differential signal exceeds a predetermined PV and a specified percentage of the restraining value (K). Equation 3.26 represents the monitored condition.

$$I_{\text{DIFF}} > PV + K \bullet I_{\text{RES}}$$

$$|\bar{\imath}_3 \, cos(\, \omega t \; + \; \theta_3)| \; > \; \{\, PV + [\, K(|\bar{\imath}_1 \, cos(\, \omega t \; + \; \theta_1)| + |\bar{\imath}_2 \, cos(\, \omega t \; + \; \theta_2)|)]\} \quad (3.26)$$

Where $\bar{\imath}_3$ is given by Equation 3.23 or Equation 3.24 and θ_3 by Equation 3.21. Implementation of the proposed algorithm and evaluation of the differential current at any specified moment is as follows;

1. Acquisition of synchrophasors ($\bar{\imath}_1, \theta_1$ and $\bar{\imath}_2, \theta_2$) from line terminal PMUs.
2. Establishing phase angle difference ($\Delta\theta$) between the line terminal PMUs.
3. Calculating the differential current (I_{DIFF}).
4. Testing for fault conditions ($I_{\text{DIFF}} > PV + K \bullet I_{\text{RES}}$).
5. Implementation of the protection scheme.

The algorithm flow chart is presented in Figure 3.14.

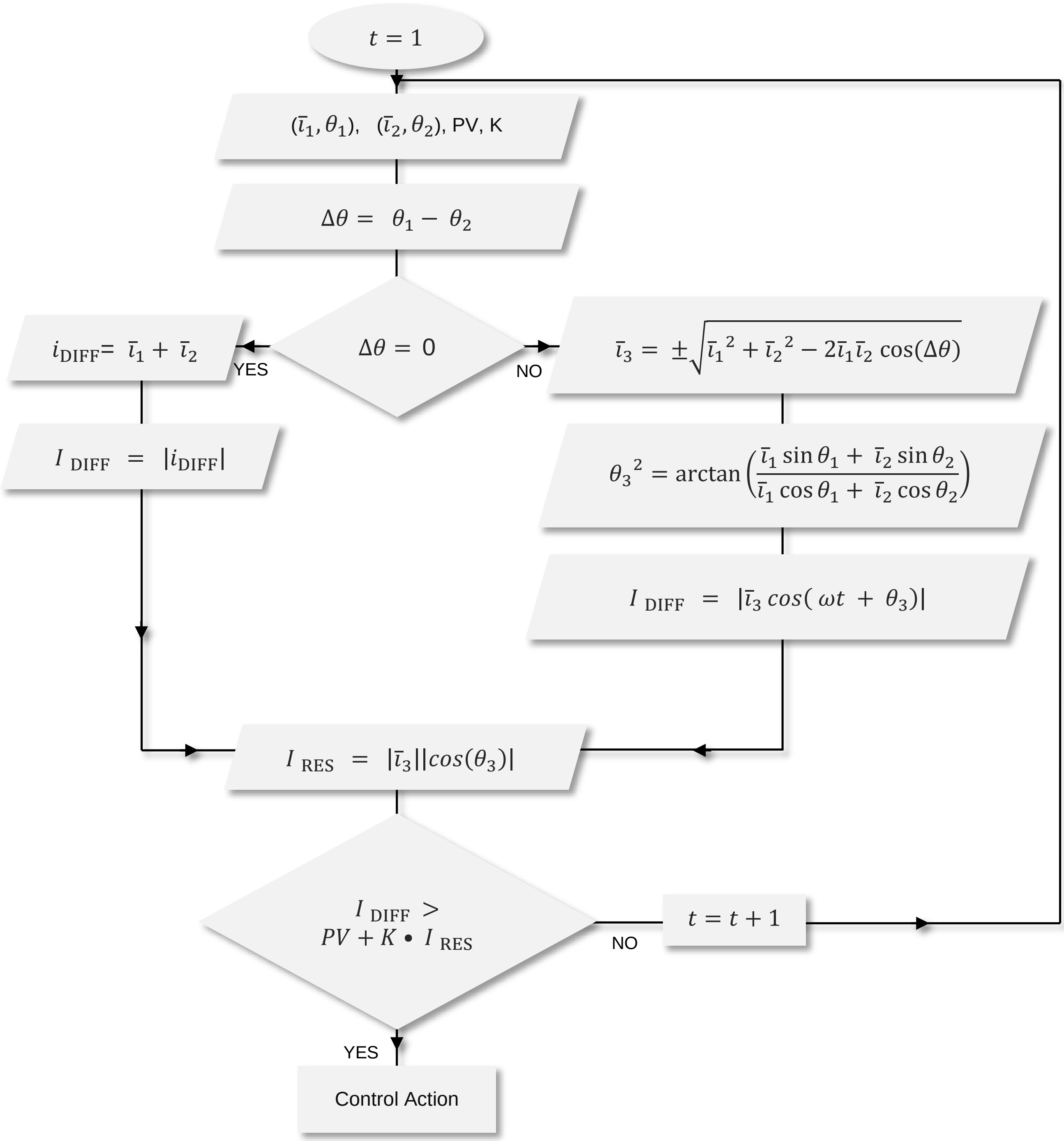


Figure 3.14: Synchrophasor-based, differential current algorithm

3.7 Implementation of protection functions in the PMU-based protection scheme

Once the differential trip algorithms evaluate the differential current signals, the differential signals are subsequently utilized to identify fault occurrence and location. Figure 3.6 presented in Section 3.4 illustrates the concept of implementing isolated circuit breaker control to administer control and protective functionality in the PMU-based transmission line differential protection scheme. If an internal fault is identified, a DTT signal is issued to control the line terminal circuit breakers. Figure 3.14 presents block diagram for circuit breaker control and DTT signal issuing. In Chapter Five, the Circuit breaker logic control is discussed in greater detail.

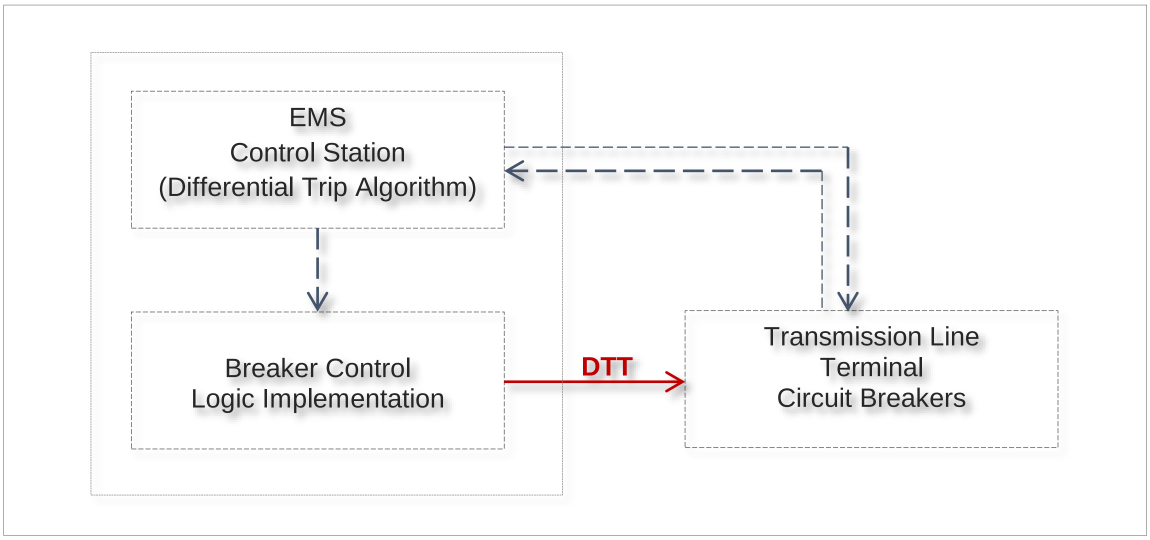


Figure 3.15: PMU-based protection scheme, circuit breaker control and DTT signal issuing

3.8 Discussion

Considering modern design criteria, contemporary transmission line current differential protection schemes provide essential groundwork in developing the proposed PMU-based protection scheme. The design methodologies adopted from conventional transmission line differential protection schemes provide critical outline for marginal operation and standard performance requirements.

3.9 Conclusion

This Chapter covers theoretic considerations for modeling line current differential protection schemes. The established groundwork for conventional protection schemes outlines a clear guideline for pragmatic considerations in modeling the PMU-based protection scheme. Chapter Four details design and development of the software utility specific to control and protection of the PMU-based protection scheme.

Chapter 4 :
Engineering of the Phasor Current Differential Action Adapter (PCDAA)

4.1 Introduction

The engineering process for the PMU-based protection scheme takes into account design considerations necessary to implement synchrophasor-based transmission line differential protection. Once aligned, real-time synchrophasor measurement data streamed from line terminals require concentration prior further application specific processing.

The Phasor Current Differential Action Adapter (PCDAA) software utility is designed to concentrate synchrophasor data and administer protection and control functions in the PMU-based transmission line differential protection scheme. This Chapter describes the engineering process for the PCDAA, presenting the synchrophasor utility software solution.

This Chapter outlines and discusses considerations for developing real-time synchrophasor data processing software specific to the PMU-based transmission line differential protection scheme. Common drawbacks in developing phasor utility software are highlighted, reviewing IEEE standards that identify protocol guidelines to mitigate these drawbacks.

This Chapter also establishes the necessity to implement contemporary resources and modern software development tools, using the IEEE standard guidelines to engineer an independent, industrial grade phasor data utility software to provide an interoperability solution that;

- Adheres to the IEC 61850 communication standard for electrical substation automation.
- Complies with the C37.238-2011 Precision Time Protocol (PTP).
- Follows guideline specifications outlined by the C37.118-2011 standard for synchrophasors.

4.1.1 Chapter organization

Section 4.2 discusses design considerations for a real-time synchrophasor data processing software utility.

Section 4.3 identifies engineering resources, introducing the OpenPDC and the Grid Solution resources utilized for modelling the PCDAA utility software.

Section 4.4 describes engineering of the PCDAA software utility.

Section 4.5 details application of PCDAA software utility.

Section 4.6 discusses the engineering process and outlines operational expectations of the PCDAA.

Section 4.7 concludes the Chapter.

4.2 Developing real-time synchrophasor data processing software

A Phasor Data Concentrator (PDC) is a software system that process streaming synchrophasor data in real-time. Conventional PDC-PMU networks route streaming synchrophasor data to PDCs in central locations via existing networks such as those based on the Inter-Control Center Communication Protocol (ICCP) (CompuSharp, 2012). Figure 4.1 illustrates a typical PMU-PDC network.

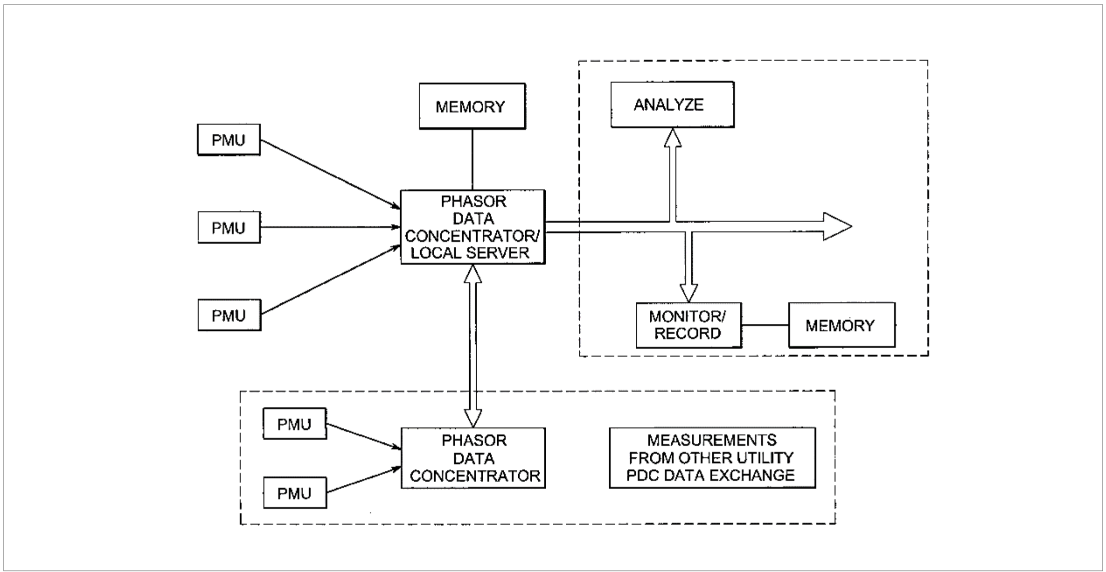


Figure 4.1: System and method for synchronized phasor measurement (Premerlani et al., 2006)

Operation of the PMU-based transmission line differential protection scheme depends on the schemes' ability to process real-time phasor data. This Section discusses design considerations for developing the synchrophasor data processing software.

4.2.1 Communication and synchrophasor data transmission

This Section considers practical implementation of the PMU-based protection scheme in the Center for Substation Automation and Energy Management Systems (CSAEMS). The PMU-based protection scheme relies on communication, and real-time data transmission of synchrophasors. In this regard the Real Time Digital Systems Simulator (RTDS) Ethernet-based communication network is examined. The Ethernet communication media provides a guaranteed bit rate that remains susceptible to errors, therefore, it is necessary for the synchrophasor data processing software utility to mitigate these errors to establish minimal delays in real-time synchrophasor data stream applications (Farrell & Ong, 2000), (Hou et al., 2006), (Li et al., 2011) and (Y. Liu et al., 2012).

The CSAEMS RTDS network administers the First-in, First-out (FIFO) queuing system (Cohen & Ofek, 1995), (Sathaye et al., 1994) and (Le et al., 2013). The FIFO queuing system detects and alleviates network congestion by reducing the data transmission rate, split streaming the bandwidth. For real-time and Constant-Bit-Rate (CBR) data transmission, intermittent data loss is inevitable as packets will be dropped from the tail of the queue when transmission rates are decreased to accommodate congestion. Employing the Transmission Control Protocol (TCP) to administer real-time and CBR data transmission partially alleviates data loss but greatly increase the transmission delays. Employing User Datagram Protocol (UDP) disregards acknowledgement of transmitted data arrival but maintains a large buffer of previously sampled data utilizing timestamps and sequence numbers, to evaluate data loss and verify transmitted data to successfully mitigate intermittent data loss.

Layered protocols specified by standards listed in Section 4.22 mitigate transmission errors, encapsulating frames into a protocol to deter transmission of synchrophasor data frames directly over a network. The encapsulated frames conform to standard specified protocols, which guarantees reliability and successful transmission.

4.2.2 Real-time, synchrophasor data concentration

IEEE standards and protocol specifications considered for developing an Industrial grade PDC software utility include;

a) IEEE C37.118-2011:

The IEEE C37.118 standard for synchrophasors is described in Chapter Three, Section 3.5.

b) IEC 61850-90-5:

The IEC TR 61850-90-5 protocol provides a specification guideline for digital state transmission and time alignment of power measurements over wide area networks. The IEC TR 61850-90-5 facilitates implementation of Wide Area Measurement Protection and control (WAMPAC) systems based on the IEC 61850 protocols (Lee et al., 2013), (Ali et al., 2015) and (Madani et al., 2015).

c) IEEE 1344-1995:

IEEE 1344 was succeeded by IEEE C37.118 in 2005, adopting the time extensions as part of the IRIG timing standard. The IEEE 1344 standard defines synchrophasor parameter specifications for power systems. The standard extends IRIG-B time code to cover year, time quality, daylight saving time, local time offset and leap second information. (IEEE Power Engineering Society, 2005).

4.2.3 Graphic User Interface (GUI)

The GUI employs a set of graphical elements that provide an interface for issuing commands related to configuring and managing the synchrophasor processing utility software. The GUI is essential to enable flexible user interaction with the software engine to provide easy control and access related to;

i. connection to multiple PMUs
ii. concentrations of synchrophasor data streams
iii. real-time synchrophasor data stream application specific processing

(Hu & Ji, 2008) emphasize GUI design practice that extends utility, in this respect, a historian is considered to archive concentrated synchrophasors. Archived time-stamped synchrophasor data potentially facilitates post disturbance and trend analysis. A GUI also provides visual tools, enabling the user to efficiently monitor the system and interact with external applications critical to establishing protection and control functions.

4.2.4 Discussion on development considerations

(Armenia & Chow, 2010) identify and implement contemporary resources to realize a modern PDC software utility that provides an interoperable and interchangeable industrial grade solution.

Software development resources identified include the IEEE standards;

a) IEEE C37.118-2011
b) IEC 61850-90-5
c) IEEE 1344-1995

Open-source Ruby and the Rails model-view-controller framework resources are also utilized (D. H. Hansson, 2010). (Armenia & Chow, 2010) administer resources emphasizing how the resources themselves form elementary software engineering units that enable development of independent PDC software utilities that comply with industrial Smart Grid standards.

Common drawbacks faced in the development of synchrophasor processing software include dealing with the Ethernet communication media errors which results intermittent data loss (Farrell & Ong, 2000). IEEE standards provide cyclic redundancy checks and layered protocols to successfully mitigate these drawbacks and ensure no data corruption.

4.3 Engineering resources

Design considerations presented in Section 4.2 provide a guideline for engineering the utility software required to establish real-time processing of synchrophasors in the PMU-based transmission line differential protection scheme. The OpenPDC and the Grid Solutions frameworks are identified as relative contemporary resources. These resources are drawn on to facilitate effective development of the PMU-based protection schemes' synchrophasor processing utility software. The resources are selected as they objectively underline adherence to standard guidelines for synchrophasor processing which include; the C37.118-2011 and the IEC 61850-90-5.

4.3.1 OpenPDC

Once a PDC concentrates phasor data based on Global Positioning System (GPS) time, it provides time-synchronized datasets for further. OpenPDC is a high-performance data concentrating platform that manages streaming synchrophasor and time-series data in real-time (CodePlex, 2016). OpenPDC is open source, which provides excellent flexibility with application specific design considerations. The OpenPDC framework provides real-time distribution of synchrophasor data at station-level within the substation environment. OpenPDC architecture is based on the Grid Protection Alliance (GPA) Time-Series library that inherits modular design that enables it to be classified as a generic event stream processor (Grid Protection Alliance, 2014) and (CodePlex, 2016).

The OpenPDC supports multiple phasor protocol standards which include; IEC 61850-90-5, IEEE C37.118, IEEE 1344, F-NET, Macrodyne, BPA PDCstream and SEL Fast Message. Key features of the OpenPDC include:

a. Analysis performance statistics and time code errors.
b. Phasor data transformation with configurable output streams.
c. A generic configuration database that supports Microsoft SQL Server and Oracle.
d. Flexible deployment in Portable Operating System Interface (POSIX) and Windows Operating System (WOS) environments.

Figure 4.2 shows the OpenPDC modular system design for managing streaming measurements adopted from (CodePlex, 2016).

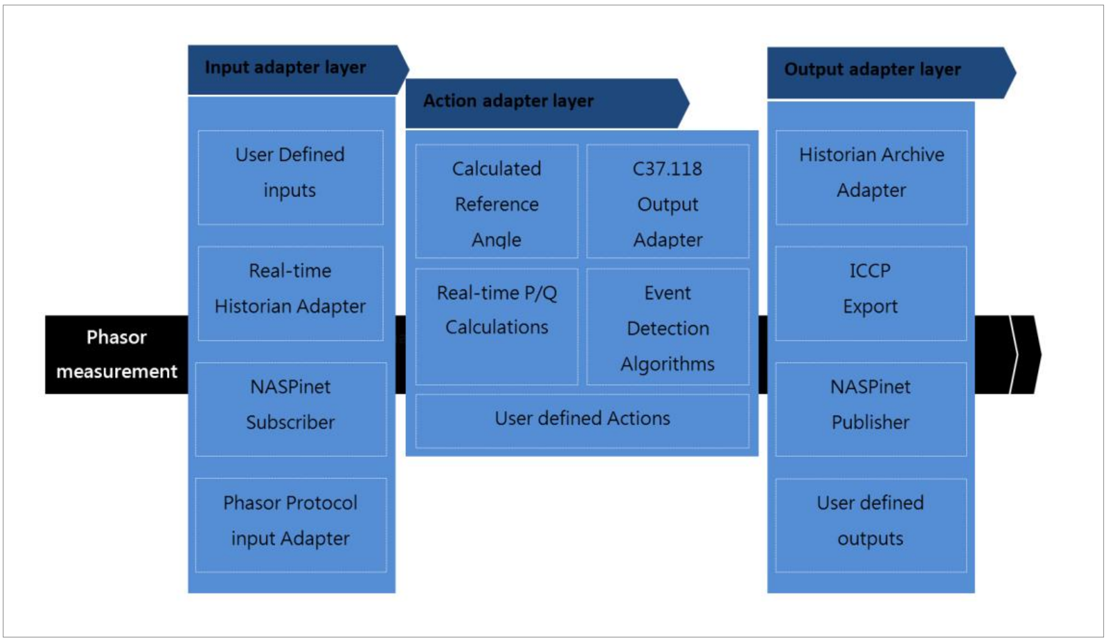


Figure 4.2: Streaming phasor measurement data management (CodePlex, 2016)

4.3.2 Grid Solutions

Grid Solutions Framework:

The Grid Protection Alliances (GPA), Grid Solutions Framework (GSF) provides extensive open source .NET code, specific to electric power utilities like PMUs and PDCs (Grid Protection Alliance, 2014). The Grid Solutions Framework and implementation platform is presented in Figure 4.3.

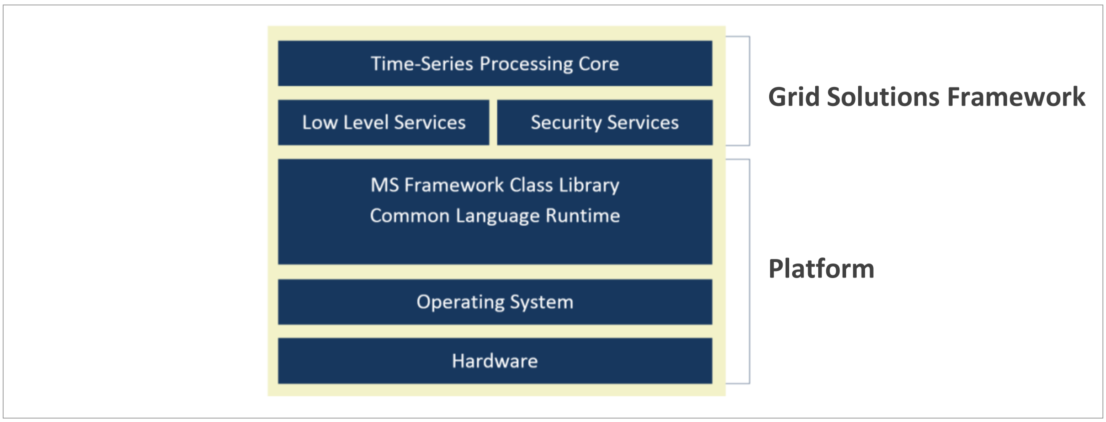


Figure 4.3: Grid Solutions Framework and Implementation platform

The GSF facilitates critical function class libraries that administer functionality included in the .NET Framework, simplifying the more complex features including socket connection and encryption. More importantly, the GSF framework facilitates utilization of add-ons enabling addition and use of functions not included in the .Net framework like the Time-Series library and TVA-Code Library from CodePlex. The add-on libraries are refactored using .Net 4.5 to mitigate framework security and provide excellent performance. Table 4.1 shows some of the GSF libraries including the Time-Series library.

Table 4.1: GSF Libraries

Grid Solution Frame-work Class functions library (.gsf)	Version utilized
gsf.core	2.1.90
gsf.security	2.1.90
gsf.timeseries	2.1.90
gsf.libraries.core	2.0.203
gsf.serviceprocess	2.1.90
gsf.communication	2.1.90
gsf.phasorprotocols	2.1.90
gsf.libraries.security	2.0.203
gsf.libraries.timeseries	2.0.203
gsf.libraries.serviceprocess	2.0.203
gsf.libraries.communication	2.0.203

Figure 4.4 provides an overview for GPA Time-Series Library. The GPA Time-Series Library is a part of the GSF.

Figure 4.4: GPA Time-Series Library (Grid Protection Alliance, 2014)

With reference to Figure 4.3;

a) Input adapter layer (MAP)

This layer provides protocol parsers that facilitate acquisition of synchrophasor data. The input adapters are flexible and allow new protocol parsing support to application specific systems.

b) Action adapter layer (SORT)

The action layer enables time-aligned analytics and calculations at sub-second intervals. Real-time calculation results are made available to output streams and other calculations at their full sample rate.

c) Output adapter layer (QUEUE)

This layer buffers streaming time-series data for output to data archiving systems.

4.4 Engineering of the PCDAA software utility

The OpenPDC and Grid Solutions resources are used to engineer the PCDAA software utility. This Section defines the aim and objective of the utility software emphasizing the theoretic expectations of the final solution. It also introduces the Real Time Digital Simulator (RTDS) GTNET component which is used to simulate the PMU models used in the research.

4.4.1 Aim

The aim of the developed PCDAA utility is to facilitate an interoperable software solution that establishes communication between remote PMUs and an energy management system control station, adhering to the IEC 61850 communication standard and following specification guidelines outlined by the C37.238-2011 Precision Time Protocol (PTP).

4.4.2 Objective

The objective of the PCDAA software utility is to process real-times synchrophasor data streams from multiple PMUs to administer current differential protection for the PMU-based transmission line differential protection scheme. Figure 4.5 presents a flow chart representing operation and conceptual expectations of the interoperable software solution.

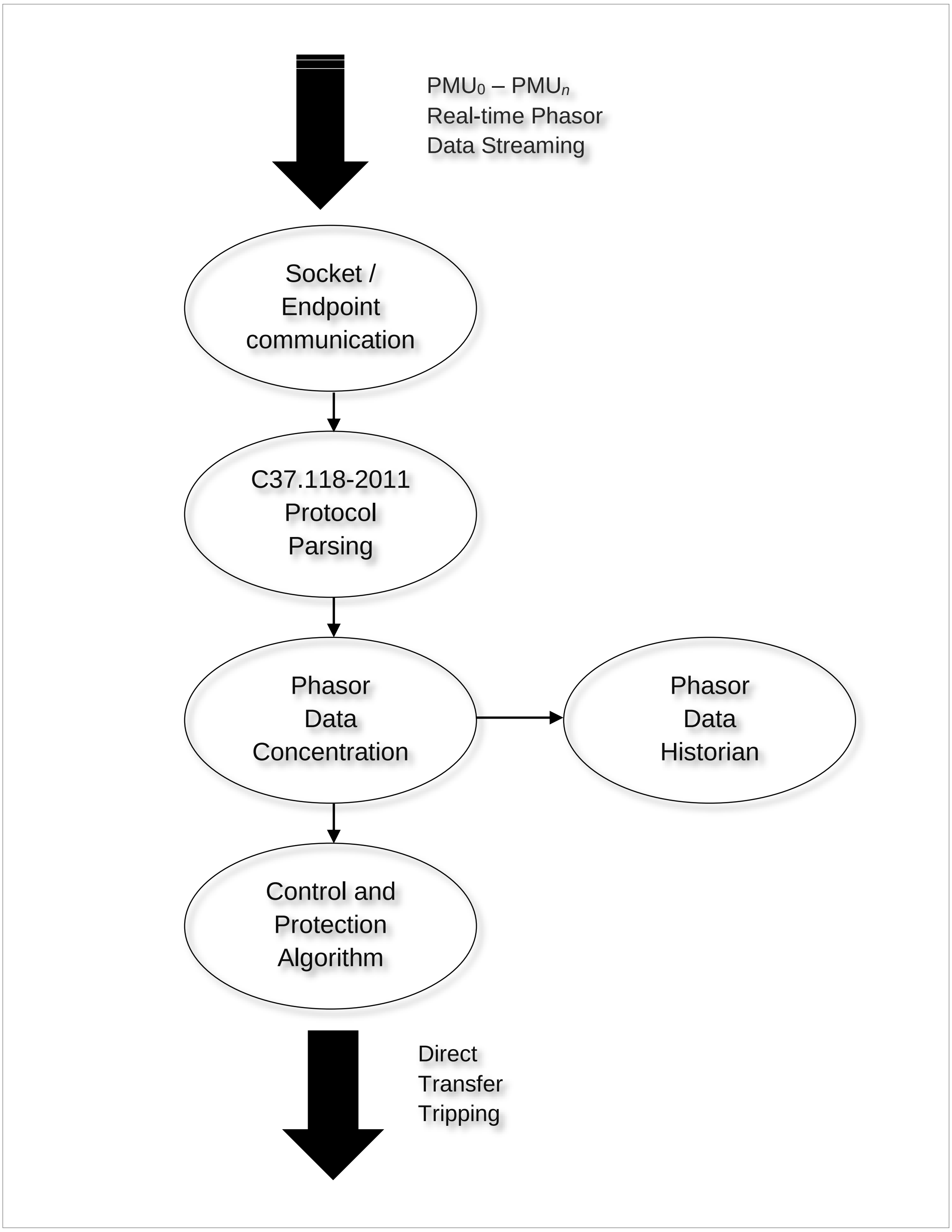


Figure 4.5: Proposed PCDAA real-time data process flow chart

With reference to Figure 4.5, all the PMU Logical Nodes (LN) synchronously stream real-time synchrophasor data over the distributed network architecture in which each PMU coherently transmits data over a bidirectional socket that connects it to the PCDAA utility software. When data from the multiple PMUs is received, the incoming message-frames (outlined in Chapter Three, Section 3.5) are parsed using the C37.118.1-2011 standard for synchrophasors.

Once the incoming data is interpreted, it is then concentrated following specifications outlined in the C37.118.2-2011 guideline for phasor data concentration. After PMU data is verified and aligned, it is archived and further processed to subsequently administer control and protection functions in the PMU-based transmission line differential protection scheme. Figure 4.6 presents the modelled software architecture, based on the utility description presented by the flow chart in Figure 4.5.

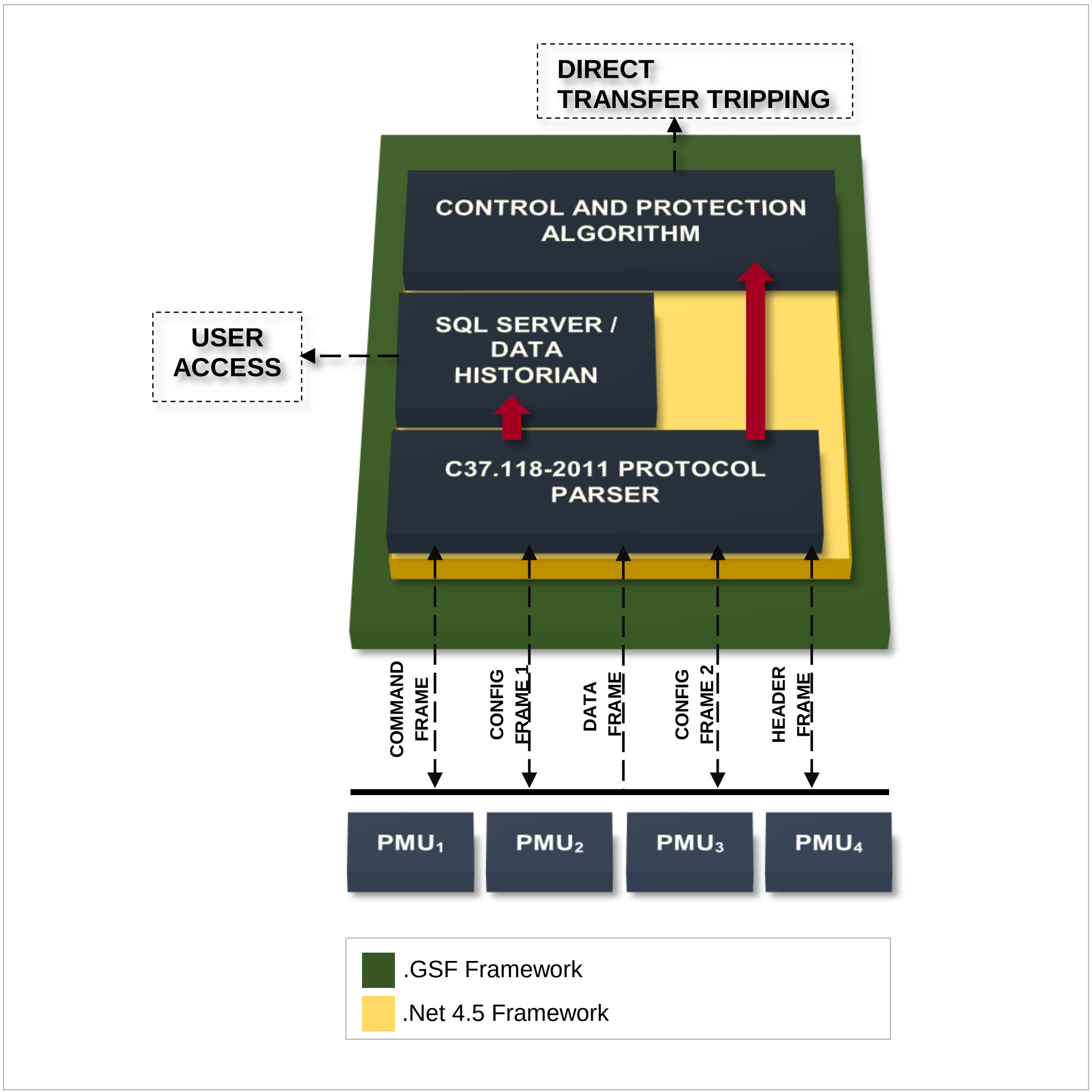


Figure 4.6: PCDAA software architecture

4.4.3 Limitations

- The scope of the research is limited to a two ended transmission line. Although the modelled PCDAA software architecture considers and gives provision for multiple PMUs, the scope of the research is limited to two PMUs, respective to the two transmission line terminals.

- The research is limited to the RTDS GTNET PMU_v4 component model detailed in Chapter Five. The GTNET PMU_v4 component model (RTDS-Technologies, 2013) is used to simulate the two PMU LNs.

4.4.4 Software development

The software development is detailed here, outlining the development tools pertaining to the development of the PCDAA software utility. It is necessary to mention that an abstraction approach is employed to develop component models based on utility features illustrated in the software architecture in Figure 4.6.

Considering the Windows Operating System (WOS)-based Test-bench PC utilized in the CSAEMS laboratory, the Microsoft .NET 4.5 software is installed to facilitate developing a .Net 4.5 framework-based software solution. The .NET framework software environment facilitates appropriate runtime requirements for developing the PCDAA software. Figure 4.7 illustrates the used .Net 4.5 framework (Lam & Thuan, 2008).

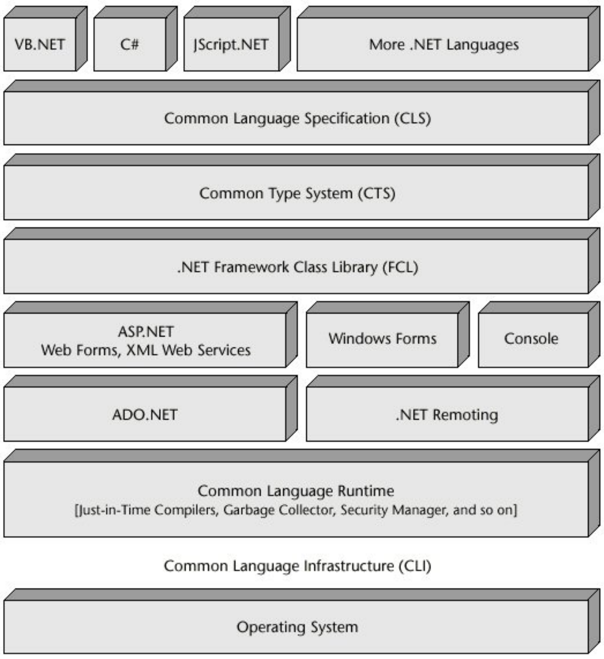


Figure 4.7: .Net 4.5 Framework (Lam & Thuan, 2008)

4.4.4.1 Methodology

Feature Driven Development (FDD) (Agile Modeling, 2016) is used for developing the software solution for implementing synchrophasor-based transmission line differential protection in this research. FDD build-by-feature process flow is employed, enabling application of logical concepts in short, iterative, feature-driven, life cycles. Figure 4.8 illustrates the build-by-feature process flow.

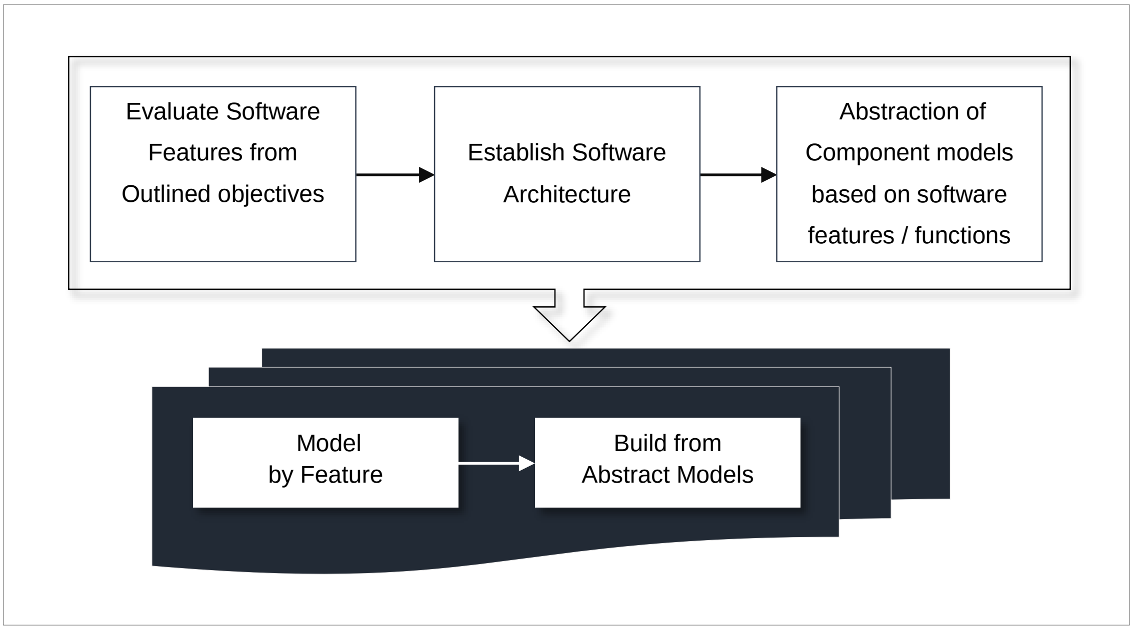


Figure 4.8: Process of feature driven development adopted from (Agile Modeling, 2016)

The csharp (C#) programming language is used with the Microsoft Visual Studio (MSVS) Integrated Development Environment (IDE) (Burrows, 2010) and (Microsoft, 2015). Together with the OpenPDC and GPA resources identified in Section 4.3, MSVS IDE is used to develop the PCDAA utility software. Figure 4.9 illustrates the MSVS IDE used to engineer the PCDAA. Considering efficient development of the software utility, important benefits of MSVS IDE that synergize with the FDD methodology include;

- Unit testing
- Diagnostics Tools
- UI Debugging Tools

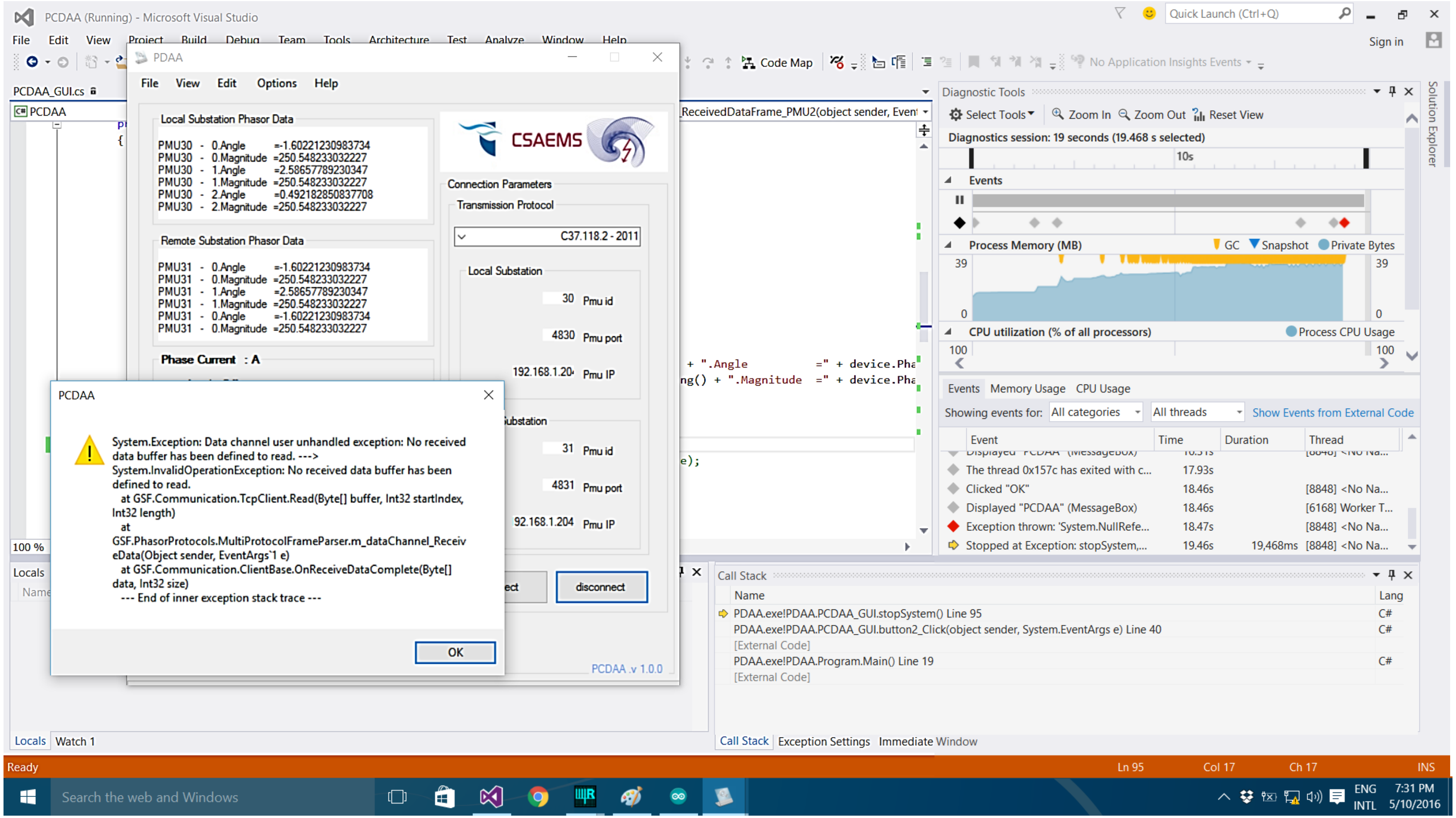


Figure 4.9: MSVS development environment and engineering the PCDAA

Employing the FDD methodology and utilizing the GPA .GSF library plug-in support, enables engineering of source control systems which facilitates advanced abstract end point communication development (Grid Protection Alliance, 2013).

4.5 Application of the PCDAA software utility

4.5.1 PCDAA Specifications

The PCDAA utility software is specifically designed to facilitate interoperability between the transmission line PMUs and the Energy Management System (EMS) control station, administering monitoring, control and protection functions in capacity of the PMU-based line current differential protection scheme. Specifications of the PCDAA utility software include;

- Real-time communication:
  The developed software utility is expected to establish real-time communication and phasor data exchange utilizing socket or endpoint bidirectional inter-process communication flow.

- Real-time synchrophasor concentration:
  The PCDAA PDC module is developed using guidelines specified by the C37.118 – 2011. The module is expected to establish real-time synchrophasor data concentration and verification to facilitate precise evaluation of transmission line current differentials.

- Real-time current differential algorithm implementation:
  With real-time coherent and accurate synchrophasor data transmission and concentration the PCDAA is expected to implement protection and control algorithms in real-time.

- Accurate Direct Transfer Tripping (DTT):
  With real-time procession of synchrophasor data and implementation of protection and control algorithms the PCDAA is expected to provide accurate (DTT).

The PCDAA software utility processes phasor data in real-time. Figure 4.10 presents an overview of PCDAA software module based on the outlined specifications.

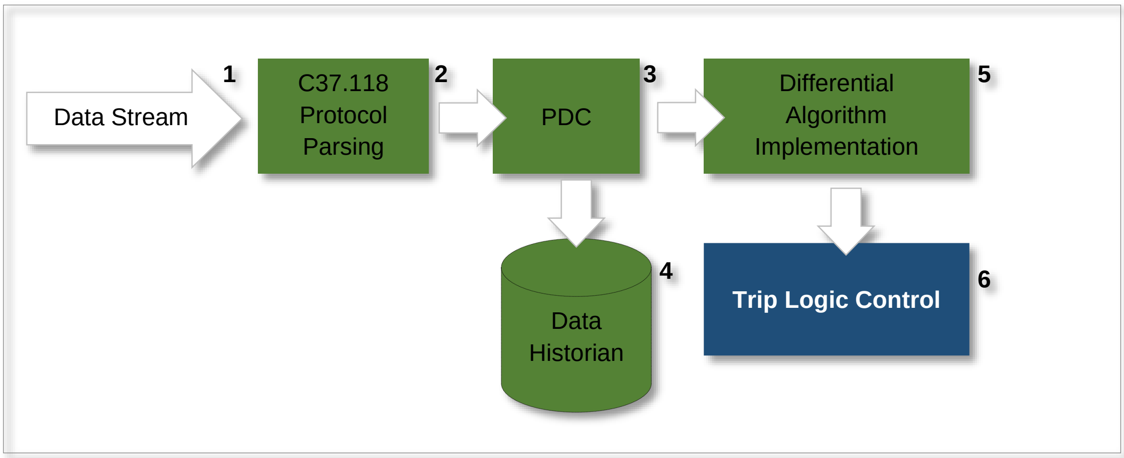


Figure 4.10: PCDAA software specification overview

Table 4.2 provides a description to the labels in Figure 4.10.

Table 4.2: Figure 4.10 label description

Figure Label	Description
1	PMU real-time synchrophasor data streams
2	protocol parsing
3	phasor data concentration and verification
4	data archiving
5	control and protection algorithms implementation
6	Direct Transfer Trip (DTT) function

4.5.2 PCDAA User interface

The final WOS based GUI for the developed PCDAA is presented in Figure 4.11. The PCDAA GUI provides;

- File menu:
  A graphical toolbar which contains file-handling commands relating PCDAA utility software

- Time display:
  Observing PC37.244 guidelines, a global time source is used to verify and validate the concentrated phasor data. This GPS time-stamp is displayed here. The laboratory Test-bench local time and system frequency are also displayed for monitoring.

- Phasor data display screens:

  The concentrated synchrophasor data streams from the transmission line terminal PMUs is displayed here. The transmission line phase currents are monitored, displaying the magnitude and angular components for each phase.

- Differential Current Displays:

  The differential current is computed for each phase, once evaluated the current differential is used to monitor, control and protect the transmission line. The angle component also facilitates stress indication.

- Connection parameter specification:

  This field specifies network identification for the line terminal PMUs. It also facilitates specification of the phasor data communication protocol.

- Connect / Disconnect control button:

  To connect or disconnect the software utility to the remote PMUs specified in the connection parameters.

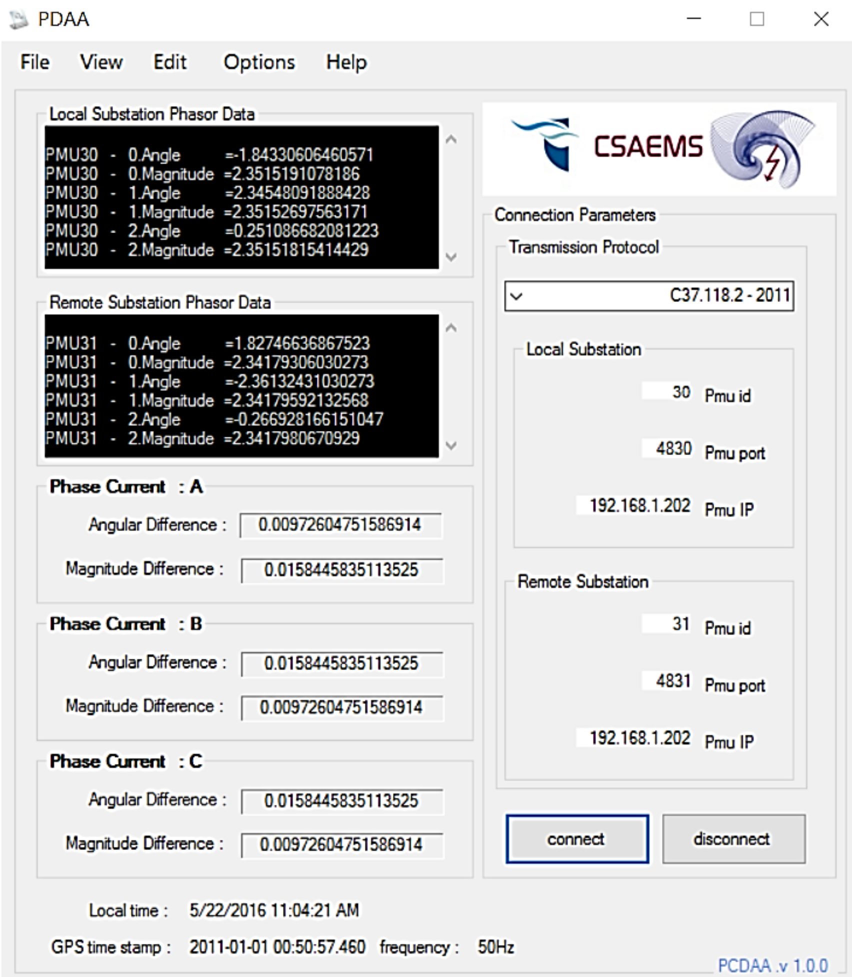


Figure 4.11: PCDAA WOS based GUI

4.6 Discussion

The OpenPDC and GPA resources are used together with MSVS to develop the protection software solution for the PMU-based line differential protection scheme. The PCDAA is an important independent research contribution which aims to provide an interoperability solution that complies with the IEC 61850 communication standard for electrical substation automation. The software utility is engineered to processes transmission line terminal synchrophasor data streams in real-time, following guideline specifications outlined by the C37.118-2011 standard for synchrophasors. Criteria outlining the design considerations for developing the software utility is based on C37.238-2011 Precision Time Protocol (PTP) which identifies and acknowledges practical application of synchrophasors in protection schemes.

The source code of the PCDAA is provided in Appendix A.

4.7 Conclusion

This Chapter provides a software design outline for engineering of a protection solution that facilitates practical implementation of the PMU-based transmission line differential protection scheme. The developed PCDAA is presented in Chapter Six as part of the results. Validation of real-time synchrophasor data processing utility is verified in the case studies where the PMU-based differential protection scheme is successfully implemented using the PCDAA.

Chapter Five details the Test-bench design, setup and configuration in the CSAEMS laboratory.

Chapter 5 :
Test bench design

5.1 Introduction

The proposed PMU-based protection scheme is a potential alternative transmission line differential back-up protection solution. This Chapter describes a laboratory scale validation method for the hypothesis, emphasizing characteristic analysis of the PMU-based transmission line differential protection scheme. The protection scheme test methods identified in the literature review are adopted and utilized to implement contemporary methods for evaluating protection functions.

The objective behind the design of the Test-bench is to verify the hypothesis, quantitatively evaluating the protection scheme characteristics with regard to; speed, security and dependability. This Chapter details design and development of the laboratory scale Test-bench.

5.1.1 Chapter organization

Section 5.2 describes development of Center for Substation Automation and Energy Management Systems (CSAEMS) laboratory scale Test-bench, highlighting the limitations and dependencies of the Test-bench model. This Section also describes specifications for the RSCAD system model components for real-time simulation.

Section 5.3 details simulation case studies.

Section 5.4 discusses design considerations of the Test-bench model.

Section 5.5 concludes the Chapter.

5.2 Test-bench development

5.2.1 Dependencies

This Section presents hardware and software utilized in the design of the laboratory scale Test-Bench.

5.2.1.1 Hardware: Real-Time Digital Simulator

A transient, Real-Time Digital Simulator (RTDS) is used to simulate the real-time power transmission line system, over which, the PMU-based protection scheme is evaluated.

The RTDS provides an efficient simulation environment that enables real-time simulation study on characteristic transient response of protection scheme protective and control functions. Its' flexible control allows extensive and repeatable testing of protection scheme functions (RTDS-Technologies, 2013).

5.2.1.2 Hardware: Ethernet switch

A PT-G7509 Series, which is an IEC 61850-3 and IEEE 1613 compliant industry specific network switch is used to bridge and inter-connect the Test-bench network devices. The network switch provides support for priority-tagged frame processing (described in Chapter Three, Section 3.5.3) and data transmission to the control station, utilizing addresses to process and forward data at the data link layer level (Moxa, 2004).

5.2.1.3 Hardware: Control station

An HP laptop is utilized to manage and administer Test-bench interface software which includes, the Phasor Current Differential Action Adapter (PCDAA) and RSCAD. The interface software follows guidelines specified by the C37.238-2011 to establish a communication network that employs monitoring, protection and control utility to equip actionable information and extend configuration settings to all Test-bench system components. An overview of the PC specifications is presented in Table 5.1.

Table 5.1: Intel desktop PC specifications

Part - Description	Specifications	
Operating System	Windows 10 Professional	: **x64 bit Architecture**
Memory	DDR3 SDRAM (1333 Mb/s)	: **8 gigabyte**
Processor	Intel $^{(R)}$ Core $^{(TM)}$ i7 (2.30 GHz)	: **4712 MQ**
Hard-drive	Solid State Drive (SSD)	: **KC300**

Processor speed:

RSCAD and the PCDAA interface utilities require processor power adequately contented by the control station hardware. The x64 Windows Operating System (OS) architecture facilitates data processing up to 64 bits of information per clock cycle (Gwennap, 2016).

Random access memory (RAM):

The operational frequency of the PCs' RAM reduces the test benches' overall performance. Once data cache initializes, the constant transfer of data between RAM and virtual memory slows the performance of the Test-bench PC considerably (JEDEC, 2012).

5.2.1.4 Software: RSCAD

RSCAD is used to interface RTDS hardware, creating the real-time transmission line system simulation model. The RSCAD interface software enables analysis of simulation results, supporting quantitative deductions to be drawn efficiently. Figure 5.1 illustrates RSCAD File system interaction.

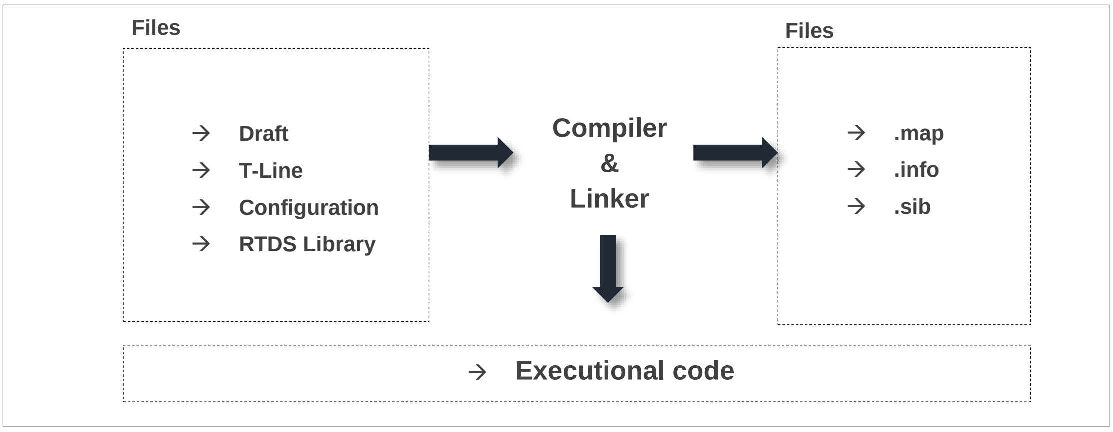


Figure 5.1: RSCAD File system Interaction (RTDS-Technologies, 2013)

Table 5.2 provides short descriptions of the main RSCAD modules (RTDS-Technologies, 2013).

Table 5.2: RSCAD Modules

Module	Description
Draft	The draft module provides a graphical interface and component configuration.
Cable	The cable module computes and manages cable characteristics.
Multi-Plot	Multi-plot provides graphical analysis and annotation of results.
Run-Time	The run-time administers and controls the simulation.
T-Line	The T-Line module specifies user defined transmission line characteristics.

5.2.1.5 Software: PCDAA

The developed Phasor Data Concentration and Differential Application (PCDAA) facilitates real-time synchrophasor data concentration from the line terminals in the two-ended transmission line system. Once the synchrophasor data is concentrated the differential current is evaluated using the modelled current differential algorithm. The PCDAA also employs a historian that facilitates post disturbance and trend analysis.

5.2.2 Test-bench design

To validate the proposed concept of a synchrophasor-based transmission line differential protection scheme, a laboratory scale Test-bench is designed to enable practical implementation the PMU-based protection scheme.

A differential relay (87T)-based protection scheme is modelled and incorporated into the Test-bench. The design criteria behind incorporating the (87T)-based protection scheme into the Test-bench is to provide standard characteristic response metrics against which the proposed PMU-based protection schemes' response can be correlated. Incorporating a conventional differential relay-based protection scheme enables parallel implementation of the protection schemes. This design feature forms the basis for the evaluation of the PMU-based differential protection scheme. Figure 5.2 presents a simplified block diagram of the Test-Bench.

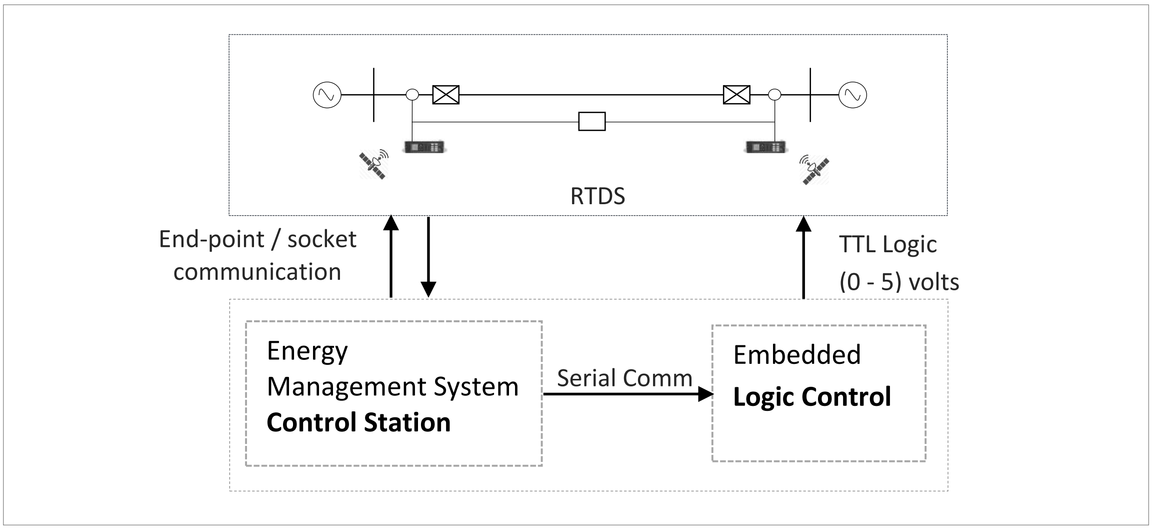


Figure 5.2: Test-bench block diagram

Table 5.3 provides point to point network identification for the components in the PMU-based line transmission test-bench model.

Table 5.3: Communication settings: IP addresses

Device Description	Location	Network-Address	Network Location
PCDAA	Energy Management System, Control Station	192.168.1.4	Control Station
PMU_1 :ID-30	Local Substation	192.168.1.202 **port:** 4830	RTDS
PMU_2 :ID-31	Remote Substation	192.168.1.204 **port:** 4831	RTDS

Figure 5.3 presents a more detailed illustration of the laboratory scale Test-bench.

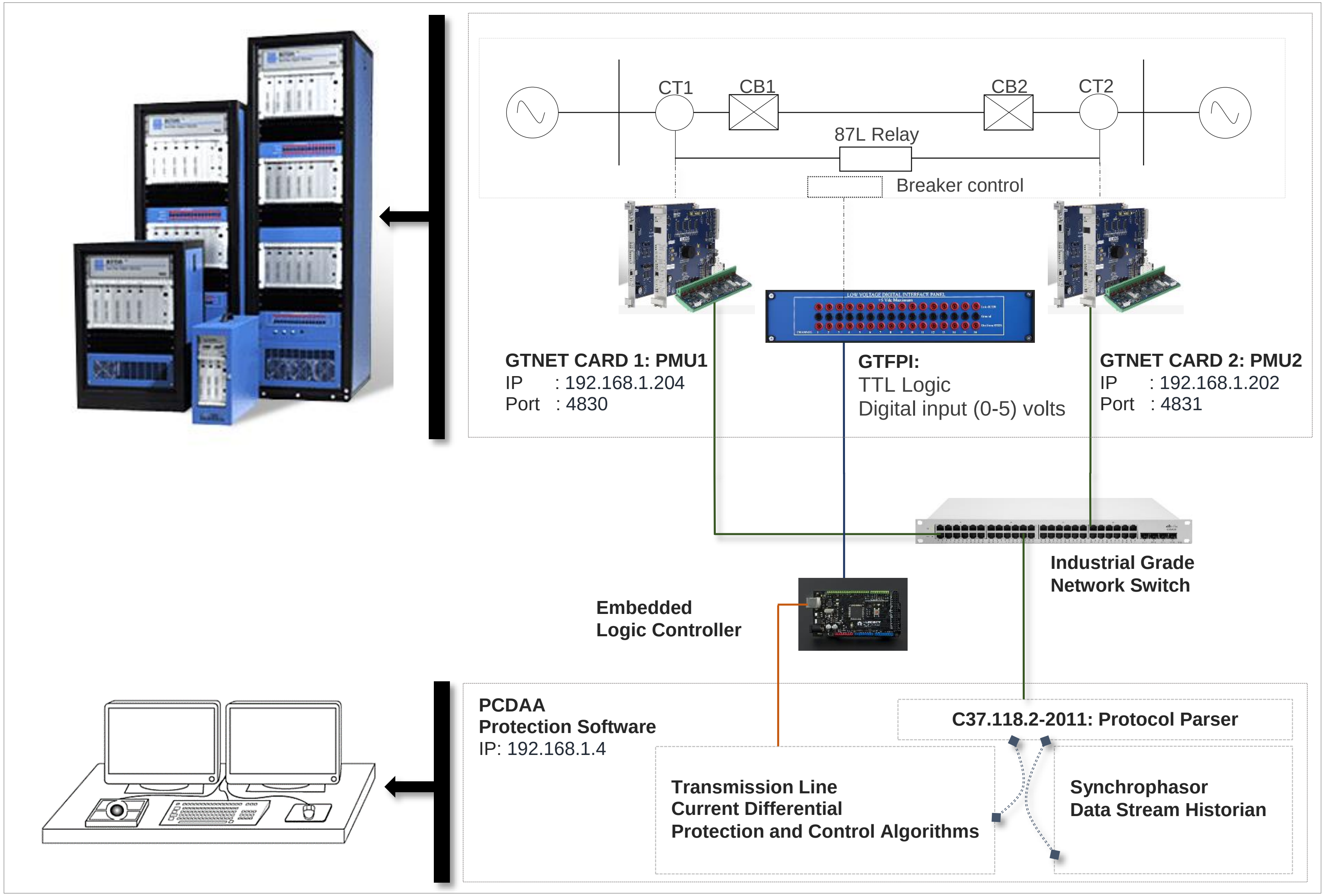


Figure 5.3: CSAEMS laboratory scale Test-bench

5.2.3 RSCAD Transmission line system draft model

Figure 5.4 presents the RSCAD transmission line draft utilized for the real-time simulation. Configuration of the RSCAD component models is described in this Section.

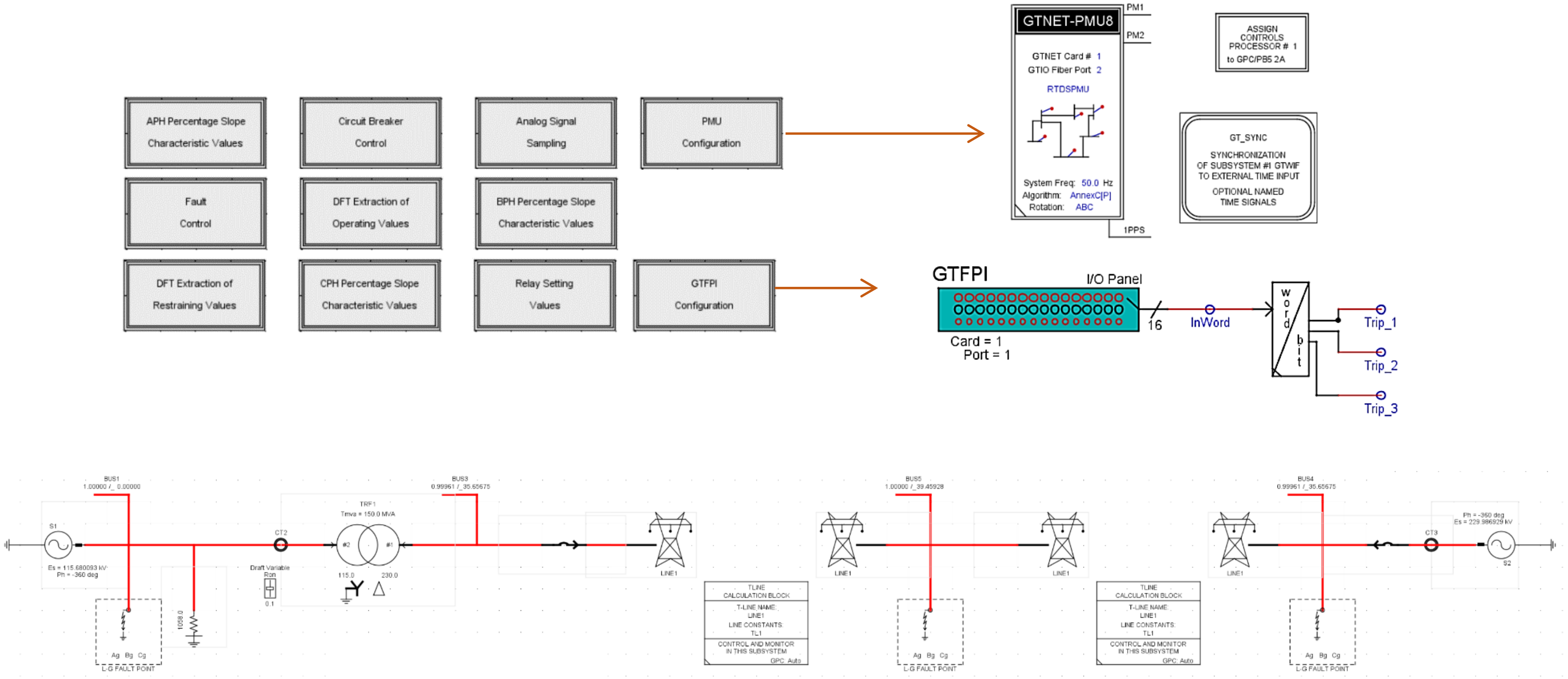


Figure 5.4: RSCAD - Draft (Transmission line differential protection scheme system)

To validate the hypothesis, it is necessary to establish operational metrics for the PMU-based transmission line differential protection scheme and correlate them against standard, marginal operation requirements. In the simulation study, a conventional differential protection scheme is modelled based on digital differential relays. Performance of the PMU-based transmission line differential protection scheme is to be comparatively correlated with standard performance of a conventional differential protection scheme based on differential relays (87L). The simulation approach provides a practical method to validate the hypothesis, where the PMU-based protection scheme and 87L-based differential protection schemes are implemented over the same transmission line model to facilitate comparative analysis between the conventional protection scheme and the PMU-based protection scheme over a transmission line system. The real-time RSCAD transmission line system is modelled utilizing draft components from the RSCAD library which include;

- a current transformer
- an AC source at each end of the line
- a capacitive coupled voltage transformer
- a circuit breaker at each end of the transmission line
- a single transmission line consisting of a faulted line model
- fault switches to create internal faults, forward looking faults, or reverse looking faults

The components of the transmission line model enable engineering of the system which incorporate;

- fault point
- fault inception,
- fault control selection
- circuit breaker logic control
- P-Class PMU configuration

A detailed description of the transmission line system components and their configuration is presented from sub-section 5.2.3.1 through sub-section 5.2.3.8.

5.2.3.1 Component configuration: AC Source model

The transmission line system model consists of two sources as illustrated in Figure 5.5.

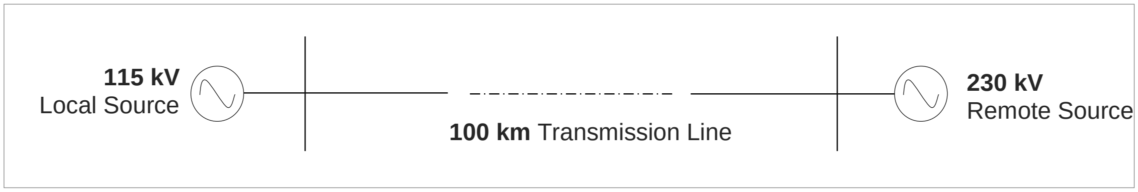


Figure 5.5: Transmission line AC source model

The local source is 115 kV and the remote source is 230 kV. Tables 5.4 and 5.5 provide specified parameters for the local and remote AC sources respectively.

Table 5.4: Local AC Source parameter specifications

Parameter	Specification
Voltage Time Input Constant	0.05
Load Flow Result Real Power	-84.903480 (MW)
Load Flow Result Reactive Power	5.259443 (MVAR)
Specified Initial Real Power	100 (MW)
Specified Initial Reactive Power	50 (MVAR)
Zero Sequence Impedance	51.2 (ohms)
Zero Sequence Impedance Phase	73.59 (degrees)
Initial Source Mag (L-L, RMS)	115 (kV)
Initial Frequency	50 (Hz)
Initial Phase	-360(degrees)

Table 5.5: Remote AC Source parameter specifications

Parameter	Specification
Voltage Time Input Constant	0.05
Load Flow Result Real Power	100.903480 (MW)
Load Flow Result Reactive Power	-13.674237 (MVAR)
Specified Initial Real Power	100 (MW)
Specified Initial Reactive Power	50 (MVAR)
Zero Sequence Impedance	1.28
Zero Sequence Impedance Phase	73.59 (degrees)
Initial Source Mag (L-L, RMS)	230 (kV)
Initial Frequency	50 (Hz)
Initial Phase	-360(degrees)

5.2.3.2 Component configuration: Transmission Line model

Transmission lines are modelled using travelling wave algorithms within the RTDS. Utilizing the RSCAD T-Line module, the line's modal characteristic impedances, travel times and transformation matrix data are determined and generated to model a multi conductor transmission line. Tables 5.6 presents' general line information and configuration data for the transmission system T-line.

Table 5.6: General line information and Data

Parameter	Specification
Units	Metric
Frequency	50 Hz
Line Name	f230x100
Line Model	Bergeon (Physical Data entry)
Line Length	100 (km)
Ground Resistivity	100 (ohm - m)

Tables 5.7 presents parameter specifications for the tower and right of way data.

Table 5.7: Tower and right of way Data

Parameter	Specification
Tower Type	Manual
Number of Towers	2
Location in Right of Way	0.0 (m)
Number of Circuits on Tower	1
Number of Conductors on Tower	3
Number of Ground Wires on Tower	2

Tables 5.8 presents ground wire configuration settings.

Table 5.8: Ground wire

Parameter	Specification
Height at Tower	35 (m)
Sag at Mid-span	10 (m)
Horizontal distance	5 (m)
Ground Wire Radius	0.55245 (cm)
DC Resistance per Wire	2.8645 (ohm / km)

Tables 5.9 presents ground wire configuration settings.

Table 5.9: Conductor

Parameter	Specification
Conductor Name	Chukar
Shunt Conductance	1.0e-11(mho/m)
Sub-Conductor Radius	0.03454 (cm)
DC Resistance per Sub Conductor	0.032062 (ohms/km)
Number of Sub Conductors per Bundle	2
Sag at Mid-span	10 (m)
Bundle Configuration	Symmetrical
Sub Conductor Spacing	45.72 (cm)

5.2.3.3 Component configuration: Fault Control Branch

A fault branch is configured and connected to the system bus to simulate line to ground and line to line faults. Fault inception and fault removal is established through utilizing a pulse generator control function with a specified pulse width that determines the duration of the fault as illustrates in Figure 5.6.

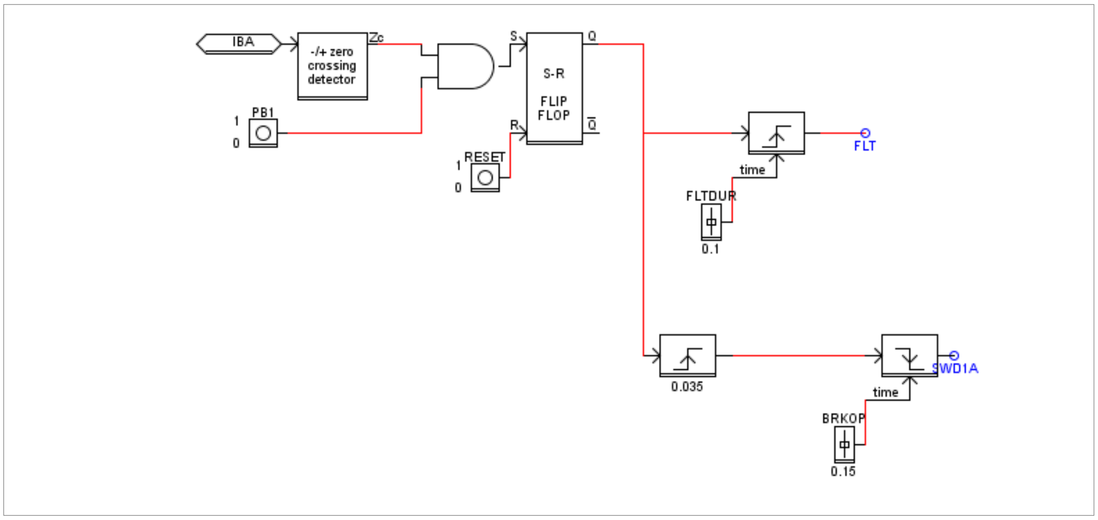


Figure 5.6: Fault Control branch

5.2.3.4 Component configuration: Circuit breaker control

The circuit breaker control logic is illustrated by the flow chart in Figure 5.7 below.

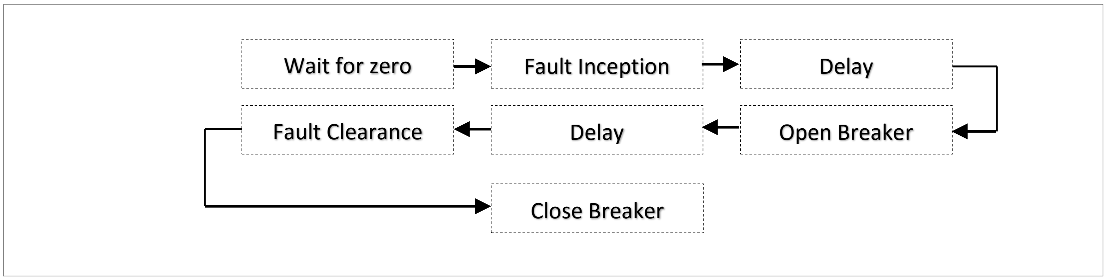


Figure 5.7: Breaker control logic flow

Circuit breaker operation logic opens and closes the breaker. The circuit breakers are administered by the status of the lockout relay (86T). If the 86T is operated the circuit breakers will open if previously closed and cannot be closed until the 86T is reset. The differential relay trip outputs for each phase set the 86T, and provide a logic signal called 87T. Figure 5.8 illustrates logical implementation of breaker control and lockout logic.

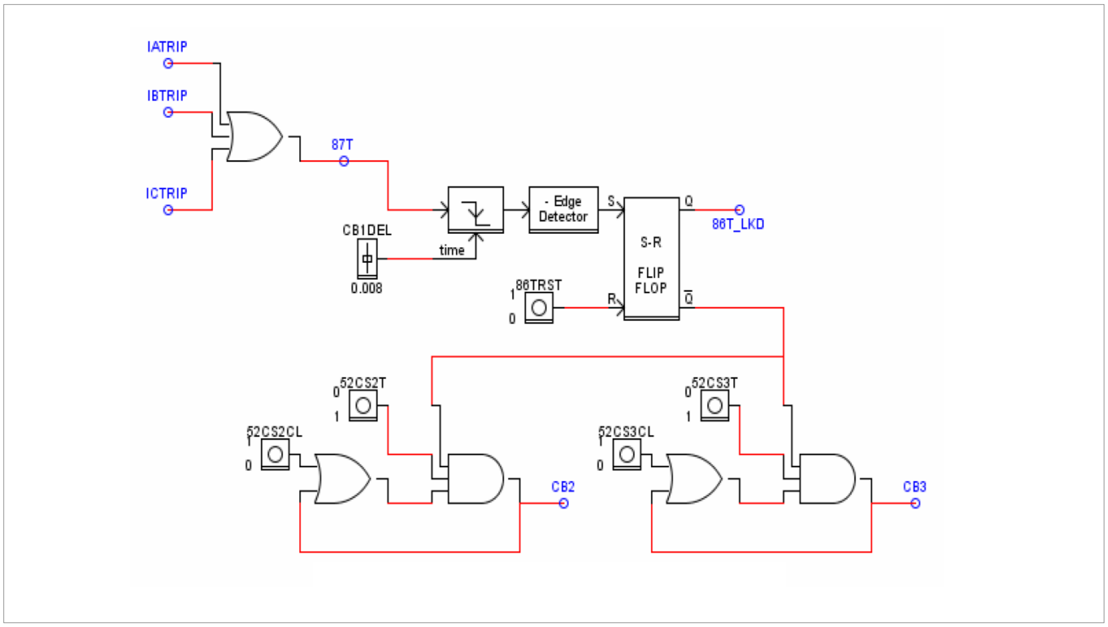


Figure 5.8: Breaker control and lockout logic

5.2.3.5 Component configuration: Instrument Transformer

Two current transformers are used to provide secondary signals. Table 5.10 specifies parameter configuration CT2.

Table 5.10: Current Transformer (CT2) configuration parameters

Parameter	Specification
Turns Ratio	400
Secondary Side Resistance	0.5 (ohms)
Secondary Side Inductance	$0.8e^{-3}$ (H)
Burden Series Resistance	0.05 (ohms)
Burden Series Inductance	$1.0e^{-3}$ (H)
Frequency	50 (Hz)
Path Length	0.5 (m)
Core Characteristics Data entry	BH

Table 5.11 specifies parameter configuration CT3.

Table 5.11: Current Transformer (CT3) configuration parameters

Parameter	Specification
Turns Ratio	200
Secondary Side Resistance	0.5 (ohms)
Secondary Side Inductance	$0.8e^{-3}$ (H)
Burden Series Resistance	0.05 (ohms)
Burden Series Inductance	$1.0e^{-3}$ (H)
Frequency	50 (Hz)
Path Length	0.5 (m)
Core Characteristics Data entry	BH

5.2.3.6 Component configuration: Differential relay model

Modelling of the 87T relay in RSCAD is an essential part of the Test-bench design as the 87T-based protection scheme provides a parallel laboratory scale simulation standard against which the PMU-based protection scheme is correlated and practically validated. The logical implementation of the 87T relay provided by (ref) is provided here.

Two winding percentage differential protective relays are modelled with dual slope characteristic and 2nd harmonic blocking to mitigate effects of transformer inrush during energization. Characteristics of the relays models include;

- Slope 1 characteristic
- Slope 2 characteristic
- 2nd harmonic blocking
- Minimum operating current
- Maximum operating current

Figure 5.8 illustrates a graphic representation of the dual slope percentage characteristic of the relay. It should be noted that the same dual slope percentage characteristic is employed by the PCDAA to establish restraint effect and security against external faults. The characteristics are used to define the trip and restraint margins, outlining operating current for trip conditions. The restraint current is proportional to the operating current, increasing with cumulative operating current to ensure evaluation of accurate trip conditions.

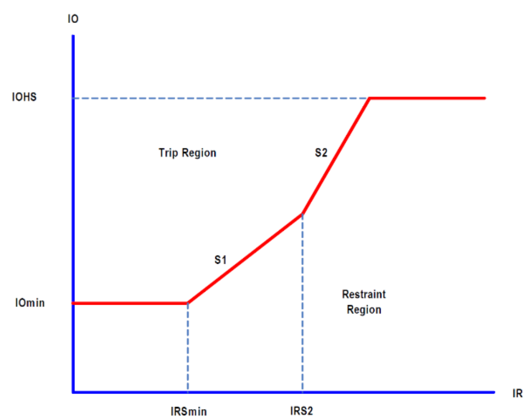


Figure 5.9: Dual Slope Percentage Characteristic

In Figure 5.9 S1 denotes the relay setting for the operating current. Should restraint current exceed the maximum restraint current for slope one, a secondary slope (higher sensitivity) is adopted evaluate fault condition operating current. S2 denotes the relay setting for the operating current on the steeper slope. Should restraint current exceed IOHS / S2 the restraint effect is disregarded and the relay will operate with the high set current trip condition. The winding percentage, differential protective relay model components are comprehensively detailed here;

a. Analog signal sampling:

The digital relays are configured to sample the analog data at a rate equivalent to the protection cycle time. For the simulation, a sample rate of 96 samples per cycle for a 50 Hz base is configured. Taking into account the protection cycle which is set to 8 times per cycle, the data is down-sampled to 8 times per cycle. Figure 5.10 presents the relay sample logic.

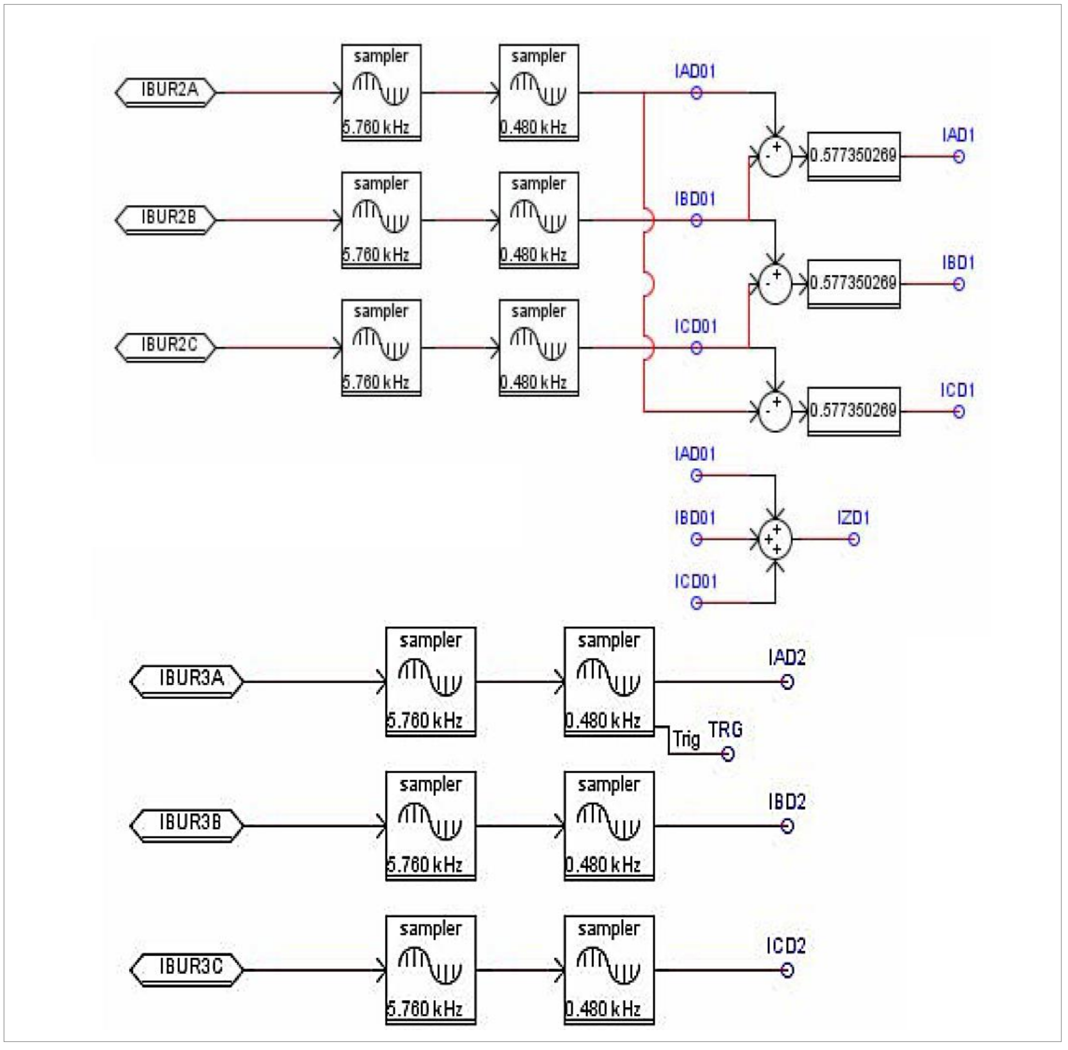


Figure 5.10: Relay Sampling Logic

b. Extraction of operating current:

Sample data from each of CT is subsequently utilized to evaluate the operating current. A Fourier Full Cycle Algorithm is then used to extract the fundamental and 2nd harmonic magnitude from the operating currents.

Figures 5.11, 5.12 and 5.13 depict representation of the operate current as a vector sum of the two restraint currents. Under normal operation, magnitude ***IAD1* = *IAD2***, with phases 180 degrees apart. Where for *A* phase;

IAD1* = *IHV

IAD2* = *ILV

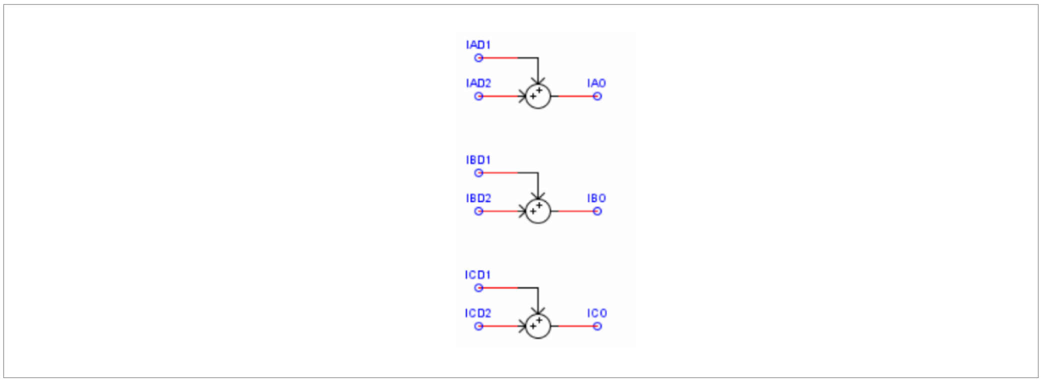


Figure 5.11: Addition of sampled quantities

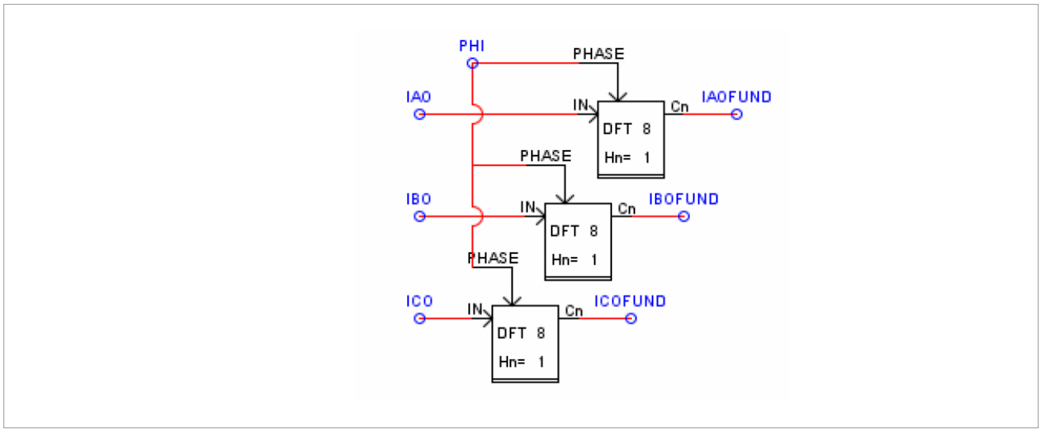


Figure 5.12: Extraction of fundamental quantities

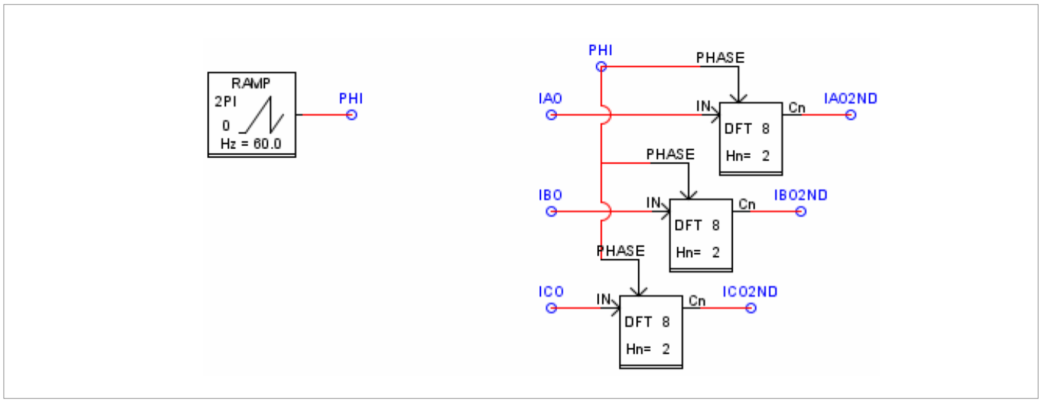


Figure 5.13: Extraction of 2nd harmonic quantities

c. Extraction of restraining current:

Sample data from each CT is used to evaluate the operating current. The imaginary and real parts for each signal are extracted using Fourier Full Cycle Algorithm. The magnitude of each signal is extracted using complex operators. Once extracted, the magnitudes are summed. The sum of the magnitudes is then divided by the number of windings to evaluate the restraint current. Implementation of the relay restraint current is presented in Figures 5.14.

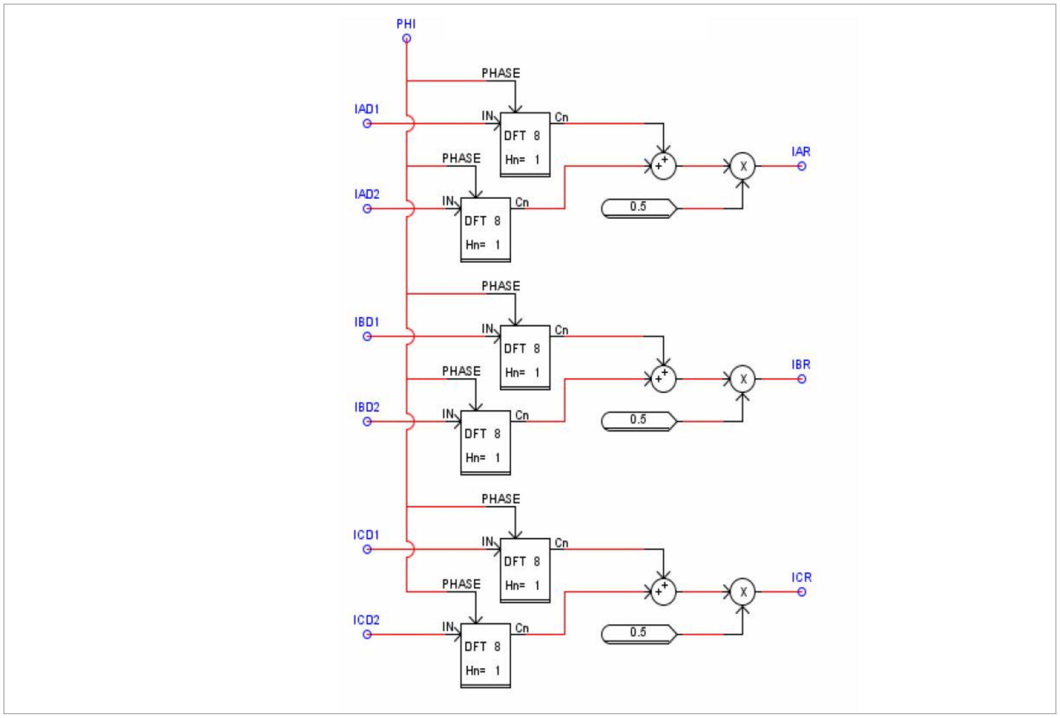


Figure 5.14: Relay Restraint Current

d. Relay setting values:

As illustrated in Figure 5.15, sliders are used to set or vary the relay characteristic in the RSCAD Runtime. The sliders will vary the settings listed below;

- low sensitivity characteristic (slope one)
- high sensitivity characteristic (slope two)
- 2nd harmonic blocking
- operating current (minimum and maximum)

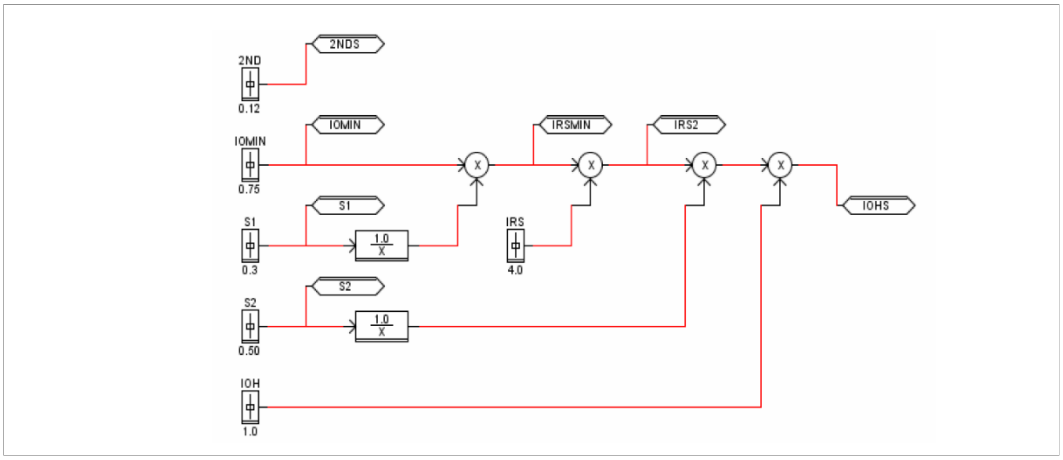


Figure 5.15: Relay Setting control logic

e. Dual slope percentage characteristic:

The dual slope percentage differential characteristic is implemented using the operating and restraint current magnitudes together with the relay setting values. With reference to Figure 5.16;

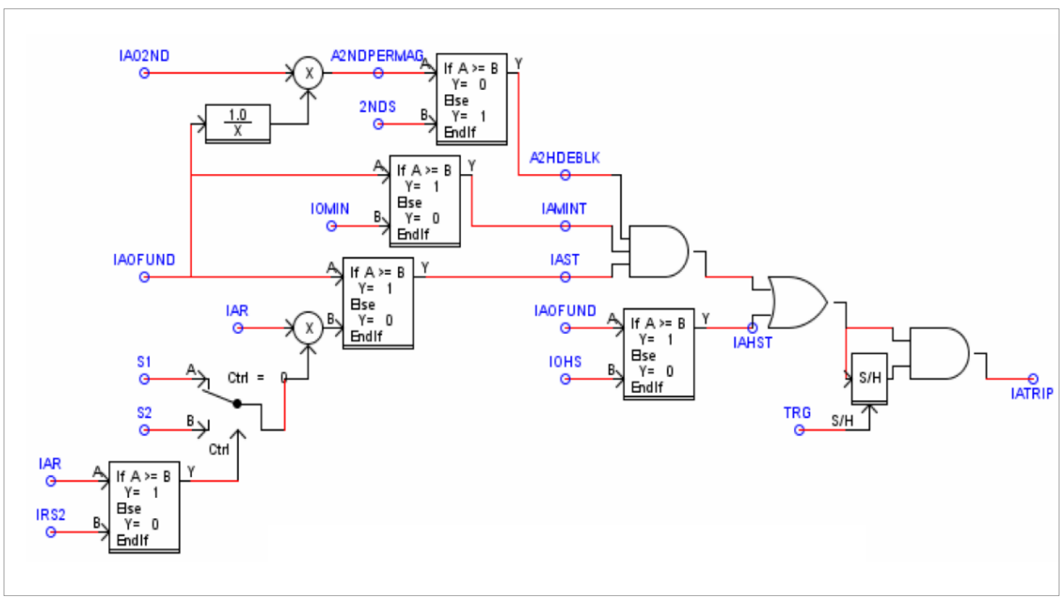


Figure 5.16: Relay setting logic

Part I:

The configured percentage restraint characteristic contrasts the 2nd harmonic against the fundamental harmonic to evaluate the 2nd harmonic content. The specified 2nd harmonic block setting is used to determine the magnetizing inrush conditions.

Part II:

The dual slope percentage differential logic compares the value of fundamental current to the minimum operating current setting. The measured operating current must be above the relay setting or the relay will not operate.

Part III:

The dual slope percentage differential logic compares the value of fundamental current to the value of restraint current characterized by the necessary relay slope setting. The measured value of restraint current is compared against the IRS2 value which is the breakpoint for the dual slope characteristic. IRS2 is evaluated automatically from the relay setting logic, when the restraint current is above this breakpoint, the second slope S2 is used.

Part IV:

The dual slope percentage differential logic uses the output of the first three parts to create a relay trip condition when the operating and restraining currents are within the trip zone of the dual slope percentage characteristic.

Part V:

The dual slope percentage differential logic compares the magnitude of operating current to the high set current setting.

Part VI:

The dual slope percentage differential logic contains a sample and hold component with an AND gate, this logic ensures relay operation limited to the trip zone during the sample period before a trip is declared.

5.2.3.7 Component configuration: GTNET PMUv4

As opposed to the 87T relay model which is completely software based in RSCAD, the PMUs utilized to implement the laboratory scale synchrophasor-based protection scheme consist of PMU hardware components facilitated by the RTDS GTNET component. The GTNET card is connected to the RTDS through a GTIO port on the RTDS processor (GPC/PB5) cards. PMU components on the GTNET provide symmetrical three phase data of instantaneous voltage and current measurements. The PMUs are set up and configured in the GTNET component. The C37.118 data output is enabled and administered by the

GTSYNC. The GTNET−PMU component output is synchronized to an external One Pulse per Second (1PPS), IRIG−B, or IEEE 1588 signal via the GTSYNC. Table 5.12 presents the GTNET PMU configuration parameters.

Table 5.12: GTNET PMU configuration parameters

Parameter	Specification
Enable Output of C37.118 data using GTNET	Yes
GTNET Component Name	Virtual PMU
Configuration frame format	Config2
Base Frequency	50 (Hz)
Number of PMU’s	2
Enable Primary Signals	yes
GTIO Fibre Port Number	1
Assigned Control Processor	1
Priority Level	1
GTNET_PMU Card Number	1

The PMU parameter specifications for the local substation PMU_1 and the remote substation PMU_2 are presented in Tables 5.13 and 5.14 respectively. Both phasors are configured to the angular output data format having each of the PMUs configured to enable monitoring of the below listed outputs.

- phasor runtime output
- rate of change of frequency (ROCOF) runtime output
- frames per second (FRACSEC) runtime output

Table 5.13: GTNET Local PMU configuration parameters

Parameter	Specification
Report Rate (frames / sec)	200
Decimate PMU runtime output	yes
Station Name	remotePmu
Hardware ID	31
Output TCP/IP or UDP local port	4831
Number of phasors	6
Phasor Number Format	Real
Frequency Number format	Real
Number of Analog Values	0
Analog Number Format	Real
Number of 16 bit Digital Status	0
Phasors 1 – 6 PMU output (respectively)	IA, IB, IC

Table 5.14: GTNET Remote PMU configuration parameters

Parameter	Specification
Report Rate (frames / sec)	200
Decimate PMU runtime output	yes
Station Name	localPmu
Hardware ID	30
Output TCP/IP or UDP local port	4830
Number of phasors	3
Phasor Number Format	Real
Frequency Number format	Real
Number of Analog Values	0
Analog Number Format	Real
Number of 16 bit Digital Status	0
Phasors 1 – 6 PMU output (respectively)	IA, IB, IC

5.2.3.8 Component configuration: Front Panel Interface (GTFPI)

The digital Input or Output (I/O) panel located in the front of the RTDS is used to send the Differential Transfer Trip (DTT) logic signal to the real-time transmission, line differential protection scheme simulation. The digital I/O panel is connected to the GTFPI card which is connected to the GTAO port of the GPC/PB5 processor card via a fiber optic cable. This enables configuration of optically isolated digital I/O connections and real-time interface with the simulated transmission line differential protection scheme. Specification of the card numbers that facilitate the hardware-in-loop real-time simulation study are obtained from physical inspection of the cards in the RTDS cubicle and presented in Table 5.12.

Table 5.15: GTFPI hardware-in-loop card number specification

Card specification	GTFPI	GTAO	GPC	GT Fibre Port
Card number	**1**	**1**	**2**	**1**

The digital inputs utilize Transistor-Transistor Logic(TTL) logic with input ports which pulled up to positive five volts (+5 v) via a one kilo ohm (1 kΩ) resistor to provide;

- bit logic = 1; for 0 volt input
- bit logic = 0; for +5 volt input

The logic can be optionally inverted and used to open or close circuit breakers as required within the real-time simulation. Table 5.13 and 5.14 present the parameter specifications for the word-bit converter and GTFPI respectively, which are used to configure the digital input ports of the RTDS.

Table 5.16: Word-bit converter, parameter specifications

Parameter	**Specification**
Priority level	2
Solve model on card type	GPC/PB5
Assigned Controls processor	1
Number of outputs	1

Table 5.17: GTFPI, parameter specifications

Parameter	Specification
GTIO Fibre port number	1
GTFPI Card number	1
Enable Digital panel Input signals	YES
Invert Digital panel Input signals	NO
Priority level	1
Assigned Controls processor	1

Figure 5.16 illustrates the control component model used to configure the digital GTFPI of the RTDS. The set-up enables external differential trip logic to be introduced into the real-time simulation. The three external control trip logic signals Trip_1, Trip_2 and Trip_3 are input to the breaker control and lockout logic circuit respectively matching them to the IATRIP, IBTRIP and ICTRIP input signals which represent IA trip, IB trip and IC trip signals.

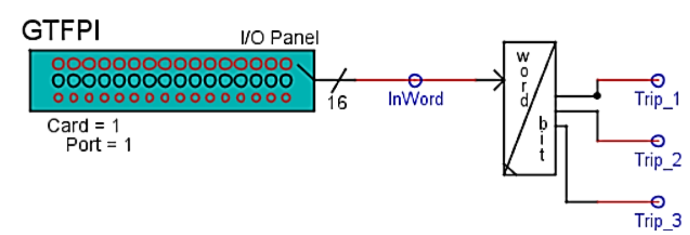


Figure 5.17: GTFPI interface

5.2.4 PCDAA breaker logic control

In Chapter Three, a concept for remotely transferring direct trip logic is proposed. The CSAEMS laboratory scale Test-bench uses an embedded system to emulate the circuit breaker logic control. Figure 5.19 illustrates how the ATmega 2650 board is used to facilitate trip logic to the real-time simulation. The ATmega 2560 is connected to the Test-bench PC via a Universal Serial Bus (USB) port, it is also connected to the digital TTL GTFPI ports of the RTDS.

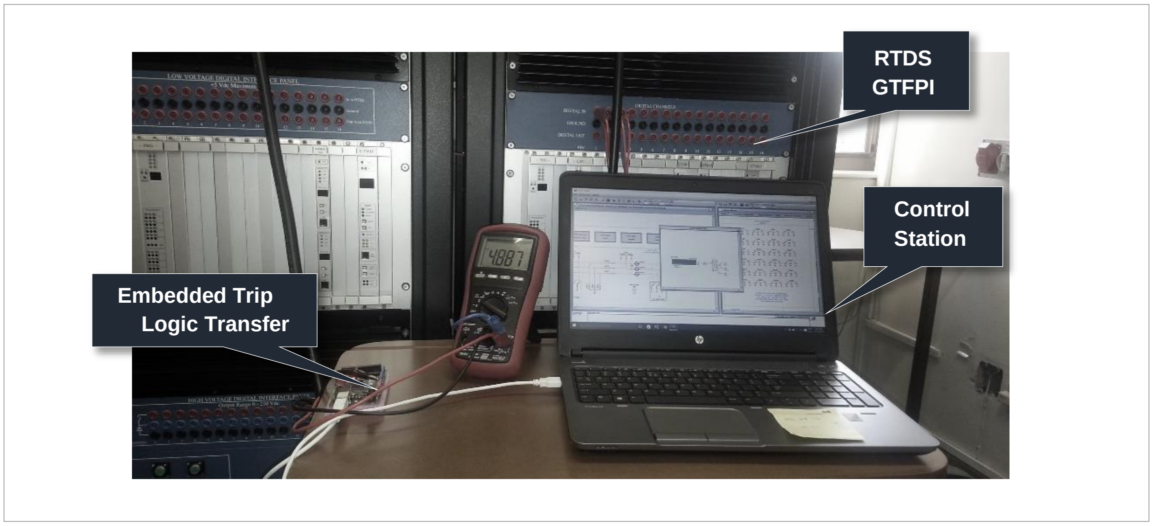


Figure 5.18: CSAEMS Laboratory scale Test-bench

The Microsoft Visual Studio (MSVS) development environment exposes serial ports to the real-time simulation communication network. MSVS enables configuring and utilizing the Test-bench PC as a serial server, transmitting data through the serial communication port. MSVS utilizes the IP address of the PC, configuring the PC to feed data to the specified port where ATmega 2560 is connected in real-time which subsequently relays the circuit breaker logic to the real-time simulation running on the RTDS. Figure 5.19 presents the microcontroller circuit diagram.

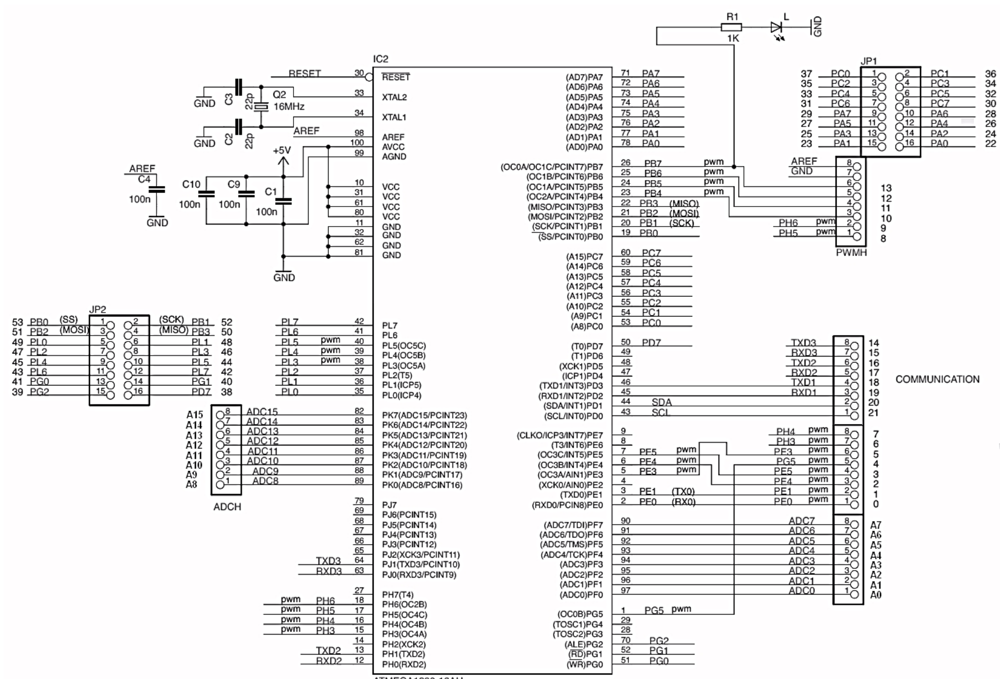


Figure 5.19: ATmega 2560 microcontroller circuit diagram

The microcontroller specifications are given in Table. 5.18.

Table 5.18: ATmega 2560 microcontroller specifications

Parameter	Specification
Microcontroller	ATmega 2560
Input / Output (I/O)	54 - Pins
Pulse With Modulation (PWM)	15 - Outputs
Analog Inputs	16
Crystal Oscillator	16 MHz
Universal Asynchronous Receiver / Transmitter (UART)	4 Hardware Serial Ports

5.3 Discussion

The Test-bench is an essential research contribution which enables practical evaluation of utilizing synchrophasors directly in transmission line differential protection. The Test-bench design is configured to provide laboratory scale implementation of the PMU-based protection scheme. The Test-bench design also emphasizes parallel implementation of the PMU-based protection scheme with the 87T-based protection scheme, for correlating results of the PMU-based protection scheme against standard operation of the conventional protection scheme.

5.4 Conclusion

This Chapter provides parameter specification for configuring the real-time transmission line component models utilized to draft the transmission system for real-time simulation. Chapter Six presents the real-time simulation study and experimental results.

Chapter 6 :
Simulation studies and results

6.1 Introduction

The simulation study to substantiate the hypothesis is carried out in the Centre for Substation Automation and Energy Management Systems (CSAEMS) laboratory. The laboratory scale Test-bench is set up and configured to fully utilize the Real-Time Digital Systems Simulator (RTDS) with the GPC & PB5 processor cards. The RTDS is interfaced by RSCAD, an engineering software that provides accurately controlled real-time simulation of the transmission line system, where the Phasor Measurement Unit (PMU)-based protection scheme is implemented. The real-time simulation runtime environment enables acquisition of reproducible results, which are used to evaluate operation of the PMU based transmission line differential protection scheme and analyze its performance.

Before carrying out the simulation study, the Test-bench communication network is tested to verify real-time data transmission of synchrophasors between line terminal PMUs and the Phasor Current Differential Action Adapter (PCDAA). Once verified the real-time simulation study is carried out. The aim of the simulation study is to correlate operational characteristics and performance of the PMU-based transmission line protection scheme against those of a conventional differential relay (87L)-based transmission line protection scheme. All results of this pragmatic study are presented in this Chapter.

6.1.1 Chapter organization

Section 6.2 describes the Test-bench communication network and its testing.

Section 6.3 introduces the RSCAD and the developed PCDAA interfaces used to implement the real-time simulation study.

Section 6.4 details the local terminal external fault case study.

Section 6.5 details the remote terminal external fault case study.

Section 6.6 details the internal fault case study.

Section 6.7 concludes the Chapter.

This Chapter emphasizes realization of a PMU-based transmission line differential protection scheme through implementing and demonstrating the feasibility of the proposed protection scheme. The aim of the experimental investigation is to “prove the concept” of using synchrophasors for real-time protection. As mentioned in Chapter Five, a differential relay (87T)-based protection scheme is modelled and incorporated into the Test-bench. The design criteria behind incorporating the (87T)-based protection scheme into the real-time simulation is to provide standard characteristic response metrics against which the proposed PMU-based protection schemes’ response can be evaluated and analyzed. Parallel implementation of the protection schemes forms the basis for evaluating of the PMU-based differential protection scheme, comparing operational results of the protection schemes and inferring conclusive deductions. This experimental method enables practical verification of robustness, extent of usability selectivity, sensitivity and degree of reliability.

The transmission line system detailed in Chapter Five is simulated in real-time, implementing the PMU-based differential protection scheme and the 87T-based differential protection scheme as illustrated in Figure 6.1 overleaf.

The 87T-based protection scheme is implemented using the RSCAD interface, employing software component models that are simulated by the RTDS in real-time.

The PMU-based protection scheme is implemented using the developed PCDAA protection utility software. This scheme utilizes the RTDS GTNET PMU component hardware. The PMU-based protection scheme employs the developed PCDAA to transfer the differential trip logic. In the laboratory scale implementation, transfer of the differential trip logic is established using an embedded system which interfaces the PCDAA and the RTDS throughout the real-time simulation of the transmission line system.

Percentage differential protection is administered by both protections schemes, implementing identical parameter specifications. This control deliberation enables equitable correlation of response from both protection schemes over the same control environment.

The simulation study follows the procedural abstraction illustrated in Figure 6.1.

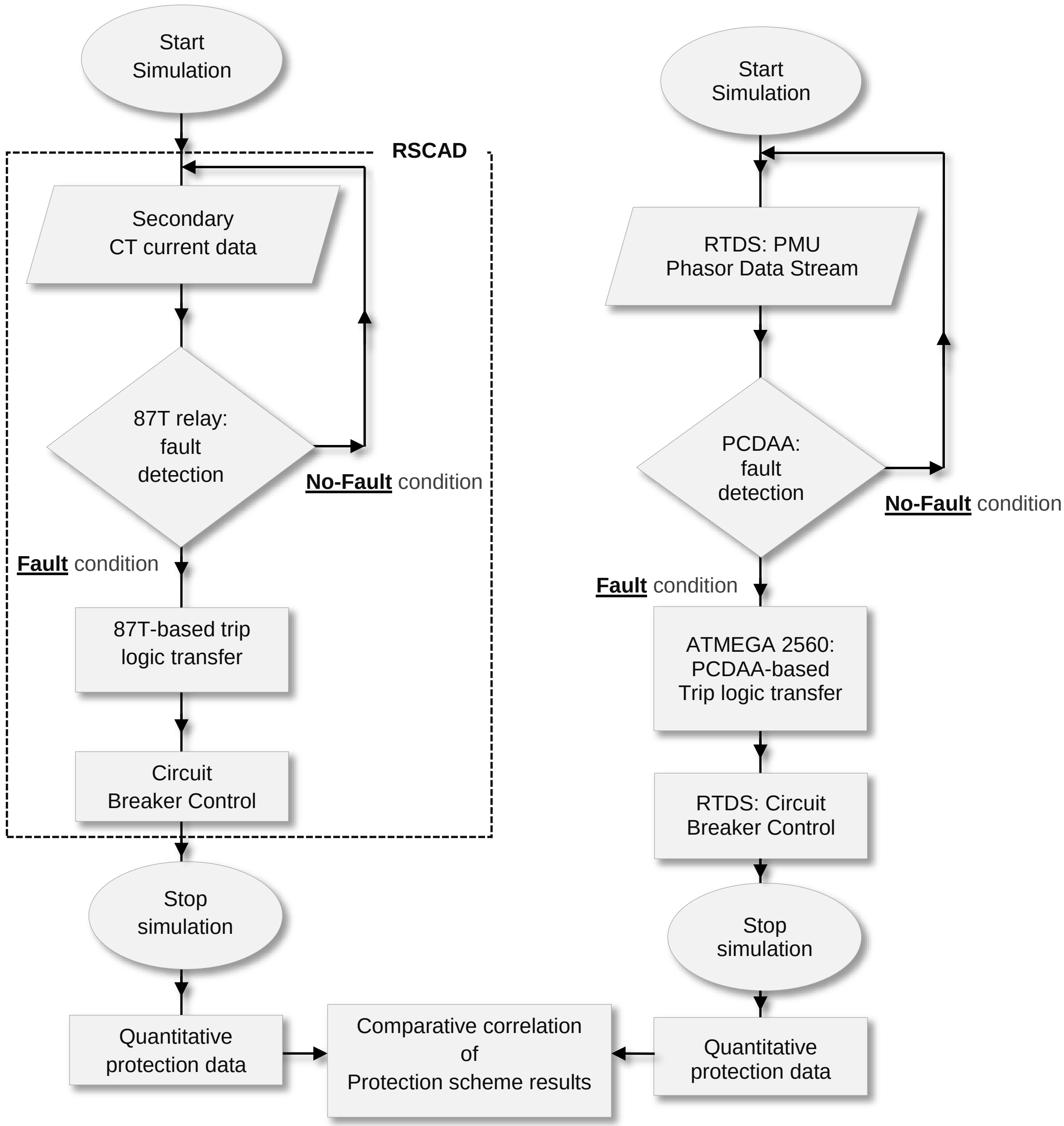


Figure 6.1: Feasibility investigation of the proposed PMU-based protection scheme

The three test cases emphasized in this Chapter are illustrated in Figure 6.2.

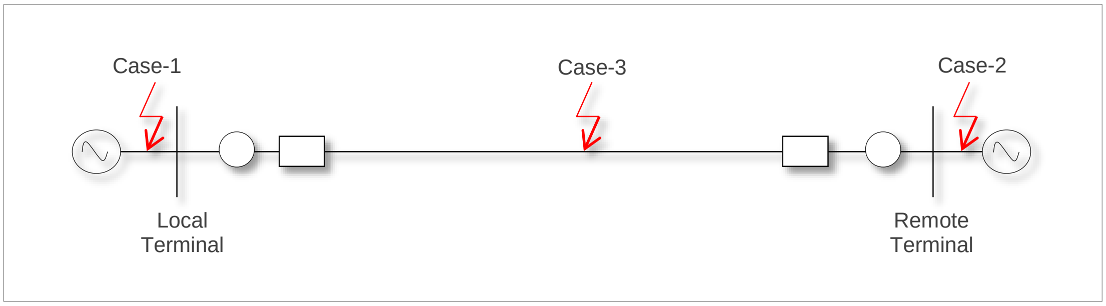


Figure 6.2: Simulation study, test cases

The three case studies include;

a. Case study 1: Local Terminal External Fault (LTEF) case
b. Case study 2: Remote Terminal External Fault (RTEF) case
c. Case study 3: Internal Fault (IF) case

For each case study, the characteristics that are considered and evaluated consist of;

i. Trip- transfer times
   - Analysis of the PMU-based protection scheme, differential trip transfer signal (87T).
   - Analysis of the PMU-based protection scheme, differential trip transfer signal (PMUT).

ii. Impulse transients
   - Analysis of the current transformer response during implementation of the PMU-based protection scheme.
   - Analysis of the current transformer response during implementation of the 87T-based protection scheme.

iii. Protection scheme characteristic response
   - Evaluation of the circuit breaker operational characteristics during implementation of the PMU-based protection scheme.
   - Evaluation of the circuit breaker operational characteristics during implementation of the 87T-based protection scheme.

6.2 Communication network configuration and verification

6.2.1 Communication network configuration

The Test-bench network host employs the internet protocol suite, which administers the simulation communication network. The real-time transmission line Logical Nodes (LN) are illustrated in Figure 6.3. The LNs interact over the communication network utilizing static network addresses assigned within the network layer. Figure 6.4 illustrates manual assignment of the static addresses within the Windows Operating System (WOS).

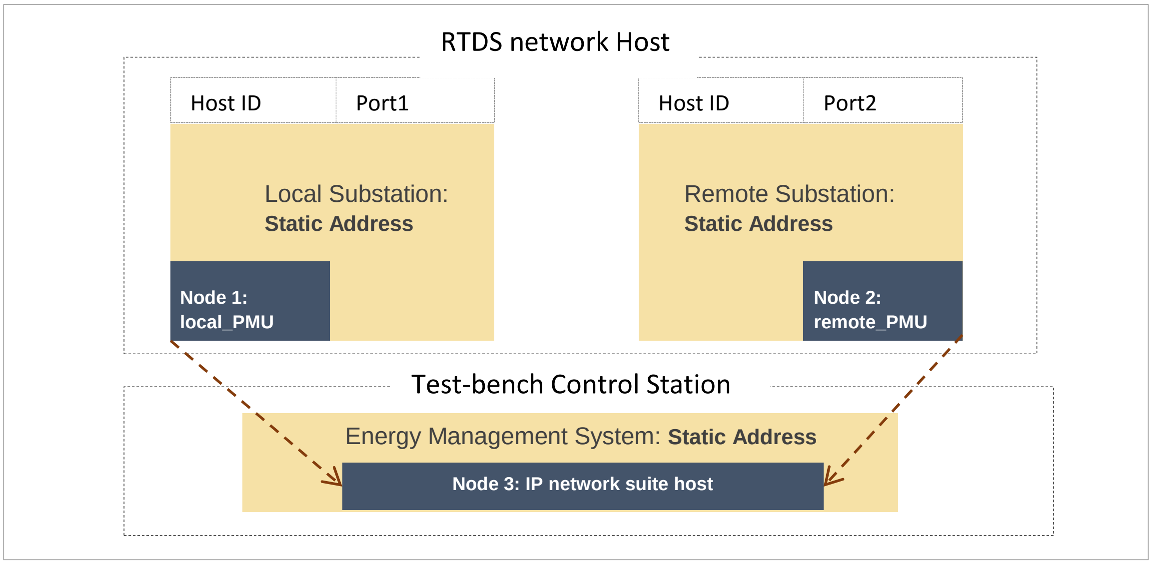


Figure 6.3: Test bench transmission line system nodes and node interaction

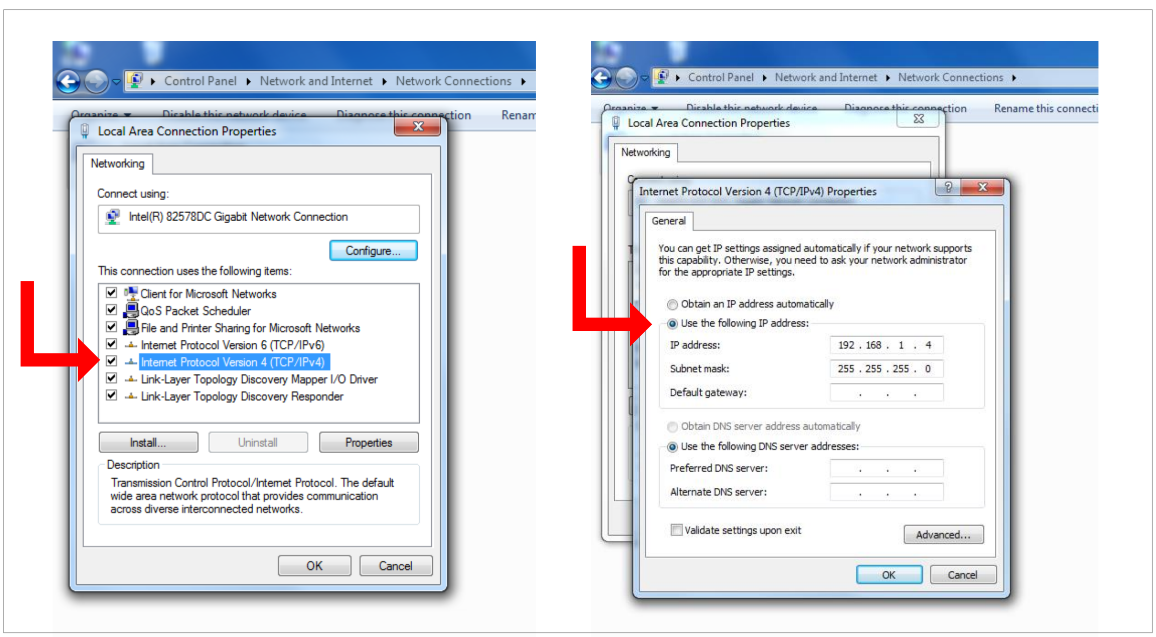


Figure 6.4: Assignment of static network addresses

6.2.2 Communication network verification

This Section describes the test carried out to verify the Test-bench communication network. A preliminary, independent RSCAD draft case is developed to simulate real-time synchrophasor data transmission over the configured communication network to validate accurate data transmission from the PMU. This Section also introduces the CodePlex PMU Connection Tester, which is used to test the streaming PMU measurement data over the Test-bench communication network (CodePlex, 2016). Figure 6.5 presents the RSCAD draft used to test real-time transmission of the phasor data over the Test-bench communication network.

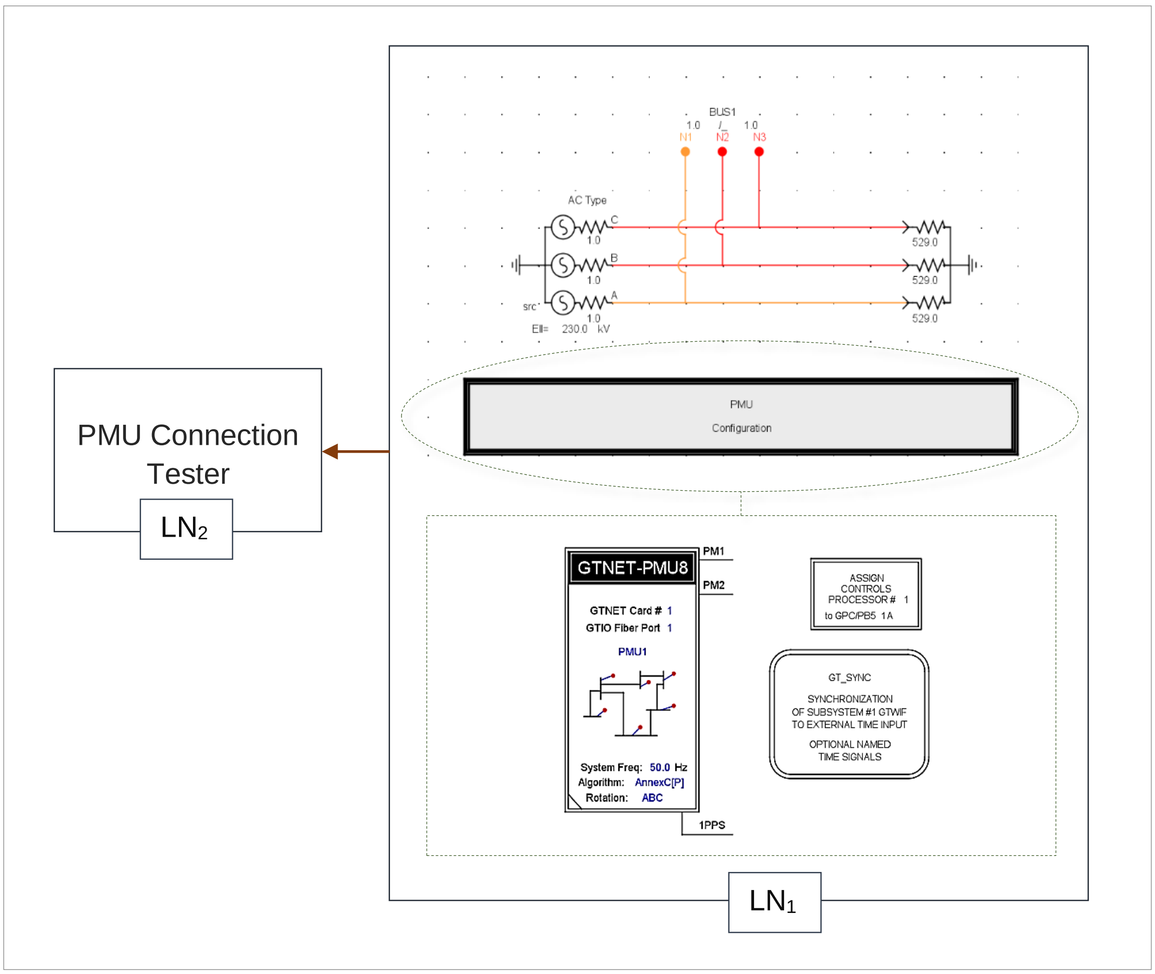


Figure 6.5: PMU real-time data transmission testing

A simple two LN network is simulated with a single PMU at LN$_1$ and the PMU connection Tester at LN$_2$. The RSCAD draft components' parameter specifications are given in Table 6.1, 6.2 and 6.3.

6.2.2.1 Component configuration: AC source component model

Table 6.1 presents the specifications for the AC source.

Table 6.1: AC source parameters

Parameter	Specification
Initial source magnitude	230 kV
Initial Frequency	50 Hz
Initial Phase	0.0 degrees

6.2.2.2 Component configuration: GTNET Component model

The GTNET card is connected to the RTDS through a GTIO port on the GPC/PB5 card. PMU components on the GTNET provide symmetrical three phase data of instantaneous voltage and current measurements. Table 6.2 presents configuration of the GTNET PMU_v4 component model.

Table 6.2: GTNET PMU_v4 Configuration

Parameter	Specification
Enable Output of C37.118 data using GTNET	Yes
GTNET Component Name	Virtual PMU
Configuration frame format	Config2
Base Frequency	50 (Hz)
Number of PMU's	2
Enable Primary Signals	Yes
GTIO Fiber Port Number	1
Assigned Control Processor	1
Priority Level	1
GTNET_PMU Card Number	1

6.2.2.3 Component configuration: PMU component model

A single PMU is set up and configured in the GTNET component. The C37.118 data output is enabled and administered by the GTSYNC card. Table 6.3 presents parameter specification for the Test PMU.

Table 6.3: Test PMU specifications

Parameter	Specification
Report Rate (frames / sec)	50
Decimate PMU runtime output	yes
Station Name	PMU1
Hardware ID	30
Output TCP/IP or UDP local port	4830
Number of phasors	6
Phasor Number Format	Real
Phasor Output Format	Cn & Phi
Analog Number Format	Real
Phasors 1 – 6 PMU output (respectively)	VA, VB, VC, IA, IB, IC

6.2.3 PMU Connection Tester

The PMU connection Testers' graphical user interface is presented in Figure 6.6 (CodePlex, 2016).

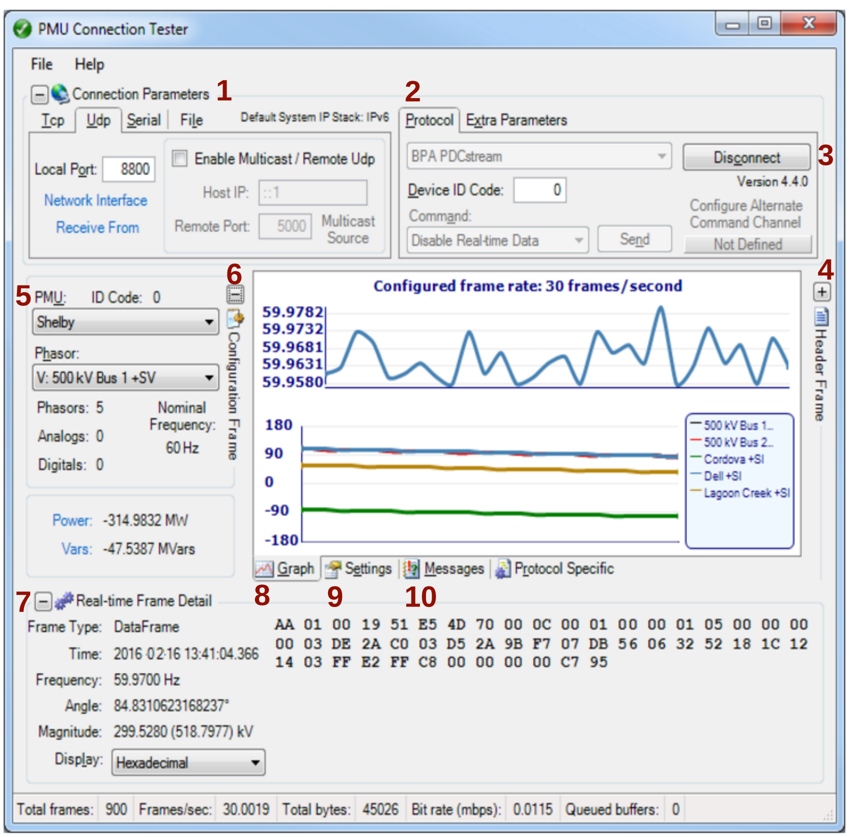


Figure 6.6: PMU Connection Tester graphic user interface (CodePlex, 2016)

The CodePlex PMU Connection Tester is a multi-protocol utility tool administered by Grid Protection Alliance (GPA) to validate real-time phasor data (CodePlex, 2016). The PMU connection Tester is used to validate the network settings before interfacing the PCDAA software and proceeding with the simulation study. Table 6.4 provides a description to the numeric labels in Figure 6.6.

Table 6.4: PMU Connection tester label description

Figure Label	Description
1	PMU connection parameter specification
2	phasor measurement data transmission protocol
3	PMU connection Tester connection status
4	header frame display
5	PMU ID selection
6	PMU configuration frame display
7	real-time frame detail
8	graph tab
9	settings tab
10	Message tab

6.2.4 Test results

The results for the communication and data transmission tests are obtained from the PMU connection Tester Graphical User Interface (GUI). With reference to Figure 6.6, the real-time frame display validates;

1. Frame type : phasor protocol frame type received
2. Time : UTC time value parsed from received frame
3. Frequency : configured report rate frequency specified in the frame
4. Angle : phase angle of the currently selected phasor
5. Magnitude : magnitude of the currently selected phasor

The PMU connection Tester shows data that coincides with PMU parameters specified in the GTNET PMU_v4 component, validating network configuration and real-time phasor data streaming between the RTDS and the control station interface software.

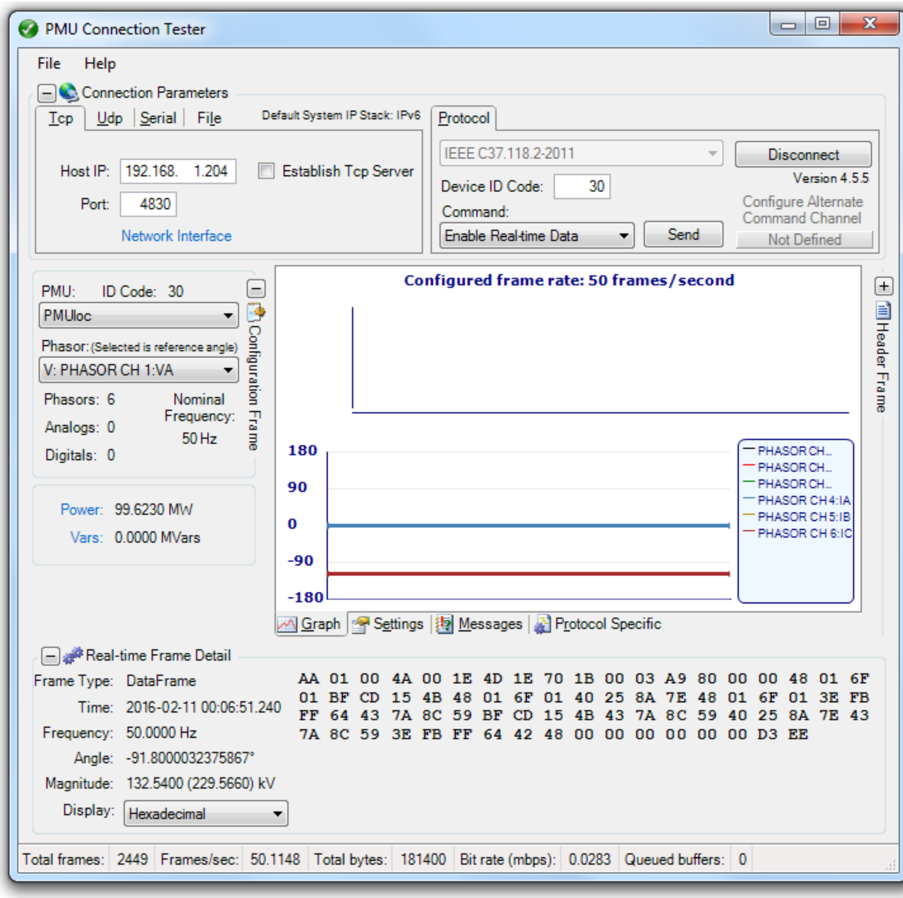


Figure 6.7: PMU communication network test results

6.3 Real-time simulation interface and control variables

This Section describes software interface used to implement and evaluate the transmission line differential protection schemes in real-time. Application of the RSCAD runtime environment and the developed PCDAA is detailed here.

6.3.1 RSCAD runtime environment

The simulation runtime interface modelled in RSCAD is presented in Figure 6.8. The interface is used to implement the real-time simulation in the RTDS. The runtime interface provides graphical displays which chart the operational characteristics of the transmission line transformer burden currents. The graphs also provide indication of fault inception and circuit breaker state.

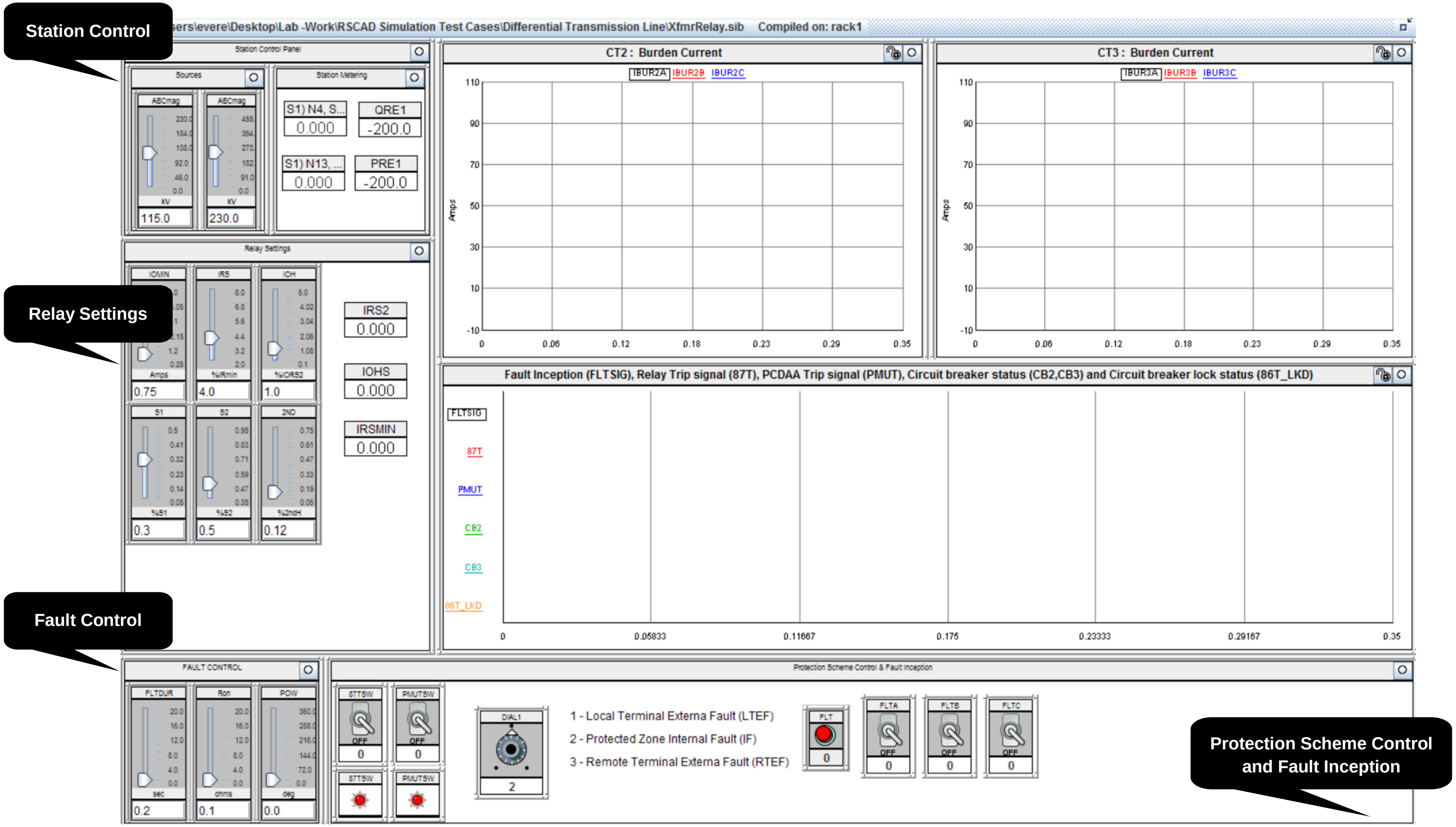


Figure 6.8: RSCAD Simulation runtime

The RSCAD runtime interface consist of;

Station Control Panel:

This panel subsists of two control sections which include;

- Source control

  This control section provides flexible terminal source control. A 115 kV source and a 230 kV source, on the wye and delta sides of the protected transformer respectively are defined here. It should be noted that the sources are not varied and are retained throughout the simulation study.

- Station metering

  This monitoring section provides indication of transmission line system node voltages.

Fault Control Panel:

The fault control panel specifies simulation case fault characteristics.

Relay Settings Panel:

Percentage characteristics of the relay can be configured here, providing value settings for;

- the operating current (S1)
- the new *steeper slope* (S2)
- the maximum allowable restraint current (IRS) utilized to define S2
- the maximum allowable restraint current (IRS2) utilized to define the relay *operate* condition

The minimum operating current (IOMIN) and 2nd harmonic parameters are also configured in this panel.

Protection Scheme Control and Fault Inception Panel:

This control panel provides control over which protection scheme to implement. It also extends fault inception specifications.

6.3.2 PCDAA utility software

The PCDAA Graphical User Interface (GUI) presented in Chapter Four, is used to implement the real-time simulation is presented in Figure 6.9. For each simulation case the PCDAA is used in conjunction with the RSCAD runtime interface.

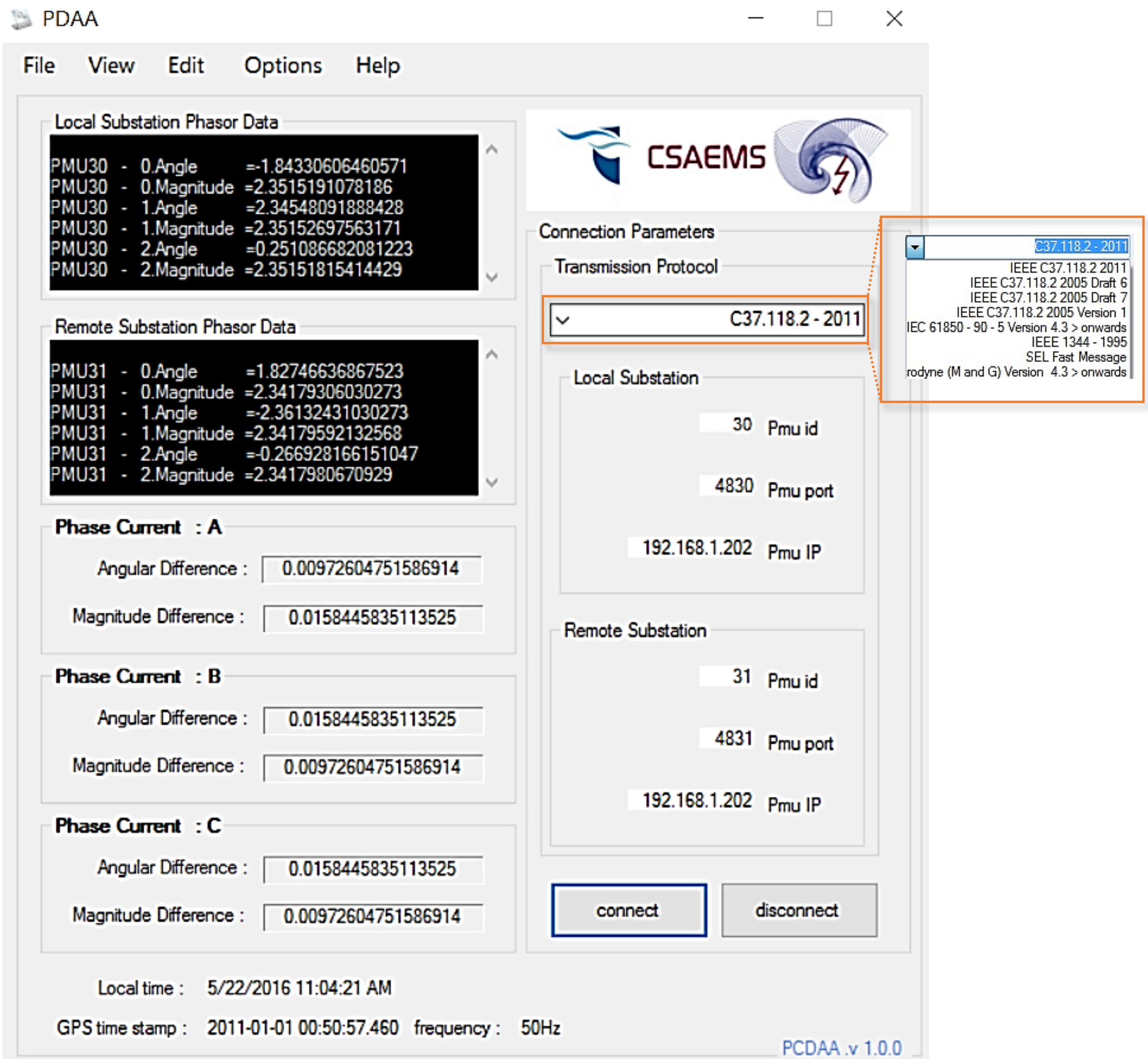


Figure 6.9: Engineered PCDAA utility software GUI

The PCDAA is an essential research contribution that provides an independent P-Class synchrophasor-based differential protection utility for the proposed PMU-based transmission line differential protection scheme.

6.3.3 Simulation control variables

Percentage restraint characteristics are employed by both the 87T-based differential protection scheme and the PMU-based differential protection scheme, where the ratio of the operating current to the restraint current is utilized to determine restraint effect and evaluate differential trip conditions.

For the 87T-based protection scheme, a differential relay is used to administer the percentage restraint characteristics. For the PMU-based protection scheme, the developed PCDAA is used to administer the percentage restraint characteristics. The percentage restraint characteristics provide selectivity and security against transient differential current that may result from current transformer (CT) saturation, making it ideal for both protection schemes.

For both protection schemes, the percentage restraint characteristics slope is established using slope settings specified in Table 6.5 and illustrated in Figure 6.10. The table also includes essential simulation control variables which are kept constant throughout all the case studies.

Table 6.5: Simulation control variables

Parameter specification	Value	Units
Local Terminal AC source	115	kV
Remote Terminal AC source initial phase	-360	Degrees
Local Terminal AC source	230	kV
Remote Terminal AC source initial phase	-360	Degrees
(2nd Harmonic)	0.12	%
(S1) – operate current	0.3	%
(S2) – steeper slope, definition point	0.5	%
(IRSMIN) – minimum restraint current	4.0	%
(IRS2) – breakpoint of dual slope characteristic	0.75	%
(IOMIN) – minimum operating current	1.0	Amps
Fault duration	0.2	seconds

The specified control variables facilitate evaluation of;

- the breakpoint of the dual slope
- the restraint current necessary to trigger trip transfer(IOHS)
- the minimum restraint current

Table 6.6 presents the established percentage restraint characteristic slope settings.

Table 6.6: Percentage restraint slope settings

	Slope parameter	Reading (Amps)
Breakpoint current	IRS2	10.00
Operating trip Current	IOHS	20.00
Minimum restraint current	IRSMIN	2.500

Figure 6.10 graphically illustrates the percentage restraint characteristic employed by the 87T relay and the PCDAA.

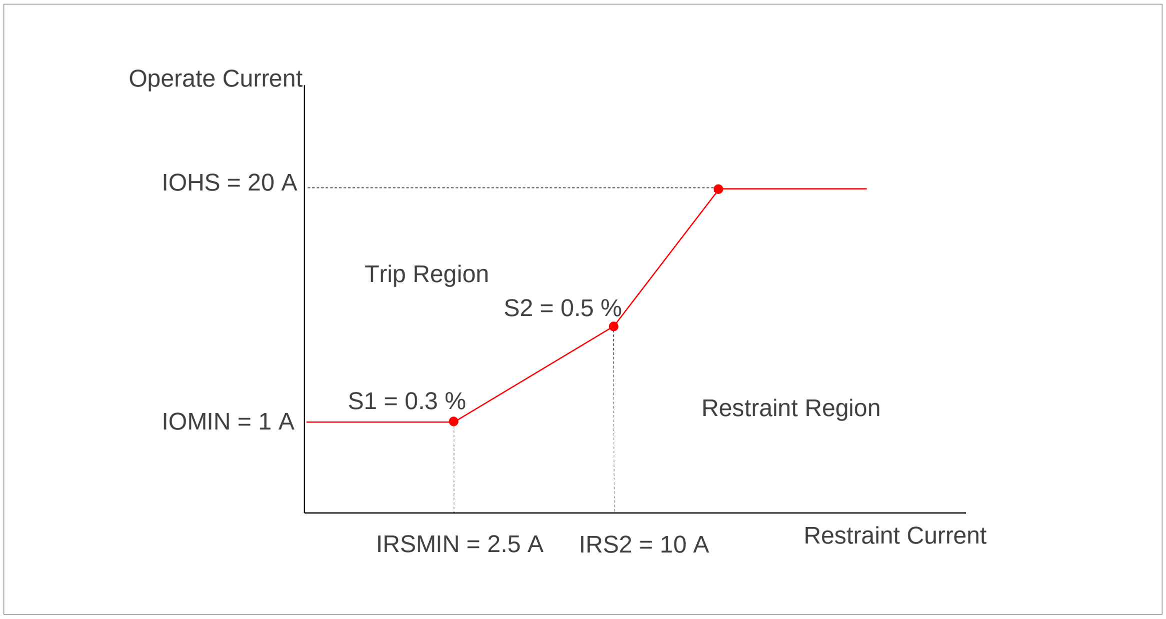


Figure 6.10: Percentage restraint characteristics implemented by the 87T relay and the developed PCDAA

6.4 Case study one:

This case study evaluates the operational characteristic and performance of the 87T-based protection scheme and the PMU-based protection scheme over the same transmission line system for the line to ground fault case illustrated in Figure 6.11.

Figure 6.11: Local terminal external fault case

RSCAD is used to implement and evaluate the response of the 87T-based protection scheme. The developed PCDAA software is used to implement and evaluate the characteristic operation of the proposed PMU-based protection scheme.

When the Local Terminal External Fault (LTEF) is applied, the response of the conventional 87T-based protection scheme is compared to the response of the PMU -based protection scheme. The correlation and comparative analysis of the obtained results is detailed in a discussion presented at the end of the case study.

Figure 6.12 presents the RSCAD runtime interface results obtained from real-time simulation of the LTEF case.

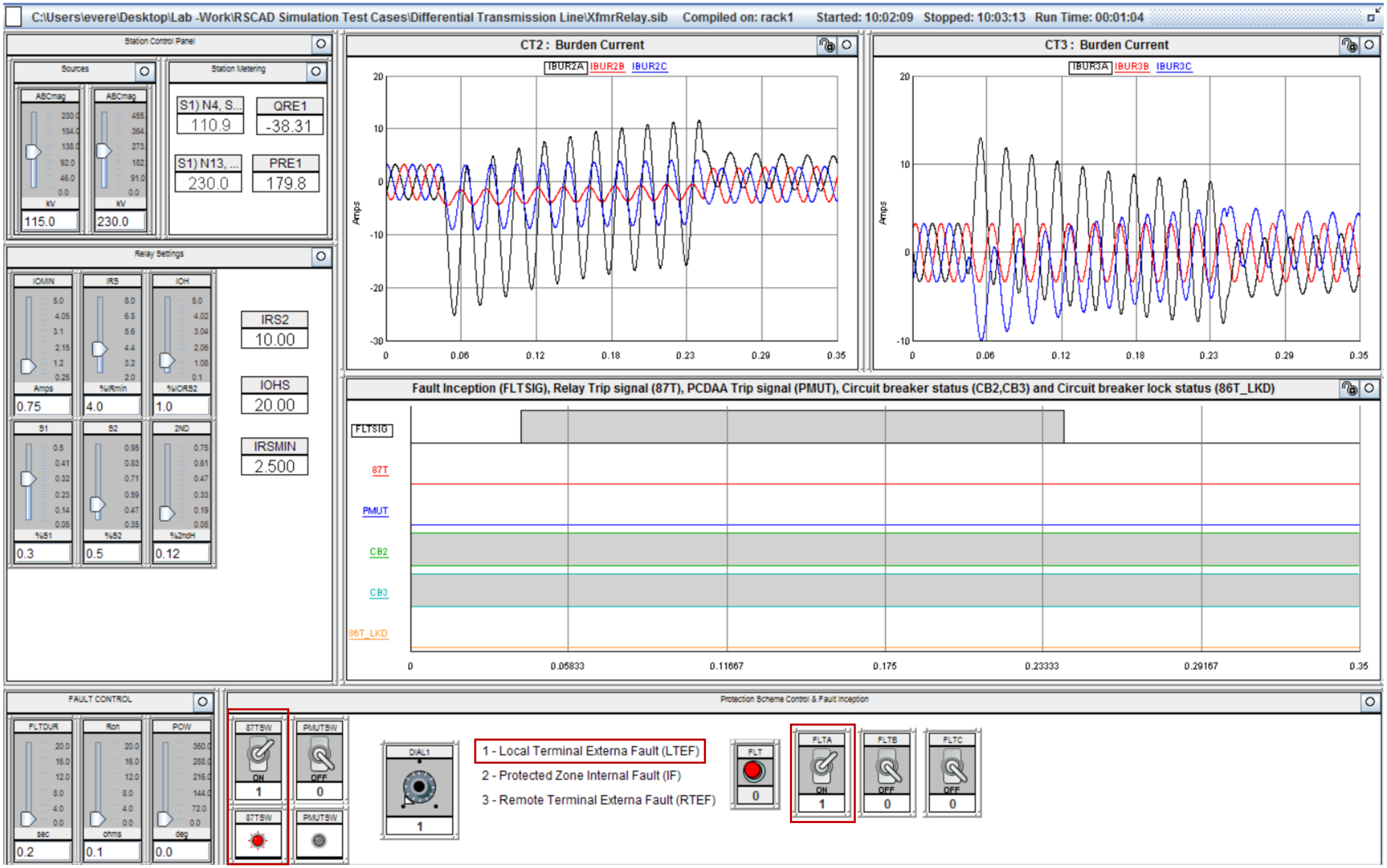


Figure 6.12: Real-time simulation results for the LTEF case implementing 87T-based differential protection

6.4.1 87T-based protection scheme implementation

When the LTEF is introduced into the real-time transmission line simulation, the response of the 87T-based differential protection scheme is exhibited in the runtime interface, as illustrated Figure 6.12. This Section presents and examines results assimilated from the runtime environment.

6.4.1.1 Current transformer burden current analysis

Figure 6.13 illustrates characteristic response for CT2 burden current. The graph presents the characteristic response of the CT2 burden current when the LTEF is introduced into the simulation of the transmission line system in real-time. The graph maps the behavior of the burden currents; IBUR2A, IBUR2B and IBUR2C for the duration; $(0 \leq t \leq +0.35)$ seconds, where t represents the independent time variable. CT2 burden current magnitudes vary in the range; $(-25.56 \leq A \leq +11.70)$ Amps where A represents the dependent Ampere variable.

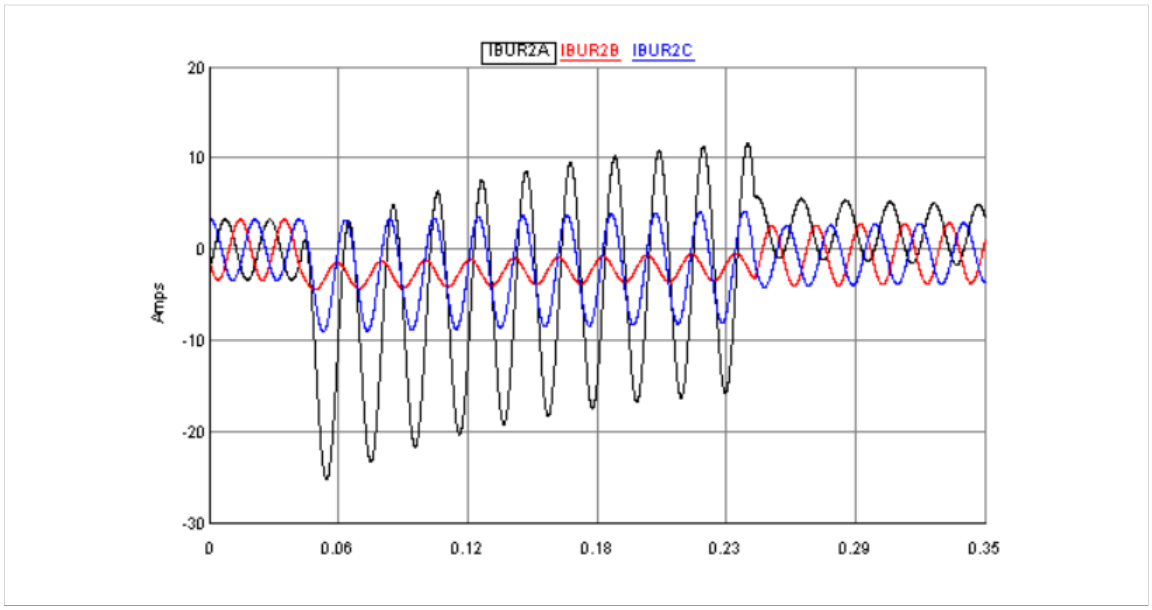


Figure 6.13: CT2 Burden Current

Figure 6.14 displays the behavior of the CT3 burden currents; IBUR3A, IBUR3B and IBUR3C for the duration; $(0 \leq t \leq +0.35)$ seconds. The CT3 burden current magnitudes vary in the range $(-10 \leq A \leq +13.05)$ Amps.

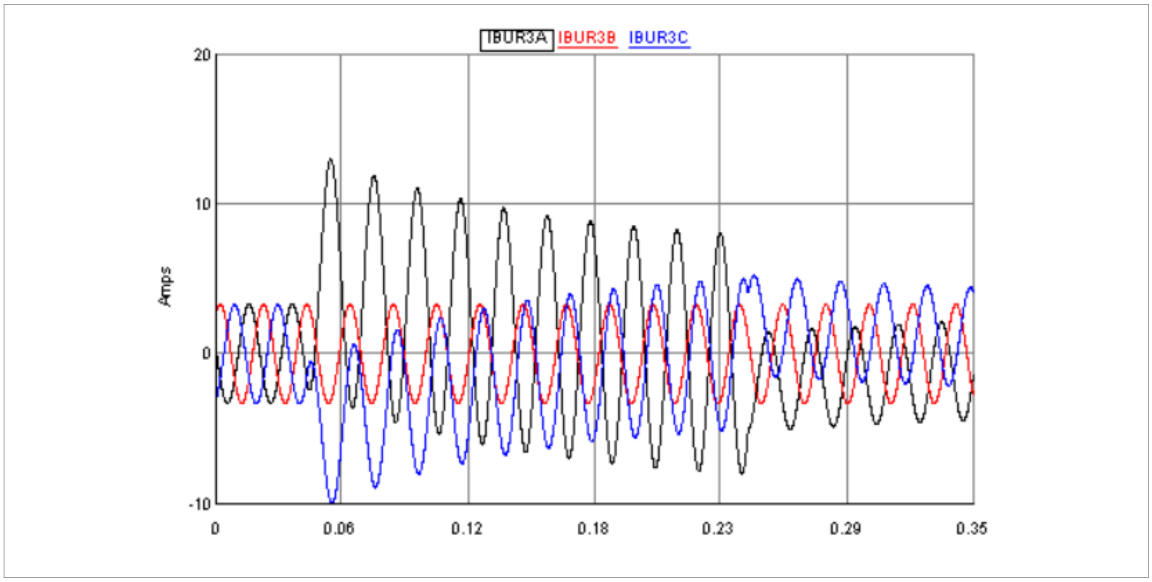


Figure 6.14: CT3 Burden Current

Description of the resulting current transformer burden currents is given Table 6.7 and Table 6.8.

Table 6.7: CT2 Burden Current Characteristic

Time - Interval	Description: CT2 Burden Current Characteristic Response
$0 \leq t \leq +0.049$	Before the LTEF is introduced, IBUR2A, IBUR2B and IBUR2C repetitively oscillate for the range; $(-3.28 \leq A \leq +3.28)$ Amps. This repetitive oscillation indicates system stability prior fault inception.
$+0.049 \leq t \leq +0.276$	When the line to ground fault is applied to phase A in the LTEF case: • IBUR2A surges sharply, oscillating for the range; $(-24.58 \leq A \leq +11.51)$ Amps. • IBUR2B decreases and becomes completely negative, oscillating for the range; $(-4.32 \leq A \leq -0.66)$ Amps. • IBUR2C increases moderately, oscillating for the range; $(-9.15 \leq A \leq +4.17)$ Amps.
$+0.276 \leq t \leq +0.350$	IBUR2A, IBUR2B and IBUR2C repetitively oscillate for the range; $(-3.75 \leq A \leq +5.33)$ Amps.

Table 6.8: CT3 Burden Current Characteristic

Time - Interval	Description: CT3 Burden Current Characteristic Response
$0 \leq t \leq +0.049$	Before the LTEF is introduced, IBUR3A, IBUR3B and IBUR3C repetitively oscillate for the range; $(-3.28 \leq A \leq +3.28)$ Amps. This repetitive oscillation indicates system stability prior fault inception
$+0.049 \leq t \leq +0.276$	When the line to ground fault is applied to phase A in the LTEF case: • IBUR3A surges sharply, oscillating for the range; $(-8.15 \leq A \leq +12.93)$ Amps. • IBUR3B remains constant, oscillating for the range; $(-3.28 \leq A \leq +3.28)$ Amps. • IBUR3C increases moderately, oscillating for the range; $(-10.20 \leq A \leq +5.17)$ Amps.
$+0.276 \leq t \leq +0.350$	IBUR3A, IBUR3B and IBUR3C repetitively oscillate for the range; $(-4.67 \leq A \leq +4.25)$ Amps.

When the line to ground fault is introduced in the LTEF case, the degraded burden currents indicated by the resultant oscillating characteristics demonstrate asymmetric current transformer saturation in CT2 and CT3.

6.4.1.2 Direct trip transfer and circuit breaker state report

Figure 6.15 indicates the operational state and characteristic response of the 87T-based protection scheme after inception of the fault. The horizontal graphs indicate the state and duration of the;

- fault signal (FLTSIG)
- transfer status of the differential relay trip signal (87T)
- transfer status of the PCDAA trip signal (PMUT)
- local terminal circuit breaker (CB2) active state status
- remote terminal circuit breaker (CB3) active state status
- the circuit breaker lock active state status (86T)

The graphs represent respective status conditions for the duration of the fault, with a time window which ranges ($0 \leq t \leq +0.35$) seconds.

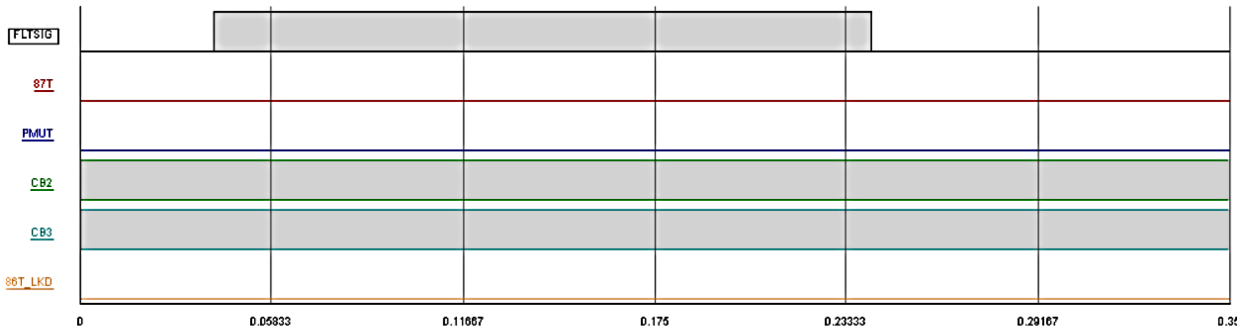


Figure 6.15: Fault signal status, Relay state, CT2 state and CT3 state

The graph characteristic descriptions are given in Table 6.9.

Table 6.9: LTEF case study simulation runtime protective characteristic results

Logical State		Time point	Description
FLTSIG	1	$+0.054 \leq t \leq +0.255$	The fault signal is incepted for 0.2 seconds
87T	0	-	The differential trip signal is not transferred to open the breaker.
PMUT	-	-	-
CB2	1	-	Circuit breaker 2 remains closed
CB3	1	-	Circuit breaker 3 remains closed
86T_LKD	0	-	The breakers remain unlocked

As indicated in Figure 6.15 no trip signal is transferred to open the breakers. The percentage restraint characteristic (Figure 6.10) employed by the real-time 87T-based protection scheme exhibits;

- selectivity, identifying the external fault
- restrained sensitivity, employing higher slope security to accommodate transient differential current resulting from asymmetric transformer saturation

6.4.2 PMU-based protection scheme implementation

In the second part of this case study, the PCDAA utility is administered to implement the PMU-based differential protection scheme. The secondary burden currents of CT2 and CT3 are represented by synchrophasors from PMU_1 and PMU_2 which are transmitted to the PCDAA where they are validated (re-aligned if necessary), and processed to evaluate the fault condition of the transmission line system.

Figure 6.16 presents instantaneous real-time synchrophasor data provided by the PCDAA utility software throughout inception of the LTEF. Figure 6.16 displays synchrophasor value readings that correspond to; the no fault condition, inception of the LTEF and post fault analysis.

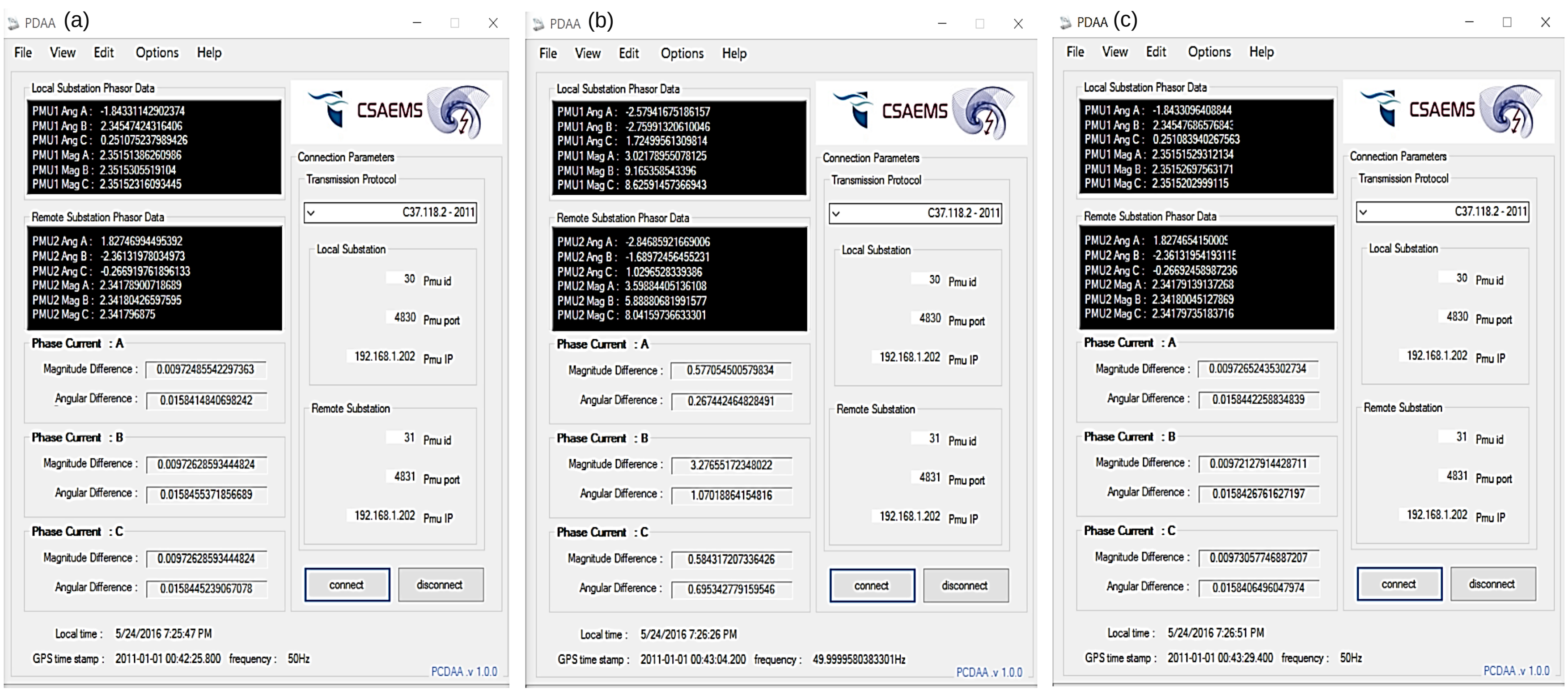


Figure 6.16: PCDAA synchrophasor data – LTEF case:

(a) no fault condition

(b) fault – inception state

(c) post fault state

In this simulation study the PCDAA software utility is used in conjunction with the RSCAD simulation runtime illustrated in Figure 6.17.

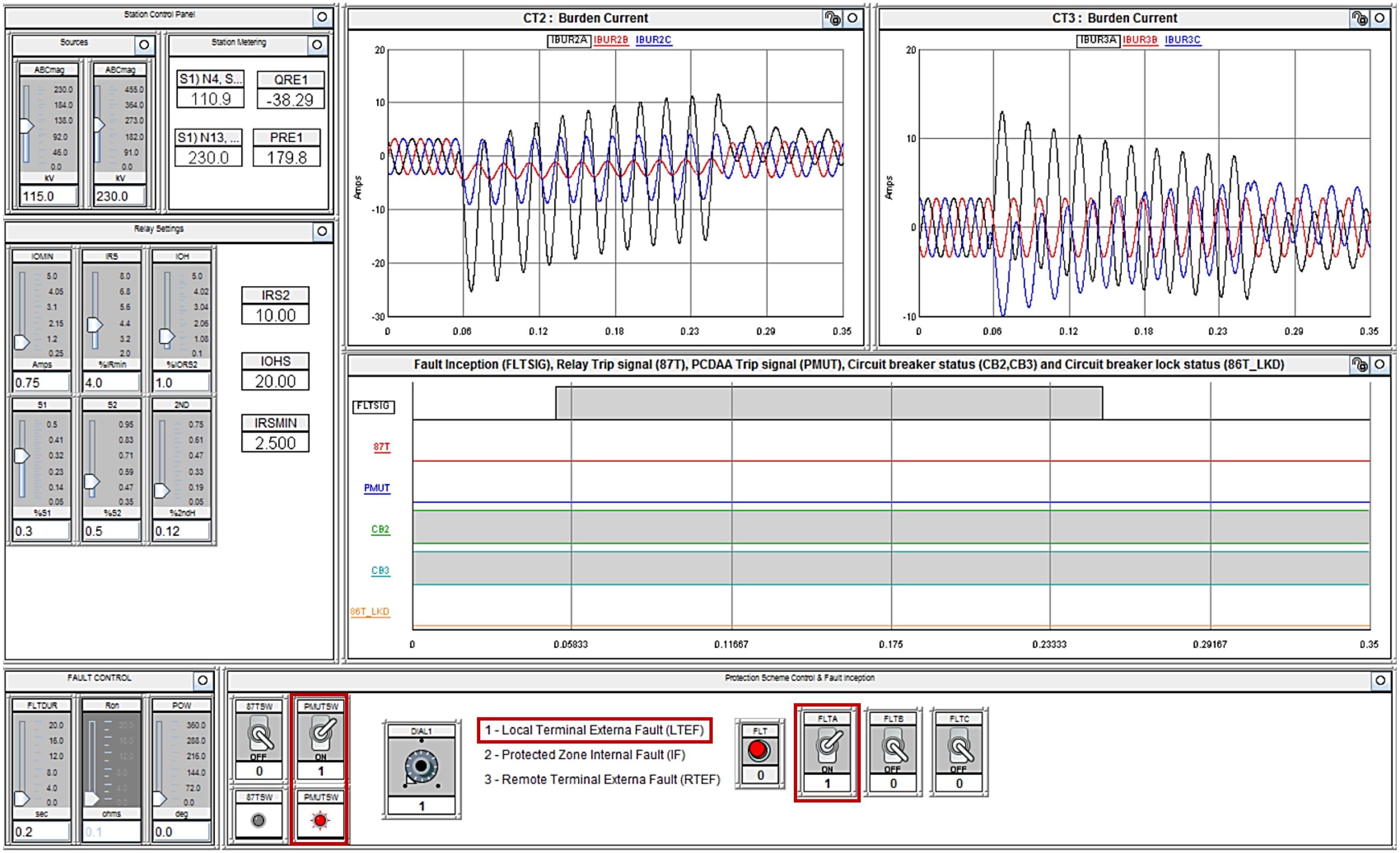


Figure 6.17: Real-time simulation results for the LTEF case implementing PMU-based differential protection

6.4.2.1 PCDAA transformer burden current

Table 6.10 presents real-time line terminal synchrophasor data, provided by the developed PCDAA.

Table 6.10: PMU burden current differential analysis

	No fault condition reading	fault condition reading	post-fault reading
PMU1 : Phasor angular component (θ_1) Degrees			
Phase A	-1.843311429024	-2.579416751861	-1.843309640884
Phase B	2.345474243164	-2.759913206100	2.345476865768
Phase C	0.251275237989	1.724995613098	0.251083940267
PMU1 : Phasor magnitude component ($\bar{\iota}_1$) Amps			
Phase A	2.351513862609	3.021789550781	2.351515293121
Phase B	2.351530551910	9.165358554339	2.351526975631
Phase C	2.351523160934	8.625914573669	2.351520299911
PMU2 : Phasor angular component (θ_2) Degrees			
Phase A	1.827469944954	-2.846859216690	1.827465415000
Phase B	-2.361319780350	-1.689724564552	-2.361319541931
Phase C	-0.266919761896	1.029652833839	-0.266924589872
PMU2 : Phasor magnitude component ($\bar{\iota}_2$) Amps			
Phase A	2.34178900719	3.598844051361	2.341791391372
Phase B	2.34180426598	5.888806811991	2.341800451278
Phase C	2.34179687500	8.041597366333	2.341797351837
(PMU2 - PMU1) : Magnitude Differential ($\bar{\iota}_3$) Amps			
Phase A	0.009724855422	0.577054500580	0.009726524353
Phase B	0.009726285934	3.276551723480	0.009721279144
Phase C	0.009726285934	0.584317207336	0.0097305774688
(PMU2 - PMU1) : Angle Differential (θ_3) Degrees			
Phase A	0.015841484070	0.267442464828	0.015844225883
Phase B	0.015841484070	1.070188641548	0.015842676162
Phase C	0.015841484070	0.695342779159	0.015840649604

Graphical presentation of the instantaneous synchrophasors obtained from the PCDAA are shown in Figure 6.18.

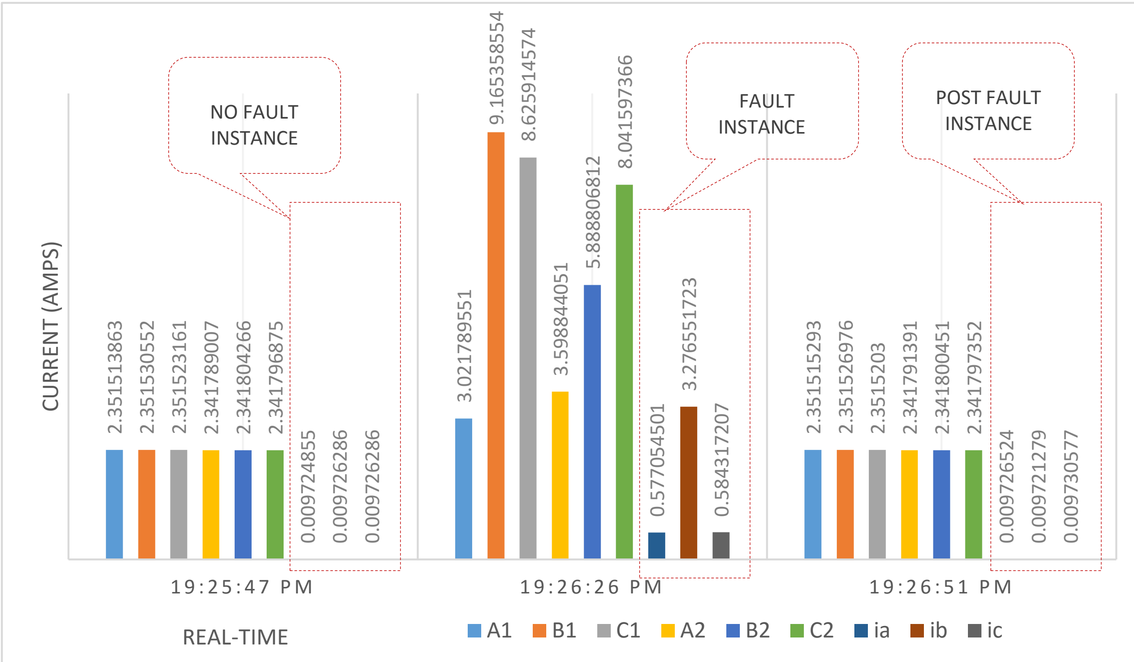


Figure 6.18: Analysis of PCDAA instantaneous synchrophasor results for the LTEF case

Where (A1-C1) represents PMU_1 synchrophasors, (A2-C2) represents PMU_2 synchrophasors and (ia –ic) represents the synchrophasor-based current differentials. Figure 6.18 emphasizes;

- No fault instance: near zero current differential
- Fault instance: fictitious differential current
- Post fault instance: near zero current differential

The PMU-based protection scheme employs the PCDAA to implement synchrophasor based current differential protection. The PCDAA administers a percentage restrained characteristic algorithm which facilitates protection for the transmission line system. The PMU-based protection scheme exhibits protection response characteristics which are indistinguishable from the conventional 87T-based protection scheme

6.4.2.2 Direct trip transfer and circuit breaker state report

Figure 6.12 indicates the operational state and characteristic response of the 87T-based protection scheme after inception of the fault. The horizontal bars indicate the state and duration of the;

- fault signal (FLTSIG)
- transfer status of the differential relay trip signal (87T)
- transfer status of the PCDAA trip signal (PMUT)
- local terminal circuit breaker (CB2) active state logic
- remote terminal circuit breaker (CB3) active state logic
- the circuit breaker lock active state logic (86T)

The graphs represent respective status conditions for the duration of the fault, with a time window which ranges ($0 \leq t \leq +0.35$) seconds.

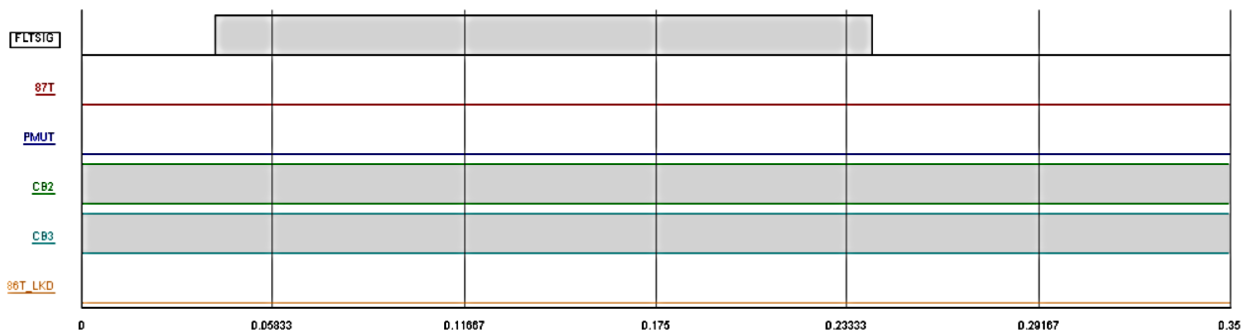


Figure 6.19: Fault signal status, Relay state, CT2 state and CT3 state

The graph characteristic description for Figure 6.14 is given in Table 6.11.

Table 6.11: LTEF case study simulation runtime protective characteristic results

Logical State		Time point	Description
FLTSIG	1	$+0.054 \leq t \leq +0.255$	The fault signal is incepted for 0.2 seconds
PMUT	0	-	The differential trip signal is not transferred to open the breaker.
87T	-	-	-
CB2	1	-	Circuit breaker 2 remains closed
CB3	1	-	Circuit breaker 3 remains closed
86T_LKD	0	-	The breakers remain unlocked

The results illustrated by the graphs highlight how the PMU-based protection scheme demonstrates characteristic response similar to the 87T-based protection scheme.

6.4.3 Discussion of case study results

For the outlined case study, control variables are maintained, and both the 87T-based protection scheme and the PMU-based protection scheme are implemented over parallel conditions. The experimental investigation centers on inception of an external fault LTEF, analyzing the real-time characteristic response of the transmission line system under protection from each scheme exclusively. For the case study, both the PMU-based protection scheme and the 87L demonstrate indistinguishable protection security characteristics, identifying the external fault and averting operation of the line terminal circuit breakers to establish selective protection and adequate line security. The PMU-based line differential protection scheme utilizes synchrophasors, and administers the developed PCDAA to employ the percentage differential scheme.

6.5 Case Study two:

This case study evaluates the operational characteristic and performance of the 87T-based protection scheme and the PMU-based protection scheme over the same transmission line system for the line to ground fault case illustrated in Figure 6.20.

Figure 6.20: Remote terminal external fault case

RSCAD is used to implement and evaluate the response of the 87T-based protection scheme. The developed PCDAA software is used to implement and evaluate the characteristic operation of the proposed PMU-based protection scheme.

When the Remote Terminal External Fault (RTEF) is applied, the response of the conventional 87T-based protection scheme is compared to the response of the PMU -based protection scheme. The correlation and comparative analysis of the obtained results is detailed in a discussion presented at the end of the case study.

Figure 6.21 presents the RSCAD runtime interface results obtained from real-time simulation of the RTEF case.

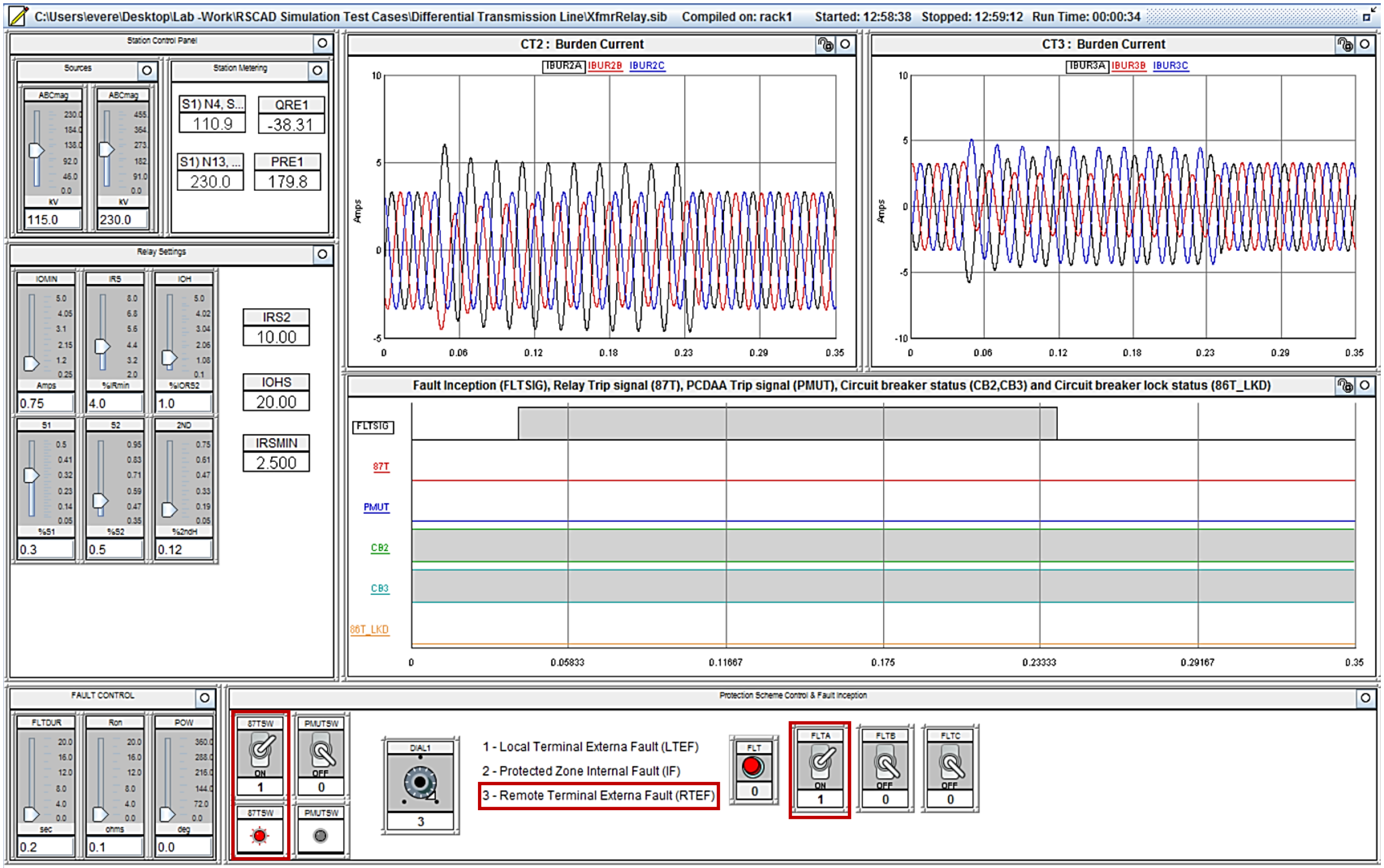


Figure 6.21: Real-time simulation results for the RTEF case implementing 87T-based differential protection

6.5.1 87T-based protection scheme implementation

When the RTEF is introduced into the real-time transmission line simulation, the response of the 87T-based differential protection scheme is exhibited in the runtime interface, as illustrated Figure 6.21. This Section presents and examines results assimilated from the runtime environment.

6.5.1.1 Current transformer burden current analysis:

Figure 6.22 illustrates characteristic response for CT2 burden current. The graph presents the characteristic response of the CT2 burden current when the RTEF is introduced into the simulation of the transmission line system in real-time. The graph maps the behavior of the burden currents; IBUR2A, IBUR2B and IBUR2C for the duration; $(0 \leq t \leq +0.35)$ seconds, where t represents the independent time variable. CT2 burden current magnitudes vary in the range; $(-4.71 \leq A \leq +6.06)$ Amps where A represents the dependent Ampere variable.

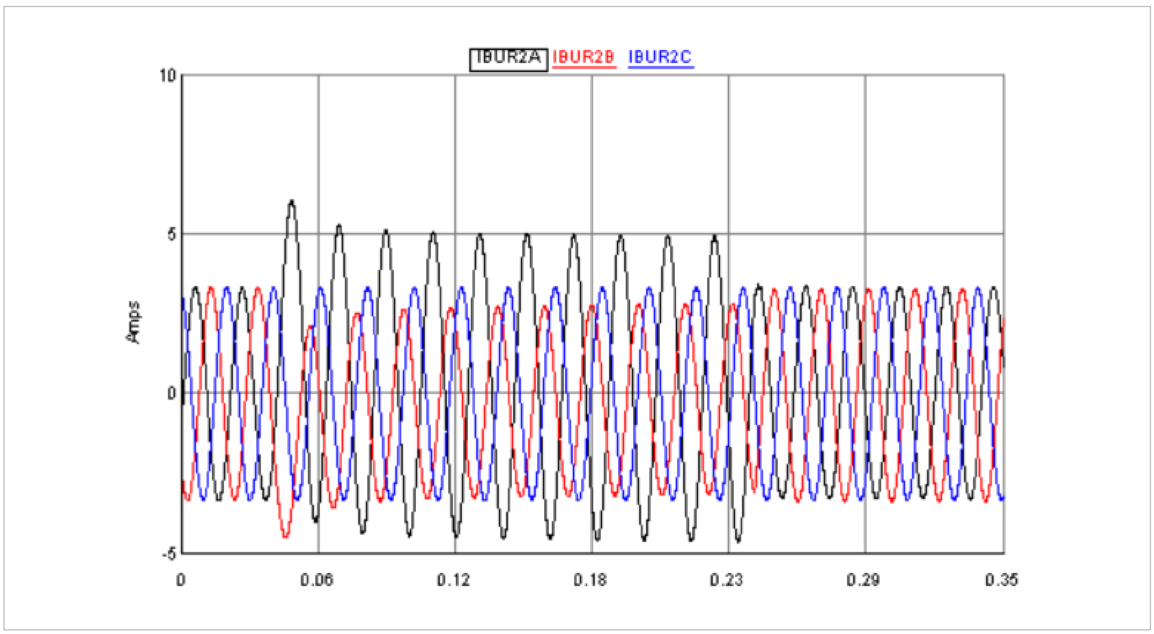


Figure 6.22: CT2 Burden Current

Figure 6.23 displays the behavior of the CT3 burden currents; IBUR3A, IBUR3B and IBUR3C for the duration; $(0 \leq t \leq +0.35)$ seconds. The CT3 burden current magnitudes vary in the range $(-5.75 \leq A \leq +5.06)$ Amps.

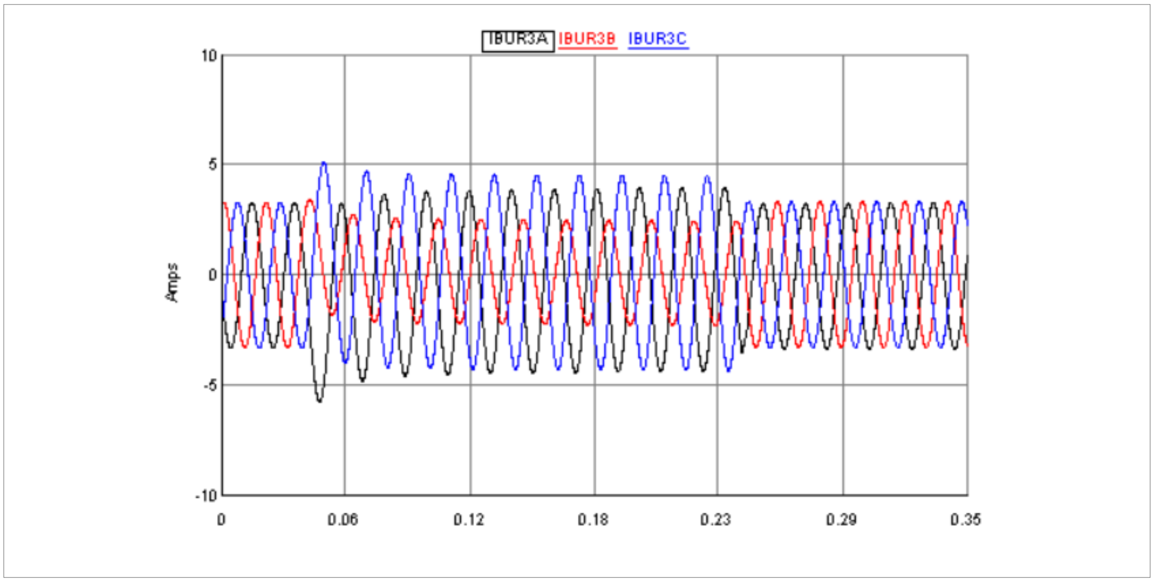


Figure 6.23: CT3 Burden Current

Description of the resulting current transformer burden currents is given in Table 6.12 and Table 6.13.

Table 6.12: CT2 Burden Current Characteristic

Time - Interval	Description: CT2 Burden Current Characteristic Response
$0 \le t \le +0.037$	Before the RTEF is introduced, IBUR2A, IBUR2B and IBUR2C repetitively oscillate for the range; $(-3.28 \le A \le +3.28)$ Amps. This repetitive oscillation indicates system stability prior fault inception
$+0.037 \le t \le +0.243$	When the line to ground fault is applied to phase A in the RTEF case: • IBUR3A surges sharply, oscillating for the range; $(-4.71 \le A \le 6.06)$ Amps. • IBUR3B remains constant, oscillating for the range; $(-4.48 \le A \le +2.76)$ Amps. • IBUR3C decreases slightly, oscillating for the range; $(-3.32 \le A \le +3.28)$ Amps.
$+0.243 \le t \le +0.350$	IBUR2A, IBUR2B and IBUR2C repetitively oscillate for the range; $(-3.32 \le A \le +3.28)$ Amps.

Table 6.13: CT3 Burden Current Characteristic

Time - Interval	Description: CT3 Burden Current Characteristic Response
$0 \leq t \leq +0.037$	Before the RTEF is introduced, IBUR3A, IBUR3B and IBUR3C repetitively oscillate for the range $(-3.28 \leq A \leq +3.28)$ Amps. This repetitive oscillation indicates system stability prior fault inception
$+0.037 \leq t \leq +0.243$	When the line to ground fault is applied to phase A in the RTEF case: • IBUR3A surges sharply, oscillating for the range; $(-5.83 \leq A \leq +3.89)$ Amps. • IBUR3B remains constant, oscillating for the range; $(-6.80 \leq A \leq +2.20)$ Amps. • IBUR3C decreases slightly, oscillating for the range; $(-4.44 \leq A \leq +4.44)$ Amps.
$+0.243 \leq t \leq +0.350$	IBUR3A, IBUR3B and IBUR3C repetitively oscillate for the range; $(-3.28 \leq A \leq +3.28)$ Amps.

Similarly to the LTEF case, when the line to ground fault is introduced in the RTEF case, the degraded burden currents indicated by the resultant oscillating characteristics demonstrate asymmetric current transformer saturation in CT2 and CT3.

6.5.1.2 Direct trip transfer and circuit breaker state report

Figure 6.24 indicates the operational state and characteristic response of the 87T-based protection scheme after inception of the fault. The horizontal graphs indicate the state and duration of the;

- fault signal (FLTSIG)
- transfer status of the differential relay trip signal (87T)
- transfer status of the PCDAA trip signal (PMUT)
- local terminal circuit breaker (CB2) active state status
- remote terminal circuit breaker (CB3) active state status
- the circuit breaker lock active state status (86T)

The graphs represent respective status conditions for the duration of the fault, with a time window which ranges $(0 \leq t \leq +0.35)$ seconds.

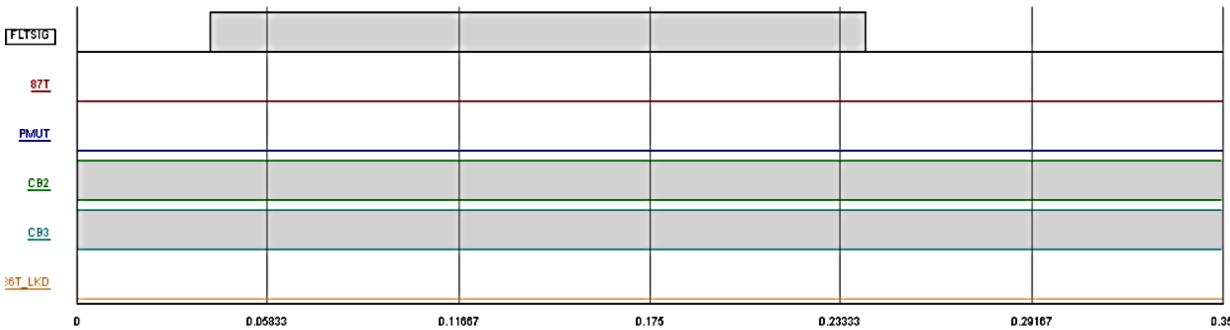


Figure 6.24: Fault signal status, Relay state, CT2 state and CT3 state

The graph characteristic descriptions are given in Table 6.14.

Table 6.14: RTEF case study simulation runtime protective characteristic results

Logical State		Time point	Description
FLTSIG	1	$+0.054 \leq t \leq +0.255$	The fault signal is incepted for 0.2 seconds
87T	0	-	The differential trip signal is not transferred to open the breaker.
PMUT	-	-	-
CB2	1	-	Circuit breaker 2 remains closed
CB3	1	-	Circuit breaker 3 remains closed
86T_LKD	0	-	The breakers remain unlocked

The percentage differential characteristic security employed by the relays prevents mal-operation and no trip signal is transferred to open the breakers. The PMU-based protection scheme is not utilized and its trip signal (PMUT) is not considered.

6.5.2 PMU-based protection scheme implementation

In the second part of this real-time simulation case study, the PCDAA utility is administered to implement the PMU-based differential protection scheme. The secondary burden currents of CT2 and CT3 are represented by synchrophasors from PMU_1 and PMU_2 which are transmitted to the PCDAA where they are validated (re-aligned if necessary), and processes to evaluate the fault condition of the transmission line system.

Figure 6.25 presents instantaneous real-time synchrophasor data provided by the PCDAA utility software throughout the simulation study. The figure displays synchrophasor value readings that correspond to; the no fault condition, inception of the RTEF and post fault analysis.

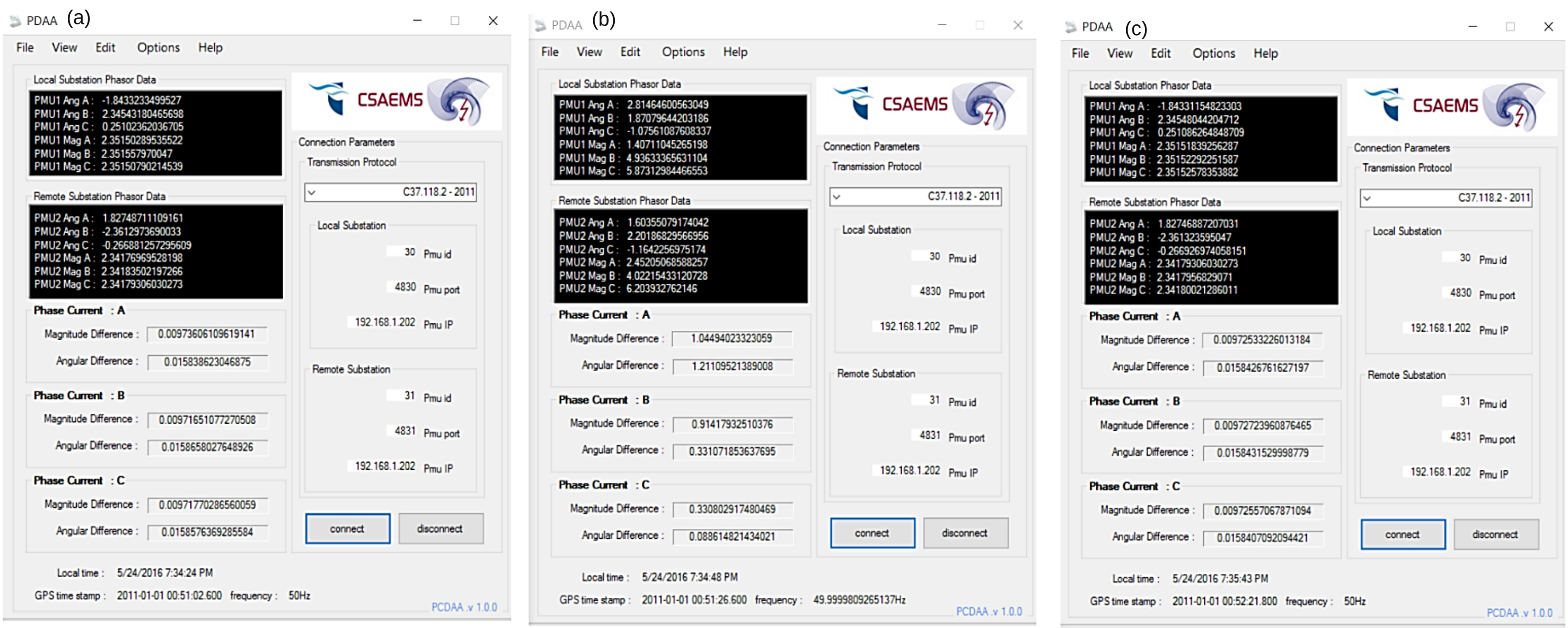


Figure 6.25: PCDAA synchrophasor data – RTEF case:

(a) no fault condition

(b) fault – inception state

(c) post fault state

In this simulation study the PCDAA software utility is used in conjunction with the RSCAD simulation runtime illustrated in Figure 6.26.

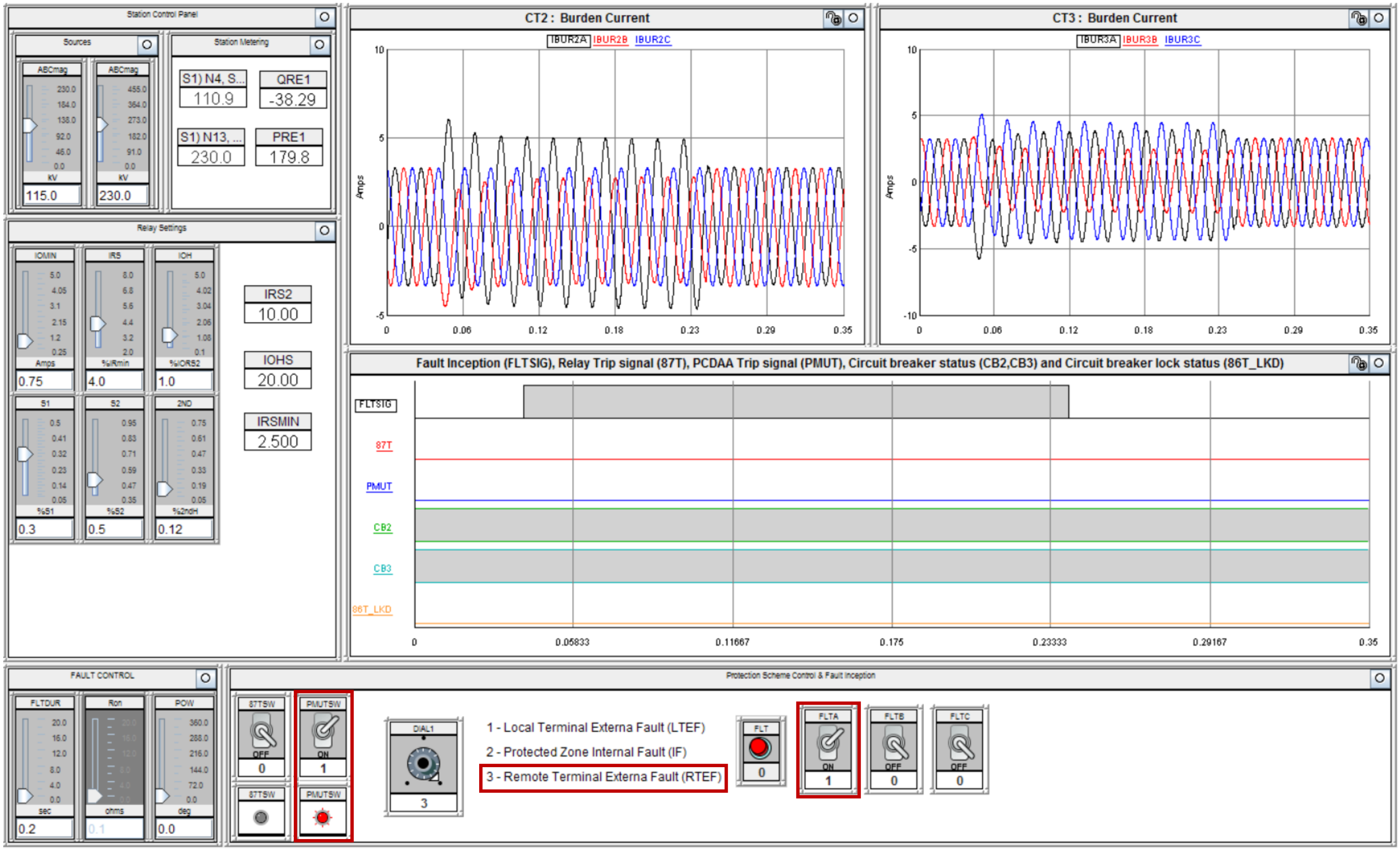


Figure 6.26: Real-time simulation results for the RTEF case implementing PMU-based differential protection

6.5.2.1 PCDAA transformer burden current

Table 6.15 presents real-time line terminal synchrophasor data.

Table 6.15: PMU burden current differential analysis

	No fault condition reading	fault condition reading	post-fault reading
PMU_1 : Phasor angular component (θ_1) Degrees			
Phase A	-1.843323349952	2.814646005630	-1.843311548233
Phase B	2.345431804656	1.870796442031	2.345480442047
Phase C	0.251023620367	-1.075610876083	0.251086264848
PMU_1 : Phasor magnitude component ($\bar{\iota}_1$) Amps			
Phase A	2.351502895355	1.407110452651	2.351518392562
Phase B	2.351557970047	4.936333656311	2.351522922515
Phase C	2.351507902145	5.873129844665	2.351525783538
PMU_2 : Phasor angular component (θ_2) Degrees			
Phase A	1.827487111091	1.603550791740	1.827468872070
Phase B	-2.361297369003	2.201868295669	-2.361323595047
Phase C	-0.266881257295	-1.256975171642	-0.266926974058
PMU_2 : Phasor magnitude component ($\bar{\iota}_2$) Amps			
Phase A	2.341769695281	2.452050685882	2.341793060302
Phase B	2.341835021972	4.022154331207	2.341795682607
Phase C	2.341793060302	6.203932762146	2.341800212860
(PMU2 - PMU1) : Magnitude Differential ($\bar{\iota}_3$) Amps			
Phase A	0.009736061096	1.044940233230	0.009725332260
Phase B	0.009716510772	0.914179325103	0.009727239608
Phase C	0.009717702865	0.330802917480	0.009725570678
(PMU_2 - PMU_1) : Angle Differential (θ_3) Degrees			
Phase A	0.015838623046	1.211095213890	0.015842676162
Phase B	0.015865802764	0.331071853637	0.015843152999
Phase C	0.015857636928	0.088614821434	0.015840709209

Graphical presentation of the instantaneous synchrophasors obtained from the PCDAA are shown in Figure 6.27.

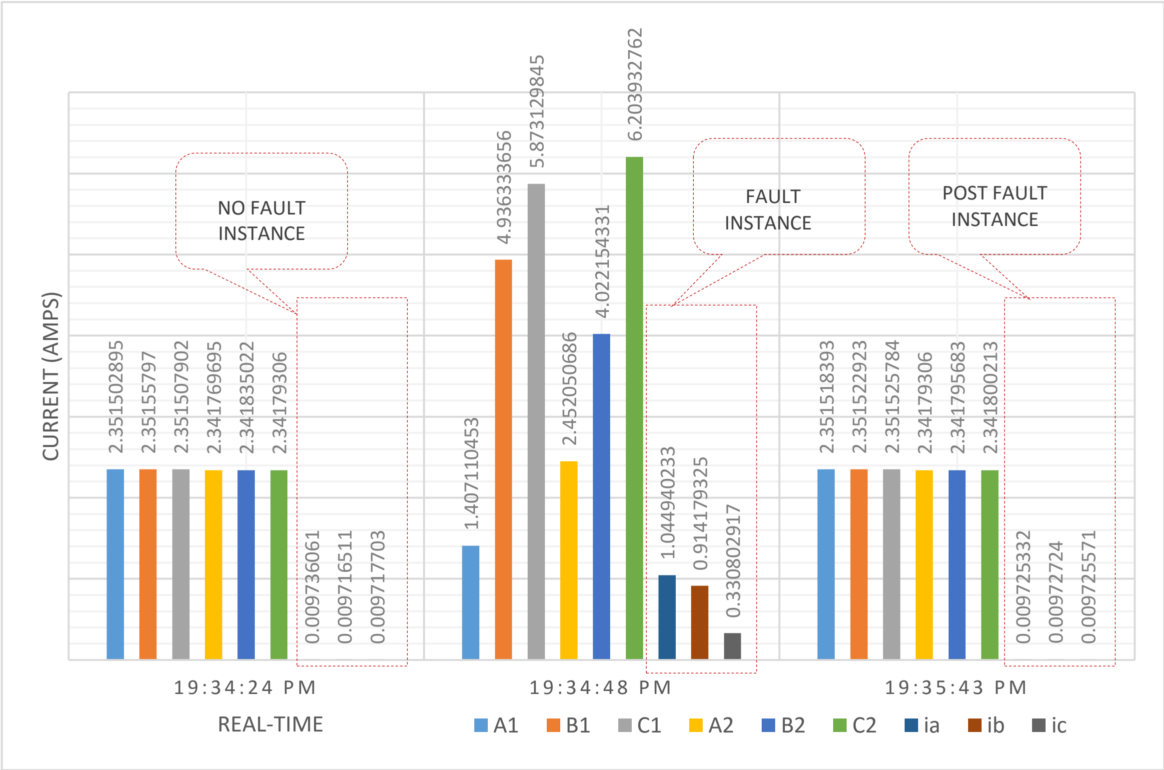


Figure 6.27: Analysis of PCDAA instantaneous synchrophasor results for the RTEF case

Where (A1-C1) represents PMU_1 synchrophasors, (A2-C2) represents PMU_2 synchrophasors and (ia –ic) represents the synchrophasor-based current differentials. Figure 6.27 emphasizes;

- No fault instance - near zero current differential
- Fault instance - fictitious differential current
- Post fault instance - near zero current differential

The PMU-based protection scheme employs the PCDAA to implement synchrophasor based current differential protection. The PCDAA administers a percentage restrained characteristic algorithm which facilitates protection for the transmission line system. The PMU-based protection scheme exhibits protection response characteristics which are indistinguishable from the conventional 87T-based protection scheme

6.5.2.2 Direct trip transfer and circuit breaker state report

Figure 6.28 indicates the operational state and characteristic response of the 87T-based protection scheme after inception of the fault. The horizontal bars indicate the state and duration of the;

- fault signal (FLTSIG)
- transfer status of the differential relay trip signal (87T)
- transfer status of the PCDAA trip signal (PMUT)
- local terminal circuit breaker (CB2) active state status
- remote terminal circuit breaker (CB3) active state status
- the circuit breaker lock active state status (86T)

The graphs represent respective status conditions for the duration of the fault, with a time window which ranges $(0 \leq t \leq +0.35)$ seconds.

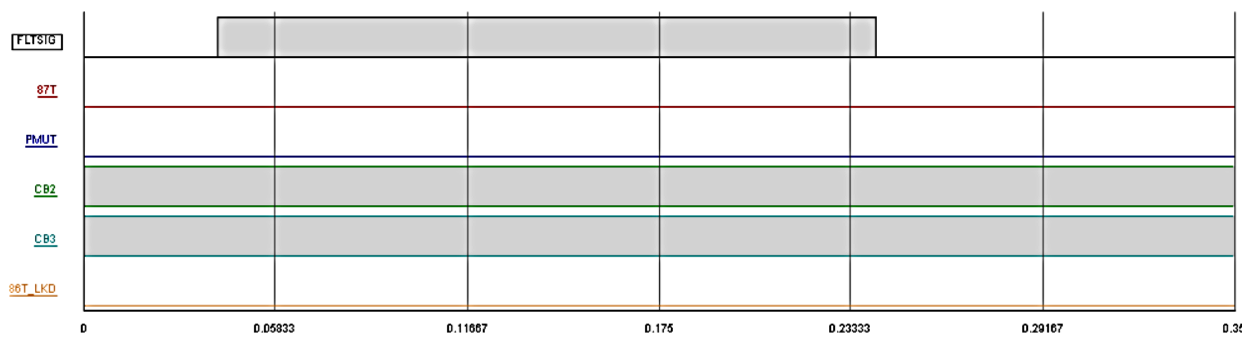


Figure 6.28: Fault signal status, Relay state, CT2 state and CT3 state

The graph characteristic description for Figure 6.28 is given in Table 6.16.

Table 6.16: RTEF case study simulation runtime protective characteristic results

Logical State		Time point	Description
FLTSIG	1	$+0.054 \leq t \leq +0.255$	The fault signal is incepted for 0.2 seconds
PMUT	0	-	The differential trip signal is not transferred to open the breaker.
87T	-	-	-
CB2	1	-	Circuit breaker 2 remains closed
CB3	1	-	Circuit breaker 3 remains closed
86T_LKD	0	-	The breakers remain unlocked

The results illustrated by the graphs highlight how the PMU-based protection scheme demonstrates characteristic response similar to the 87T-based protection scheme.

6.5.3 Discussion of case study results

For the outlined case study, control variables are maintained, and both the 87T-based protection scheme and the PMU-based protection scheme are implemented over parallel conditions. The experimental investigation centers on inception of an external fault RTEF, analyzing the real-time characteristic response of the transmission line system under protection from each scheme exclusively. For the case study, both the PMU-based protection scheme and the 87L demonstrate indistinguishable protection security characteristics, identifying the external fault and averting operation of the line terminal circuit breakers to establish selective protection and adequate line security. The PMU-based line differential protection scheme utilizes synchrophasors, and administers the developed PCDAA to employ the percentage differential scheme.

6.6 Case Study three:

This case study evaluates the operational characteristic and performance of the 87T-based protection scheme and the PMU-based protection scheme over the same transmission line system for the line to ground fault case illustrated in Figure 6.29.

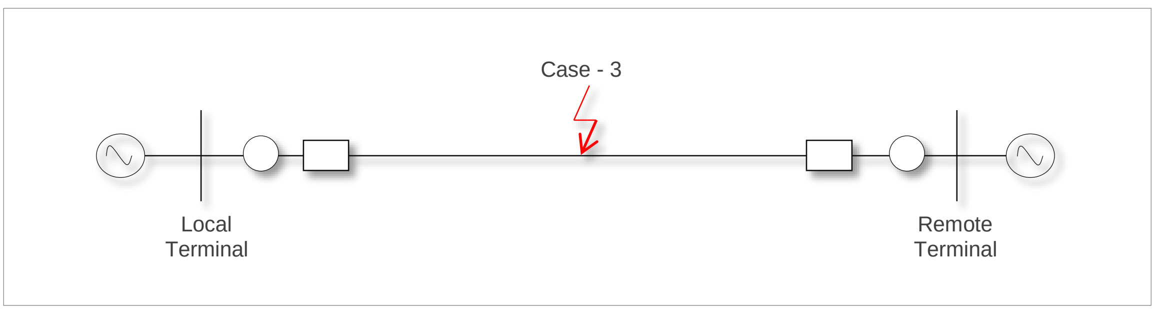


Figure 6.29: Internal fault case

RSCAD is used to implement and evaluate the response of the 87T-based protection scheme. The developed PCDAA software is used to implement and evaluate the characteristic operation of the proposed PMU-based protection scheme.

When the Internal Fault (IF) is applied, the response of the conventional 87T-based protection scheme is compared to the response of the PMU -based protection scheme. The correlation and comparative analysis of the obtained results is detailed in a discussion presented at the end of the case study.

Figure 6.30 presents the RSCAD runtime interface results obtained from real-time simulation of the IF case.

Figure 6.21 presents the runtime interface for the Internal Fault (IF) real-time simulation case.

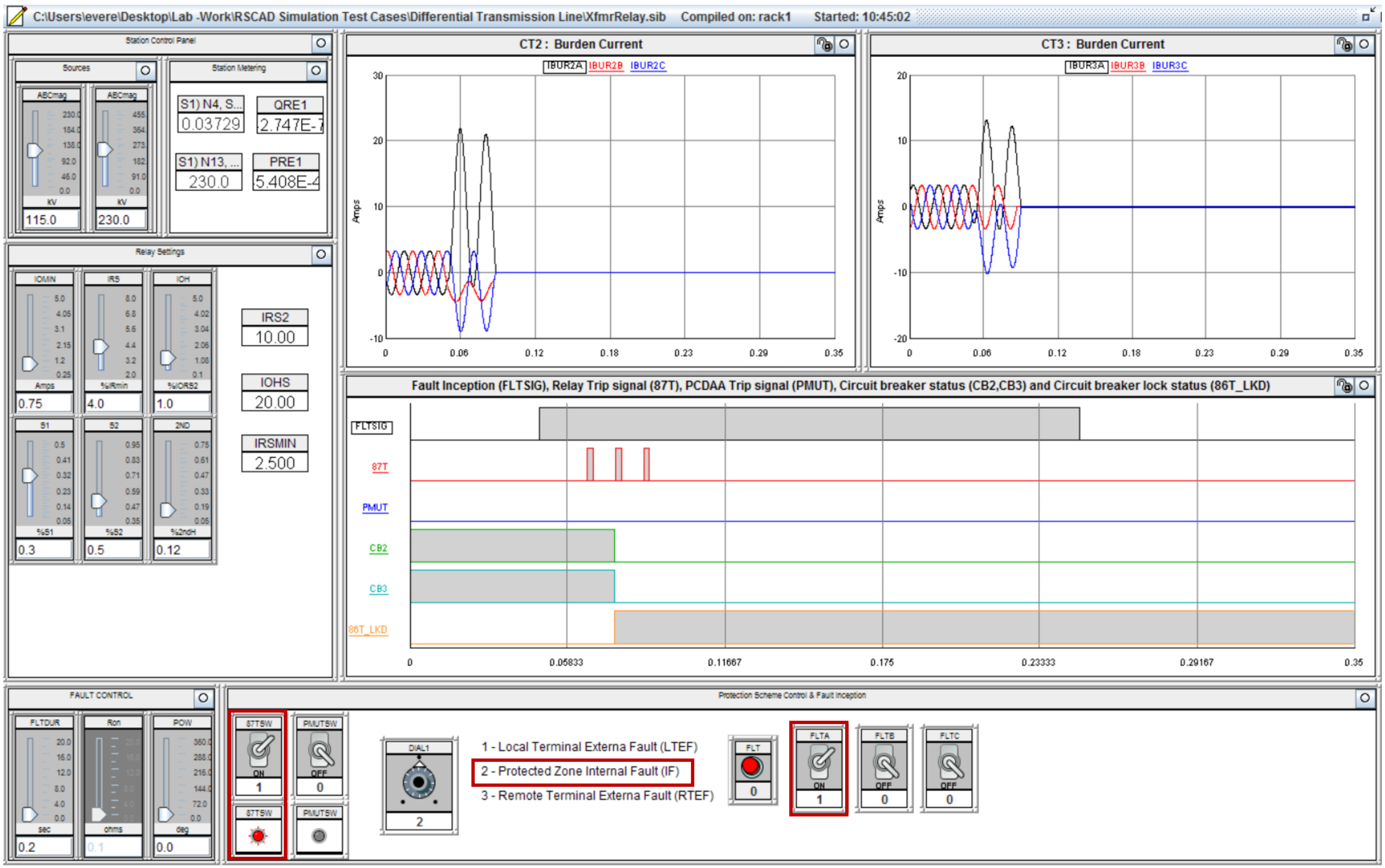


Figure 6.30: Real-time simulation results for the IF case implementing 87T-based differential protection

6.6.1 87T-based protection scheme implementation

When the IF is introduced into the real-time transmission line simulation, the response of the 87T-based differential protection scheme is exhibited in the runtime interface, as illustrated Figure 6.30. This Section presents and examines results assimilated from the runtime environment.

6.6.1.1 Current transformer burden current analysis

Figure 6.31 illustrates characteristic response for CT2 burden current. The graph presents the characteristic response of the CT2 burden current when the IF is introduced into the simulation of the transmission line system in real-time. The graph maps the behavior of the burden currents; IBUR2A, IBUR2B and IBUR2C for the duration; $(0 \leq t \leq +0.35)$ seconds, where t represents the independent time variable. CT2 burden current magnitudes vary in the range; $(-9.07 \leq A \leq +21.97)$ Amps, where A represents the dependent Ampere variable.

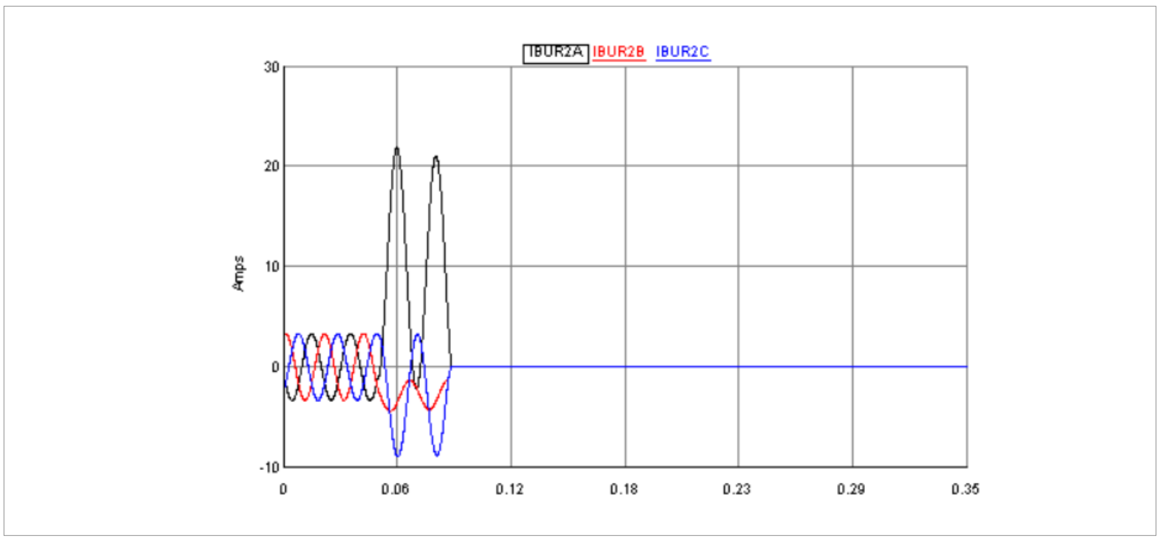


Figure 6.31: CT2 Burden Current

Figure 6.32 displays the behavior of the CT3 burden currents; IBUR3A, IBUR3B and IBUR3C for the duration; $(0 \leq t \leq +0.35)$ seconds. The CT3 burden current magnitudes vary in the range $(-10.27 \leq A \leq +13.05)$ Amps.

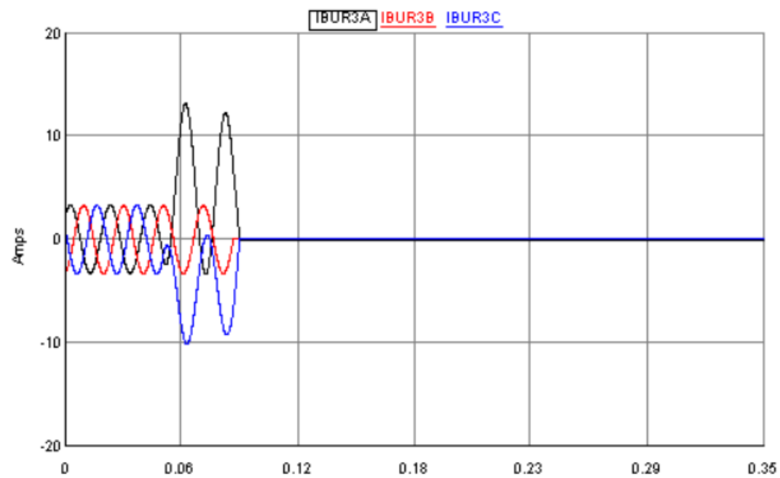


Figure 6.32: CT3 Burden Current

Description of the resulting current transformer burden currents is given in Table 6.17 and Table 6.18.

Table 6.17: CT2 Burden Current Characteristic

Time - Interval	**Description: CT2 Burden Current Characteristic Response**
$0 \leq t \leq +0.034$	Before the IF is introduced, IBUR2A, IBUR2B and IBUR2C repetitively oscillate for the range $(-3.28 \leq A \leq +3.28)$ Amps. This repetitive oscillation indicates system stability prior fault inception
$+0.034 \leq t \leq +0.074$	When the line to ground fault is applied to phase A in the IF case: • IBUR3A surges sharply, oscillating for the range; $(-1.81 \leq A \leq +21.97)$ Amps. • IBUR3B remains constant, oscillating for the range; $(-4.44 \leq A \leq -1.50)$ Amps. • IBUR3C decreases slightly, oscillating for the range; $(-9.07 \leq A \leq +3.28)$ Amps.
$+0.074 \leq t \leq +0.35$	IBUR3A, IBUR3A and IBUR3A current flow is interrupted and remains zero

Table 6.18: CT3 Burden Current Characteristic

Time - Interval	Description: CT3 Burden Current Characteristic Response
$0 \leq t \leq +0.034$	Before the IF is introduced, IBUR3A, IBUR3B and IBUR3C repetitively oscillate for the range $(-3.28 \leq A \leq +3.28)$ Amps. This repetitive oscillation indicates system stability prior fault inception
$+0.034 \leq t \leq +0.074$	When the line to ground fault is applied to phase A in the IF case: • IBUR3A surges sharply, oscillating for the range; $(-3.47 \leq A \leq +13.05)$ Amps. • IBUR3B remains constant, oscillating for the range; $(-3.47 \leq A \leq +3.17)$ Amps. • IBUR3C surges sharply, oscillating for the range; $(-10.12 \leq A \leq +0.39)$ Amps.
$+0.074 \leq t \leq +0.35$	IBUR3A, IBUR3A and IBUR3A current flow is interrupted and remains zero

When the line to ground fault is introduced in the IF case, the resultant oscillating characteristics indicate occurrence of an internal fault.

6.6.1.2 Direct trip transfer and circuit breaker state report

Figure 6.33 indicates the operational state and characteristic response of the 87T-based protection scheme after inception of the fault. The horizontal graphs indicate the state and duration of the;

- fault signal (FLTSIG)
- transfer status of the differential relay trip signal (87T)
- transfer status of the PCDAA trip signal (PMUT)
- local terminal circuit breaker (CB2) active state status
- remote terminal circuit breaker (CB3) state status
- the circuit breaker lock active state status (86T)

The graphs represent respective status conditions for the duration of the fault, with a time window which ranges $(0 \leq t \leq +0.35)$ seconds.

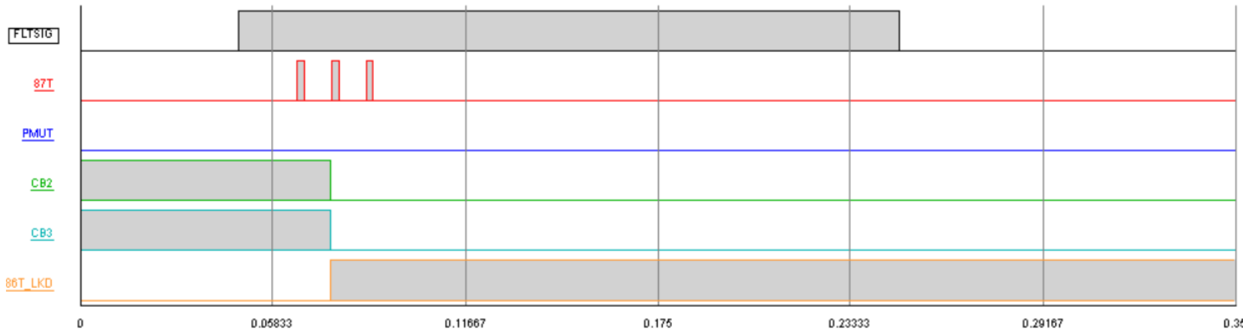


Figure 6.33: Fault signal status, Relay state, CT2 state and CT3 state

The graph characteristic descriptions are given in Table 6.19.

Table 6.19: IF case study simulation runtime protective characteristic results

Logical State		Time point	Description
FLTSIG	1	$+0.035 \leq t \leq +0.235$	The fault signal is incepted for 0.2 seconds
87T	0	$t = +0.053$	The differential trip signal is transferred to open the breaker.
PMUT	-	-	-
CB2	1	$t = +0.064$	Circuit breaker 2 is opened
CB3	1	$t = +0.064$	Circuit breaker 3 is opened
86T_LKD	0	$t = +0.064$	The breakers are locked in the open state

The restraint effect is disregarded and 87T high set trip condition is transferred at ($t = +0.053$). CB2 and CB3 are subsequently opened at ($t = +0.064$). The PMU-based protection scheme is not utilized and its trip signal (PMUT) is not considered.

6.6.2 PMU-based protection scheme implementation

In the second part of this real-time simulation case study, the PCDAA utility is administered to implement the PMU-based differential protection scheme. The secondary burden currents of CT2 and CT3 are represented by synchrophasors from PMU_1 and PMU_2 which are transmitted to the PCDAA where they are validated (re-aligned if necessary), and processes to evaluate the fault condition of the transmission line system.

Figure 6.34 presents instantaneous real-time synchrophasor data provided by the PCDAA utility software throughout the simulation study. The Figure displays synchrophasor value readings that correspond to; the no fault condition, inception of the IF and post fault analysis.

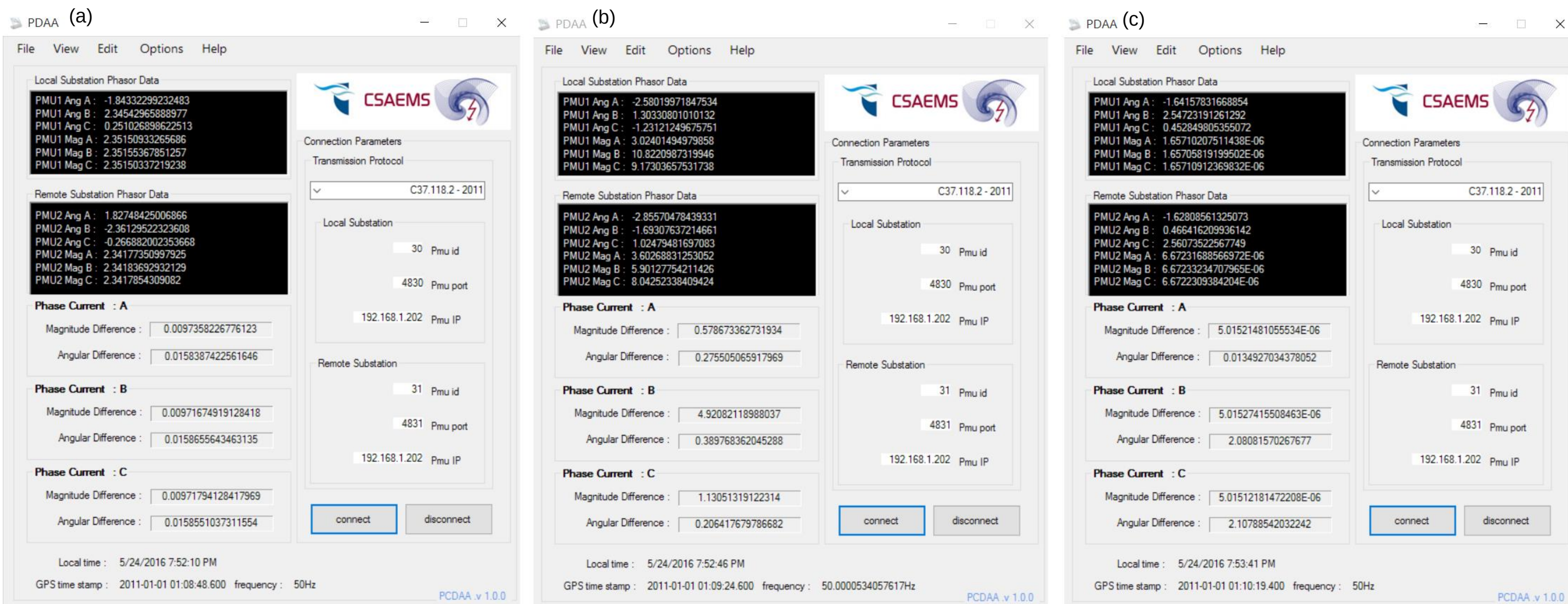


Figure 6.34: PCDAA synchrophasor data – IF case:

(a) no fault condition

(b) fault – inception state

(c) post fault state

In this simulation study the PCDAA software utility is used in conjunction with the RSCAD simulation runtime illustrated in Figure 6.35.

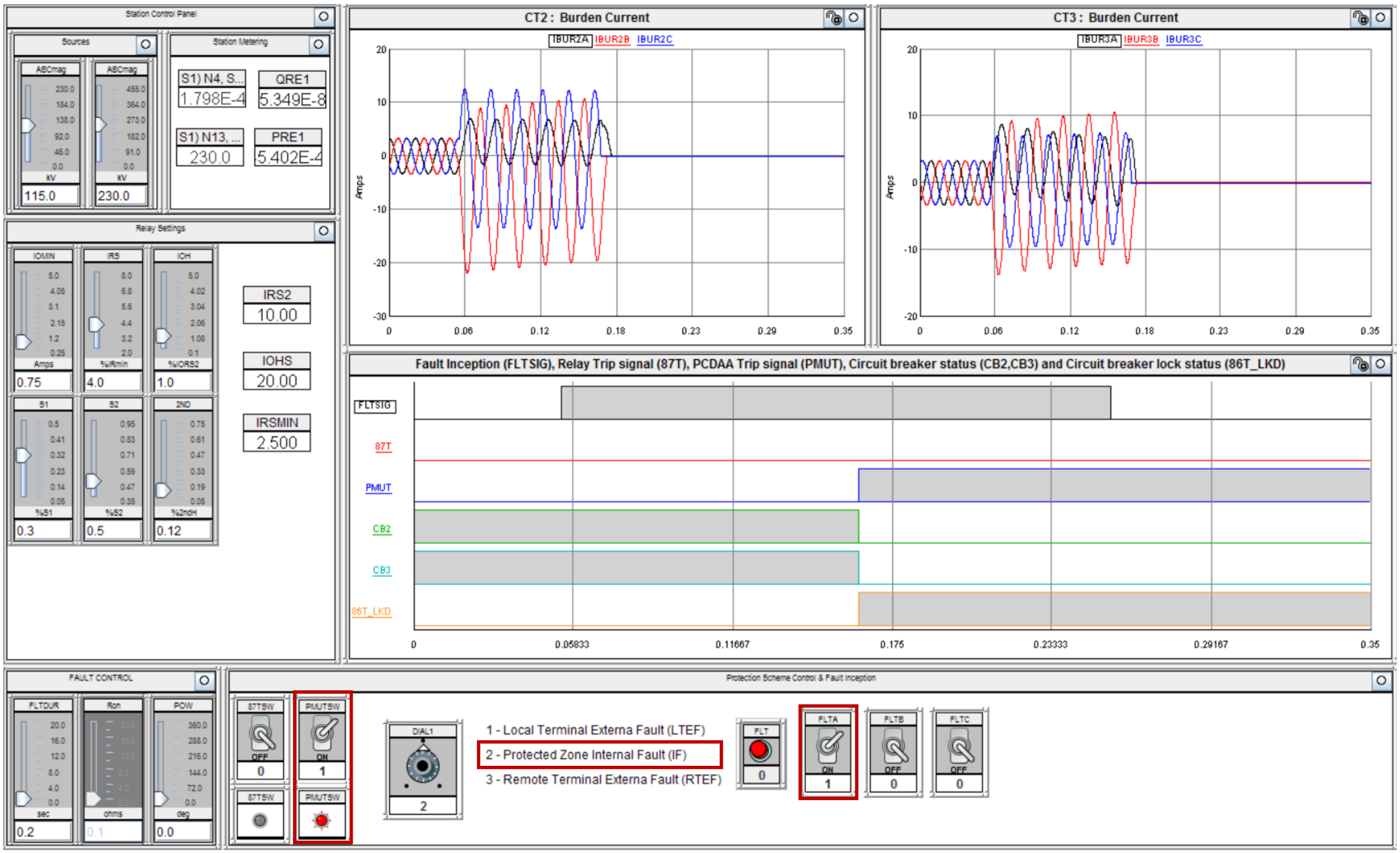


Figure 6.35: Real-time simulation results for the IF case implementing PMU-based differential protection

6.6.2.1 PCDAA transformer burden current

Table 6.10 presents real-time line terminal synchrophasor data.

Table 6.20: PMU burden current differential analysis

	No fault condition reading	fault condition reading	post-fault reading
PMU₁ : Phasor angular component (θ_1) Degrees			
Phase A	-1.843322992324	-2.580199718475	-1.641578316688
Phase B	2.345429658889	1.303308010101	2.547231912612
Phase C	0.251026898622	-1.231212496757	0.452849805355
PMU₁ : Phasor magnitude component ($\bar{\imath}_1$) Amps			
Phase A	2.351509332656	3.024014949798	1.657102075114 e^{-6}
Phase B	2.351553678512	10.822098731994	1.657058191995 e^{-6}
Phase C	2.351503372192	9.173036575317	1.657109123698 e^{-6}
PMU₂ : Phasor angular component (θ_2) Degrees			
Phase A	1.827484250068	-2.855704784393	-1628085613250
Phase B	-2.361295223236	-1.693076372146	0.466416209936
Phase C	-0.266882002353	1.024794816970	2.560735225677
PMU₂ : Phasor magnitude component ($\bar{\imath}_2$) Amps			
Phase A	2.341773509979	3.602688312530	6.672316885669 e^{-6}
Phase B	2.341836929321	5.901277542114	6.672332347079 e^{-6}
Phase C	2.341785430908	8.042523384094	6.672230938420 e^{-6}
(PMU₂ - PMU₁) : Magnitude Differential ($\bar{\imath}_3$) Amps			
Phase A	0.009735822677	0.578673362731	5.015214810555 e^{-6}
Phase B	0.009716749191	4.920821189880	5.015274155084 e^{-6}
Phase C	0.009717194128	1.305131912231	5.015121817422 e^{-6}
(PMU₂ - PMU₁) : Angle Differential (θ_3) Degrees			
Phase A	0.015838742256	0.275505065918	0.013492703437
Phase B	0.015865564346	0.389768362045	2.080815702667
Phase C	0.015855103731	0.206417679786	2.107885420322

Graphical presentation of the instantaneous synchrophasors obtained from the PCDAA are shown in Figure 6.36.

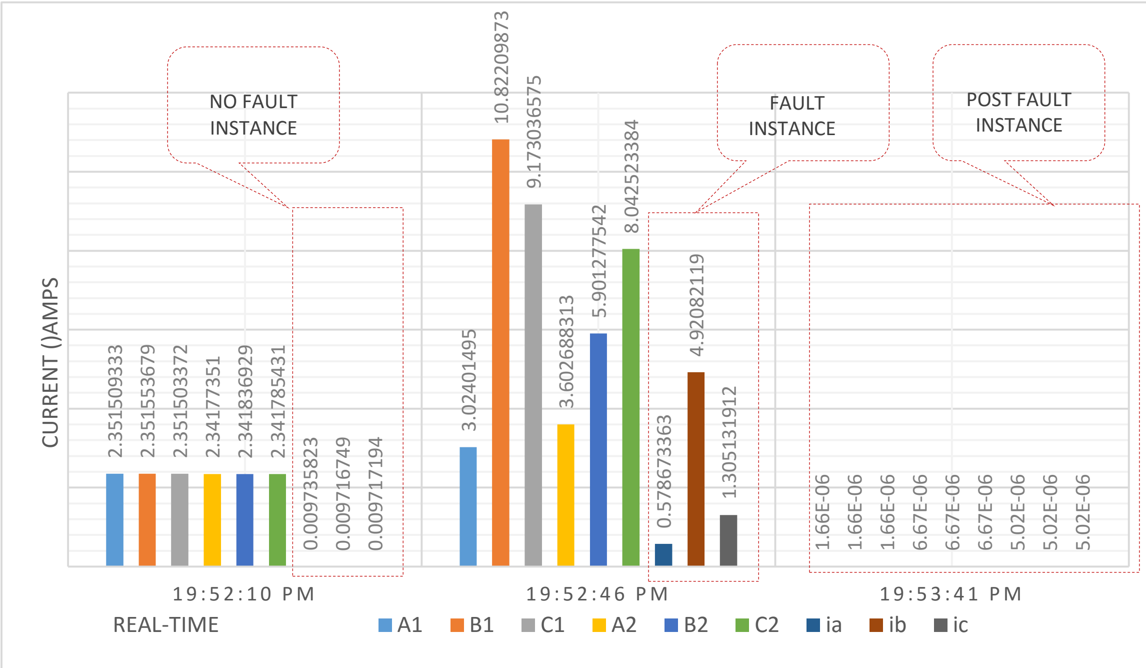


Figure 6.36: Analysis of PCDAA instantaneous synchrophasor results for the IF case

Where (A1-C1) represents PMU_1 synchrophasors, (A2-C2) represents PMU_2 synchrophasors and (ia –ic) represents the synchrophasor-based current differentials. Figure 6.36 emphasizes;

- No fault instance: near zero current differential
- Fault instance: differential current indicates internal fault
- Post fault instance: no current flow

The PMU-based protection scheme employs the PCDAA to implement synchrophasor based current differential protection. The PCDAA administers a percentage restrained characteristic algorithm which facilitates protection for the transmission line system. The PMU-based protection scheme exhibits protection response characteristics which are indistinguishable from the conventional 87T-based protection scheme.

6.6.2.2 Direct trip transfer and circuit breaker state report

Figure 6.37 indicates the operational state and characteristic response of the 87T-based protection scheme after inception of the fault. The horizontal bars indicate the state and duration of the;

- fault signal (FLTSIG)
- transfer status of the differential relay trip signal (87T)
- transfer status of the PCDAA trip signal (PMUT)
- local terminal circuit breaker (CB2) active state status
- remote terminal circuit breaker (CB3) active state status
- the circuit breaker lock active state status (86T)

The graphs represent respective status conditions for the duration of the fault, with a time window which ranges $(0 \leq t \leq +0.35)$ seconds.

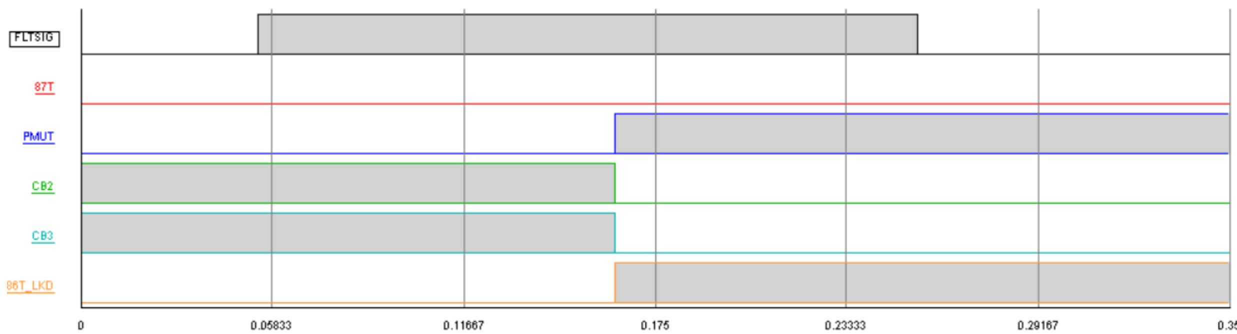


Figure 6.37: Fault signal status, Relay state, CT2 state and CT3 state

The graph characteristic description for Figure 6.37 is given in Table 6.21.

Table 6.21: IF case study simulation runtime protective characteristic results

Logical State		Time point	Description
FLTSIG	1	$+0.054 \leq t \leq +0.255$	The fault signal is incepted for 0.2 seconds
PMUT	0	$t = +0.163$	The differential trip signal is not transferred to open the breaker.
87T	-	-	-
CB2	1	$t = +0.163$	Circuit breaker 2 is opened
CB3	1	$t = +0.163$	Circuit breaker 3 is opened
86T_LKD	0	$t = +0.163$	The breakers are locked in the open state

The restraint effect is disregarded and 87T high set trip condition is transferred at $(t = +0.163)$. CB2 and CB2 are subsequently opened at $(t = +0.163)$. The 87T-based protection scheme is not utilized and its trip signal 87T is not considered.

6.6.3 Discussion of case study results

For the outlined case study, control variables are maintained, and both the 87T-based protection scheme and the PMU-based protection scheme are implemented over parallel conditions. The experimental investigation centers on inception of an internal fault IF, analyzing the real-time characteristic response of the transmission line system under protection from each scheme exclusively. For the case study, both the PMU-based protection scheme and the 87L demonstrate indistinguishable protection security characteristics, identifying the internal fault and operating the line terminal circuit breakers to establish decisive transmission line security. The PMU-based line differential protection scheme utilizes synchrophasors, and administers the developed PCDAA to employ the percentage differential scheme.

Figure 6.38 illustrates the difference in the trip transfer time of the protection schemes.

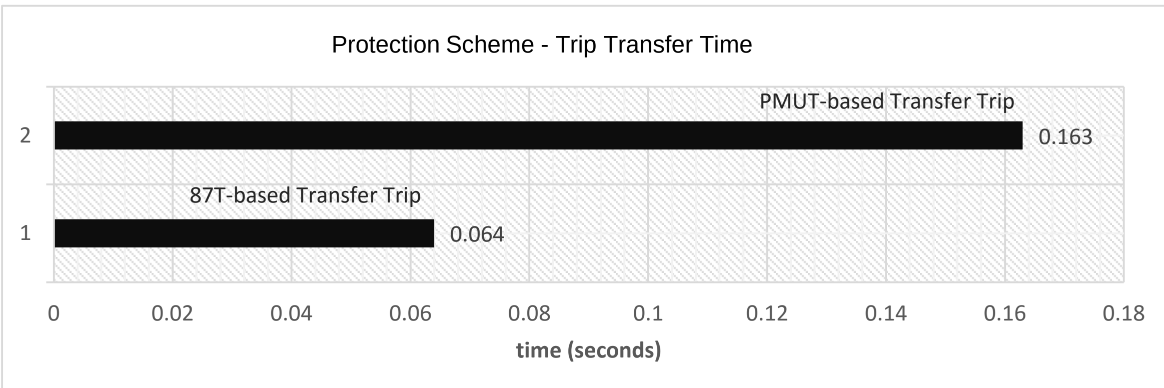


Figure 6.38: Trip time comparison between the PMU-based and 87T-based transmission line differential protection schemes

The internal fault case study indicates an increase in trip transfer time and circuit breaker operation, where the 87T-based protection scheme operates faster than the PMU-based protection scheme by $99.0\ ms$. The 87T-based differential protection scheme is based on software component models that do not require external hardware interface. This makes the 87T-based differential protection scheme highly ideal as opposed to the PMU-based protection scheme.

The PMU-based differential protection scheme makes use of two GTNET cards which employ PMU hardware components. The hardware GTIO communication interface between the GTNET card and the RTDS processor card contributes to the time delay in trip transfer and circuit breaker operation. The PMU-based protection scheme also makes use of an embedded system which employs serial communication to transfer trip logic from the control station to the RTDS which also contributes to the overall time delay in trip transfer time and circuit breaker operation. This delay can be improved through the implementation of IEC61850 Generic Object Oriented Substation Event (GOOSE) messaging which is be further explained in Chapter Seven.

6.7 Conclusion

This Chapter presents the real-time simulation study in which performance of the PMU-based transmission line differential protection scheme is comparatively correlated against a conventional differential protection scheme based on 87L digital relays. The performance of the conventional protection scheme is used to benchmark and provide a practical protection performance metric for the PMU-based protection scheme. This method enables practical comparative correlation between the two schemes and subsequent verification of the hypothesis.

The practical evaluation of PMU protection scheme validates the hypothesis of an alternate P-Class synchrophasor-based backup transmission line differential protection scheme solution. The next Chapter presents final conclusions, deliverables, future work recommendations and the Appendix.

Chapter 7 :
Conclusion and Recommendations

7.1 Introduction

Power system backup protection adds to system stability, supplementing fault identification and isolation in instances where main protection may operate inaccurately or completely fail. In this regard, conventional line protection schemes remain susceptible to disturbances, which necessitates backup protection with characteristics listed below.

- auxiliary protection utility
- simple system integration
- interchangeability and interoperability
- self-sufficient function impartial main protection

Conventional backup techniques are not easily implemented and rely too heavily on main protection. Most conventional backup protection models employ duplication techniques which involve backing up main protection relays with secondary relays from a different vendor. The clear objective of this method is based on the premise that relays from both the main and backup protection models are less likely to fail at the same instance. This method has proven inept as it increases probability for false tripping while adding to system complexity without adding any supplementary utility.

This research underlines accurate use of established concepts and proposes an innovative synchrophasor-based backup protection scheme as a unique solution that compensates for the shortcomings of conventional backup protection schemes. The research also provides a distinctive opportunity to meet the necessity for development of PMU-based applications.

Chapter Seven presents the research deliverables, detailing the drawn conclusions. The industrial and academic impacts of the research are listed, outlining foreseeable future work and recommendations related to the research and further development of PMU-based applications. This Chapter also lists the references and provides an Appendix.

7.1.1 Chapter organization

Section 7.2 presents thesis deliverables, correlating sub-problems and the problem statement to research findings.

Section 7.3 discusses academic benefits of the research.

Section 7.4 discusses impact of the research on industrial applications.

Section 7.5 recommends future work and further potential research and development.

7.2 Deliverables

The deliverables include a literature review that provides critical and in-depth revision of existing development and research in synchrophasor-based transmission line current differential protection. The established theoretic frame work is used as a guideline for the research process and engineering of the PMU-based transmission line differential protection scheme. The deliverables also include conclusive evaluation results. Preliminary simulation study presented in this Thesis verifies feasibility of applying synchrophasors directly for transmission line differential protection, validating the research hypothesis based on the prevailing C37.238-2011 Precision Time Protocol which identifies and acknowledges use of synchrophasors for protection.

The Thesis research deliverables are presented here, respectively correlating each sub-deliverable to the sub-problem listed in Chapter One.

7.2.1 Deliverable one

This research deliverable proposes an alternate scheme for transmission line differential back-up protection based on P-Class synchrophasors. Chapter Three presents the concept and feature analysis of the PMU-based protection scheme to actualize implementation of P-Class synchrophasors directly for line current differential protection. The scope of the research is limited to a two terminal transmission line, employing PMUs at respective line terminals. In the PMU-based differential protection scheme, the line terminal PMUs transmit synchrophasors to a remote control station which administers monitoring, protection, and control functions relative to the state of the transmission line.

In contrast with conventional relay-based differential protection schemes, the PMU-based protection scheme employs a remote control station to implement monitoring, control, and protection functions corresponding to the state of the protected zone. Significant benefits of the PMU based protection scheme include;

- A reduction in hardwire connection (no pilot channel needed to interconnect the PMUs)

- Line terminal PMUs already commissioned on the electric power grid can be utilized without any supplementary infrastructure
- Precise GPS synchronization between line terminal phasors

7.2.2 Deliverable two

A method and algorithm for design and implementation of the PMU-based transmission line differential protection scheme is presented in Chapter Three. The algorithm is developed from current differential principles that frame the conventional algorithm employed in contemporary digital differential relays.

7.2.3 Deliverable three

Protection software for implementing the PMU-based protection scheme is developed and engineered in Chapter Four. The control station employs the developed Phasor Current Differential Action Adapter (PCDAA) software utility to administer the differential characteristic protection and control algorithms. The non-proprietary independent software utility incorporates as part of its design;

- real-time concentration of real-time synchrophasor data streams based on guidelines specified by the C37.118-2011
- real-time application of a synchrophasor-based algorithm for evaluating differential current
- real-time implementation of the percentage restraint characteristic algorithm to employ adaptive current differential protection for the PMU-based protection scheme

It is worth mentioning that, protection and control consider a further stage in substation evolution with shift of the system control and protection assets from hardware standards to interchangeable and interoperable future proof software architecture-based standards. The PCDAA utility software is a deliverable that reinforces this paradigm shift. Engineering of the PCDAA delivers a software architecture deliverable which is illustrated in Chapter Fours' Figure 4.6. The architectural components of this frame work include;

- user access
- direct transfer tripping

- implementation of the developed synchrophasor-based current differential algorithm
- an SQL server / data historian
- a C37.118-2011-based protocol parser
- Multi parallel PMU data processing capability

7.2.4 Deliverable four

An embedded system used to administer and facilitate serial communication is presented as a research deliverable in Chapter Five. The Atmega 2560 microcontroller is utilized to transfer the differential trip logic, interfacing the developed PCDAA software to the RTDS simulation in real-time.

7.2.5 Deliverable five

Deliverable five provides a laboratory scale Test-bench for validating the concept of the PMU-based transmission line differential protection scheme. The Test-bench is designed, set up, and configured in the Centre for Substation Automation and Energy Management Systems (CSAEMS) laboratory. It facilitates industrial grade feasibility validation for the proposed PMU-based protection scheme, and enables establishment of conclusive and repeatable results. The Test bench in itself has a deliverable that includes the RSCAD simulation runtime. The simulation runtime model developed in RSCAD provides comprehensive and flexible control for the simulation case studies.

7.2.6 Deliverable six

The software developed in various environments are integrated within the Test-bench in order to implement the designed function of the synchrophasor-based transmission line differential protection scheme. Table 7.1 tabulates the software employed and describes the objectives established using the software.

Table 7.1: Software employed throughout the research

Developed Software	Development Environment	Functionality Established
Phasor Current Differential Action Adapter (PCDAA)	**Microsoft Visual Studio :**	• UDP based socket communication • Multi – PMU , parallel data processing • C37.118-2011 protocol parsing • Application of the developed synchrophasor-based current differential algorithm • Successful evaluation for all trip and no trip conditions
Simulation Runtime – PMU-based transmission line differential protection scheme evaluation	**RSCAD:**	• Flexible and comprehensive simulation control that facilitates pragmatic evaluation tests for the proposed PMU-based protection scheme
RTDS-PCDAA Serial Communication interface	**Arduino:**	• Serial communication interface between the developed PCDAA utility software and the RTDS real-time simulation

The engineered software is successfully implemented and utilized in the simulation and validation of the transmission line differential protection scheme based on P-Class synchrophasors.

7.2.7 Deliverable seven

The experimental investigation provides positive conclusive deductions that validates feasibility of the PMU-based protection scheme. The simulation study carried out emphasizes a pragmatic method for assessing operational characteristics of the proposed scheme. Throughout the simulation study, the PMU-based protection scheme exhibits operational characteristics similar to the conventional relay-based protection scheme.

7.3 Application to Academia

7.3.1 Revision of the electrical engineering curriculum

The research work carried out focuses on power system protection methods, integrating software and communication engineering principles. The combined aspects antiquate the engineering exclusivity notion, challenging researchers to be proficient in all fields and not constricted to a particular single stream of engineering (i.e. Heavy current, Communication or Programming).

The developed Test-bench provides a hardware platform where researchers and academics can understand the dynamics of application of P-Class synchrophasors in protection schemes.

7.3.2 Real-time laboratory scale, protection performance evaluation

Regarding feasibility tests associated with novel synchrophasor-based applications, the Real-Time Digital Simulation (RTDS) hardware provides a closed-loop testing environment. This simulation environment facilitates standard testing prior commission which provides:

- accurate digital modelling of the PMU component models
- comprehensive records of event sequence and oscillo-graphic signals
- programmable control facilities for external control of the test parameters

Specification of exact simulation parameters enables flexible and extensive evaluation of PMU applicability over different protection schemes for separate synchrophasor data transmission protocols as evident in the research.

7.4 Impact of a PMU-based protection scheme on the electric grid industry

Table 7.2 compares conventional duplication backup protection to the proposed PMU-based backup protection scheme.

Table 7.2: Comparison between duplication backup and the PMU-based backup protection scheme.

Description	Duplication back-up	PMU based protection
economy	- Not economic	- Economic
integration	- Demanding system integration	- Easy system integration
utility	- No additional utility	- Post disturbance analysis - Wide are monitoring

The proposed PMU-based differential protection scheme can be implemented over existing PMUs which have already been commissioned into the electric grid network. The developed PCDAA protection utility requires remote network identification of line terminal PMUs considered to implement the proposed synchrophasor-based transmission line differential back up protection scheme. An important consideration which greatly influences the simplicity of industrial application is implementing IEC 61850 based Generic Object-Orientated Substation Event (GOOSE) messages to facilitate circuit breaker control. The proposed PMU-based backup protection scheme conclusively supplements invaluable benefits.

7.5 Future work

7.5.1 Considerations

As mentioned in the previous Section, an essential consideration in the implementation of the PMU-based transmission line differential protection scheme is the application of GOOSE messages, replacing the preliminary serial communication interface used for proof of concept in the laboratory scale synchrophasor based current differential protection scheme.

7.5.2 Future Tests

Implementation of IEC 61850 GOOSE messages necessitates emphasis on communication latency tests, which form a critical part in further development of the proposed PMU-based differential protection scheme.

7.5.3 Future synchrophasor application

This research plays a role in an effort to broaden synchrophasor application considerations. The research is limited to transmission line differential protection, leaving room for further development on PMU-based protection applications.

It is worth mentioning that although the protection scheme is designed based on the IEC 61850 communication standard for electrical substation automation systems, only the RTDS simulator and RSCAD interface were utilized not substantiating interoperability. Establishing interoperability would be the first step towards future proofing the protection scheme in anticipation of the effects of future hardware and firmware.

7.5.4 Cyber Security

Smart grid cyber interactions continue to advance and evolve. The PCDAA developed in this research plays a role in a typical physical to cyber crossover. Integrating advanced computer and communication technologies into the smart grid introduces cyber vulnerabilities which contribute to the overall system stability and security. Power system security therefore extends, from solely considering the physical aspect, to the physical-cyber security domain.

This shift of focus towards cyber operations requires future work and establishment of research and development in cyber security and defense. Research in this regard is highly important as targeted attacks on vulnerable system components could potentially cripple the overall system. Future work needs to be carried out regarding cyber-physical interactions, analysing and modelling critical security for the overall smart grid (Mangiatordi, 2011).

7.6 Thesis Publication

Mthunzi .E, C. Y. Adewole, R. Tzoneva (2016) Algorithm for mitigating adverse synchrophasor measurement latency and packet losses in wide area power systems. (Sent to the IET international journal and currently under review).

References

Adamiak, M., Premerlani, W. & Kasztenny, B. 2006. Synchrophasors : Definition, Measurement, and Application.

Adamiak, M.G., Alexander, G.E. & Premerlani, D.W. 1998. A New Approach to Current Differential Protection for Transmission Lines. : pp.11 – 13.

Agile Modeling. 2016. Feature Driven Development (FDD) and Agile Modeling. http://www.agilemodeling.com/essays/fdd.htm 1 February 2016.

Albrecht, N.P., Fleck, W.C., Fodero, K.J. & Ince, R.J. 1992. Charge comparison protection of transmission lines--Communications concepts. *IEEE Transactions on Power Delivery*, 7(4): pp.1853–1860.

Ali, I., Hussain, S.M.S. & Aftab, A. 2015. Communication Modeling of Phasor Measurement Unit based on IEC 61850-90-5. : pp.1–6.

Altuve, H., Benmouyal, G., Roberts, J. & Tziowaras, D.A. 2004. Transmission line differential protection with an enhanced charecteristic. , 1988(1).

Altuve, H.J., Mooney, J.B. & Alexander, G.E. 2009. Advances in series-compensated line protection. *Protective Relay Engineers, 2009 62nd Annual Conference for*: pp.263–275. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4982518.

Andersson, G. 1994. DAMPING OF POWER SYSTEM OSCILLATIONS BY USE OF CONTROLLABLE COMPONENTS. , 9(4): 2046–2054.

Andersson, L., Brunner, C. & Engler, F. 2003. Substation automation based on IEC 61850 with new process-close technologies. *2003 IEEE Bologna PowerTech - Conference Proceedings*, 2: pp.271–276.

Apostolov, A., Member, S., Brunner, C. & Clinard, K. 2003. Use of IEC 61850 Object Models for Power System Quality / Security Data Exchange. *Distribution*: pp.155–164. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1259346.

Armenia, A. & Chow, J.H. 2010. A flexible phasor data concentrator design leveraging existing software technologies. *IEEE Transactions on Smart Grid*, 1(1): pp.73–81.

Arnold, T., Adewole, A.C. & Tzoneva, R. 2014. Performance Testing and Assessment of Multi-Vendor Protection Schemes Using Proprietary Protocols and the IEC 61850 Standard.

Bi, T., Liu, H., Zhang, D. & Yang, Q. 2012. The PMU dynamic performance evaluation and the comparison of PMU standards. *IEEE Power and Energy Society General Meeting*, 973.

Brunner, C. 2008. The impact of IEC 61850 on protection. *IET 9th International Conference on Developments in Power Systems Protection (DPSP 2008)*, 2008: pp.14–19. http://digital-library.theiet.org/content/conferences/10.1049/cp_20080003.

Buldyrev, S. V, Parshani, R., Paul, G., Stanley, H.E. & Havlin, S. 2010. Catastrophic cascade of failures in interdependent networks. *Nature*, 464(7291): 1025–8. http://havlin.biu.ac.il/Publications.php?keyword=Catastrophic+cascade+of+failures+in+interdependent+networks&year=*&match=all 28 December 2015.

Burrows, C. 2010. C# 4.0 - New C# Features in the .NET Framework 4. https://msdn.microsoft.com/en-us/magazine/ff796223.aspx 4 March 2016.

Castello, P., Ferrari, P., Flammini, A., Muscas, C., Pegoraro, P.A. & Rinaldi, S. 2015. Assessment of time synchronization quality in a distributed PMU. *Conference Record - IEEE Instrumentation and Measurement Technology Conference*, 2015-July: pp.1549–1554.

Chen, F., Han, X., Pan, Z. & Han, L. 2008. State Estimation Model and Algorithm Including PMU. , (April): pp.1097–1102.

Cheng, Y., Lu, C., Men, K. & Tu, L. 2014. Application of the complex algorithm in PMU-only state estimaiton. *POWERCON 2014 - 2014 International Conference on Power System Technology: Towards Green, Efficient and Smart Power System, Proceedings*, (Powercon): pp.349–354.

Cisco. 2016. Cisco SFP Optics For Gigabit Ethernet Applications. : 2016. http://www.cisco.com/en/US/prod/collateral/modules/ps5455/ps6577/product_data_sheet0900aecd8033f885.html 4 March 2016.

Clavel, F., Savary, E., Angays, P. & Vieux-Melchior, A. 2015. Integration of a new standard: A network simulator of IEC 61850 architectures for electrical substations. *IEEE Industry Applications Magazine*, 21(1): pp.41–48.

CodePlex. 2016. The Open Source Phasor Data Concentrator. http://openpdc.codeplex.com/ 2 March 2016.

Cohen, R. & Ofek, Y. 1995. Reliable transmission of data over a semi-FIFO routing layer. *Computer Networks and ISDN Systems*, 27(12): pp.1633–1649.

CompuSharp. 2012. Inter Control Center Protocol. http://www.compusharp.com/intercontrol.html 4 May 2016.

D. H. Hansson. 2010. Ruby on Rails. http://rubyonrails.org/ 5 May 2016.

Dalcastagnê, A.L., Filho, S.N., Zurn, H.H. & Seara, R. 2008. An iterative two-terminal fault-location method based on unsynchronized phasors. *IEEE Transactions on Power Delivery*, 23(4): pp.2318–2329.

Dambhare, S., Soman, S. a & Chandorkar, M.C. 2009. Adaptive current differential protection schemes for transmission-line protection. *IEEE Transactions on Power Delivery*, 24(4): pp.1832–1841.

Darwish, H.A., Taalab, S.A., Malik, S.E.S.A.O.P., Ieee, F., Ower, I.I.P., Elay, D.I.R. & For, P.R. 2009. Experimental Evaluation of Power Differential Relay for Transmission Line Protection. : pp.1–7.

Electric Light and Power. 2015. IEC 61850 Multivendor Substation Applications. http://www.elp.com/articles/powergrid_international/print/volume-16/issue-2/features/iec-61850-multivendor-substation-applications-whats-working-in-europe.html 17 April 2016.

Elgargouri, A. & Elmusrati, M. 2015. IEC 61850 Based Smart Grid Security. : pp. 2461–2465.

Eskom. 2007. The roles of voltage and frequency in the transmission of electricity. : pp.3–4.

Farantatos, E., Member, S., Stefopoulos, G.K., George, J., Member, S. & Meliopoulos, A.P. 2009. PMU-Based Dynamic State Estimation for Electric Power Systems.

Farrell, P. & Ong, H. 2000. Communication performance over a gigabit ethernet network. *Performance, Computing, and …*: pp.181–189. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=830317.

Gartia, A., Gartia, A., Gulati, A. & Kumar, C. 2013. Microcontroller Based Line Differential Protection Using Fiber Optic Communication. *2013 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia)*: pp.1–4. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6698741.

General Electric. 2007. Transmission Line Protection Principles. : pp.7–16. http://www.gedigitalenergy.com/smartgrid/Dec07/1-transmission.pdf.

Ghaisari, J., Bakhshai, A., Jain, P.K. & Bu, A. 2005. POWER OSCILLATIONS DAMPING BY MEANS OF THE SSSC : A MULTIVARIABLE CONTROL APPROACH. , (May): pp.2249–2252.

Goraj, M. 2010. Overview of IEC 61850. , 57(57): pp.1–40.

Grid Protection Alliance. 2014. Grid Protection Alliance. https://www.gridprotectionalliance.org/ 4 March 2016.

Grid Protection Alliance. 2013. Time-series Framework.

http://timeseriesframework.codeplex.com/ 4 March 2016.

Gwennap, L. 2016. Intel's P6 Uses Decoupled Superscalar Design. http://www.eecg.toronto.edu/~moshovos/ACA05/read/ppro1.pdf 13 March 2016.

Hanno Georg, Nils Dorsch, M.P. and C.W. 2013. Performance Evaluation of Time-critical Communication Networks for Smart Grids based on IEC 61850. : 6.

Hauer, J.F. 1996. Validation of phasor calculations in the macrodyne PMU for California-Oregon transmission project tests of march 1993. *IEEE Transactions on Power Delivery*, 11(3): pp.1224–1229.

Hongga, Z. 2007. A new state estimation model of utilizing PMU measurements. *2006 International Conference on Power System Technology, POWERCON2006*, 1(2): pp.1–5.

Hou, H., You, D., Yin, X., Xu, T., Jin, M., He, X., Wang, B. & Xie, Y. 2006. A Newly Developed Line Protection with Ethernet Communication Interface Based on 1EC61850. *2006 International Conference on Power System Technology*: pp.1–4. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4116087.

Hu, F. & Ji, L. 2008. GUI design: From information architecture to interactive experience. *9th International Conference on Computer-Aided Industrial Design and Conceptual Design: Multicultural Creation and Design - CAIDCD 2008*: pp.913–917.

Huang, Z., Hauer, J.F. & Martin, K.E. 2007. Evaluation of PMU dynamic performance in both Lab environments and under field operating conditions. *2007 IEEE Power Engineering Society General Meeting, PES*: pp.1–6.

Idaho Power. 2011. Transmission line Protection. https://www.eiseverywhere.com/file_uploads/aaf42a76a5588f69c7a1348d6f77fe0f_Introduction_to_System_Protection-_Protection_Basics.pdf 19 February 2016.

IEEE Power and Energy Society. 2011a. *IEEE Draft Standard Profile for Use of IEEE Std. 1588 Precision Time Protocol in Power System Applications*.

IEEE Power and Energy Society. 2012. *IEEE PC37.244 Draft Guide for Phasor Data Concentrator Requirements for Power System Protection , Control and Monitoring*.

IEEE Power and Energy Society. 2011b. *IEEE Standard for Synchrophasor Data Transfer for Power Systems*.

IEEE Power and Energy Society. 2011c. *IEEE Standard for Synchrophasor Measurements for Power Systems*.

IEEE Power Engineering Society. 2009. IEEE 61850 Standard for Power System Communication and Integration. , 2003.

IEEE Power Engineering Society. 2005. *IEEE C37.188 Standard for synchrophasors for power systems*. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=660853.

J. Skea, D. Anderson, T. Green, R. Gross, P.H. and M.L. 2007. Intermittent renewable generation and the cost of maintaining power system reliability. *Generation, Transmission & Distribution, IET*, 1(2): 324. http://link.aip.org/link/IGTDAW/v1/i2/p324/s1&Agg=doi.

Jain, A. 2009. P OWER S YSTEM T RACKING AND D YNAMIC.

JEDEC. 2012. DOUBLE DATA RATE (DDR) SDRAM STANDARD | JEDEC. http://www.jedec.org/standards-documents/docs/jesd-79f 13 March 2016.

Jiang, J.-A., Yang, J.-Z., Lin, Y.-H., Liu, C.-W. & Ma, J.-C. 2000. An Adaptive PMU Based Fault Detection/Location Technique for Transmission Lines Part I: Theory and Algorithms. *Ieee Transactions on Power Delivery*, 15(2): pp.486–493.

Jorge, C., Kumar, M. & Romero, J. 2006. Problems and Solutions for Line Differential Applications.

Kaminsky, D., Sima, J., Bilik, P. & Horinek, M. 2011. Flexible PMU tester based on virtual instrumentation. *11th International Conference on Electrical Power Quality and Utilisation*: pp.1–5. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6128942.

Kasztenny, B., Benmouyal, G., Altuve, H.J. & Fischer, N. 2011. Operating Characteristics of Microprocessor-Based Multiterminal Line Current Differential Relays. *Sel*: 1–30.

Kasztenny, B. & Finney, D. 2005. New algorithm for generator differential protection. *Energize*, (july): pp.52–55. http://www.ee.co.za/wp-content/uploads/legacy/New algorithm.pdf.

Kasztenny, B. & Fischer, N. 2010. Advanced line current differential protection with reduced channel requirements. *10th IET International Conference on Developments in Power System Protection (DPSP 2010). Managing the Change*: pp.94–94. http://digital-library.theiet.org/content/conferences/10.1049/cp.2010.0251.

Keyvani, B., Zadeh, M.K. & Lesani, H. 2014. A New Method for Mitigation of Power Oscillations With Fast Reclosing of Transmission Lines Based on SIME. : pp.1–5.

Khan, M., Li, M., Ashton, P., Taylor, G. & Liu, J. 2014. Big data analytics on PMU measurements. *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2014*: pp.715–719.

Khavnekar, A., Wagh, S. & More, A. 2015. Comparative Analysis of IEC 61850 Edition-I and II Standards for Substation Automation.

Khederzadeh, M. 2010a. Application of synchrophasors to adaptive protection of transmission lines compensated by Uni Power Flow Control. *10th IET International Conference on Developments in Power System Protection (DPSP 2010). Managing the Change*: pp.112–112. http://digital-library.theiet.org/content/conferences/10.1049/cp.2010.0256.

Khederzadeh, M. 2010b. Application of synchrophasors to adaptive protection of transmission lines compensated by UPFC. *10th IET International Conference on Developments in Power System Protection (DPSP 2010). Managing the Change*: 112–112. http://digital-library.theiet.org/content/conferences/10.1049/cp.2010.0256.

Kolosok, I., Korkina, E. & Buchinsky, E. 2014. The Test Equation Method for Linear State Estimation Based on PMU Data. , (Project 01201369131).

Komarnicki, P., Dzienis, C., Styczynski, Z. a., Blumschein, J. & Centeno, V. 2008. Practical experience with PMU system testing and calibration requirements. *IEEE Power and Energy Society 2008 General Meeting: Conversion and Delivery of Electrical Energy in the 21st Century, PES*: pp.1–5.

Korba, P. 2008. Monitoring Amplitude , Frequency and Damping of Power System Oscillations with PMU Measurements H : : pp. 1–7.

Kosterev, D.N., Esztergalyos, J. & Stigers, C.A. 1998. Feasibility study of using synchronized phasor measurements for generator dropping controls in the colstrip system. *IEEE Transactions on Power Systems*, 13(3): pp.755–761.

Kosterev, D.N., Taylor, C.W. & Mittelstadt, W.A. 1999. Model validation for the August 10, 1996 WSCC system outage. *IEEE Transactions on Power Systems*, 14(3): pp.967–979. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=780909.

Kostic, T., Preiss, O. & Frei, C. 2005. Understanding and using the IEC 61850: a case for meta-modelling. *Computer Standards & Interfaces*, 27(6): pp.679–695. http://dl.acm.org/citation.cfm?id=1648864.1648944 15 February 2016.

Lam, H. & Thuan, T. 2008. .NET Framework. http://etutorials.org/Programming/Programming+C.Sharp/Part+I+The+C+Language/Chapter+1.+C+and+the+.NET+Framework/1.2+The+.NET+Framework/ 2 May 2016.

Lathi, B.P. 2004. *Signal Processing and Linear Systems*.

Le, N.M., Guatto, A., Cohen, A. & Pop, A. 2013. Correct and efficient bounded FIFO queues. *Proceedings - Symposium on Computer Architecture and High Performance Computing*: pp.144–151.

Lee, J.-D., Lee, S.-J., Bae, J.-H. & Kwon, D.-Y. 2013. The PMU interface using IEC 61850. : pp.1125–1128. http://bibliotecavirtual.ups.edu.ec:2065/ielx7/6661630/6675287/06675573.pdf?tp=&arnumber=6675573&isnumber=6675287.

Lei, H., Singh, C. & Sprintson, A. 2014. Reliability Modeling and Analysis of IEC 61850 Based Substation Protection Systems. *IEEE Transactions on Smart Grid*, 5(5): pp.1–9. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6868302.

Li, H., Zhang, H. & Peng, D. 2011. Research and Design of Industrial Ethernet Communication Gateway on Power Station. : pp.986–989.

Li, H.Y., Southern, E.P. & Crossley, P.A. 1997. A new Type of Differential Feeder Protection Relay using Global Positioning System for Data Synchronization. , 12(3): pp.1090–1099.

Liang, X., Zhou, J., Yang, Y., Shen, J., Xie, L. & Zhou, B. 2008. Development of new kind of merging unit based on Iec61850. *2008 China International Conference on Electricity Distribution, CICED 2008*.

Lin, Y.H., Liu, C.W. & Chen, C.S. 2004. A new PMU-based fault detection/location technique for transmission lines with consideration of arcing fault discrimination - Part I: Theory and algorithms. *IEEE Transactions on Power Delivery*, 19(4): pp.1587–1593.

Lin, Y.-H., Liu, C.-W. & Chen, C.-S. 2004. A New PMU-Based Fault Detection/Location Technique for Transmission Lines With Consideration of Arcing Fault Discrimination—Part II: Performance Evaluation. *IEEE Transactions on Power Delivery*, 19(4): pp.1594–1601. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1339320.

Lira, R., Mycock, C., Wilson, D. & Kang, H. 2011. PMU performance requirements and validation for closed loop applications. *IEEE PES Innovative Smart Grid Technologies Conference Europe*: pp.1–7.

Liu, H., Bi, T. & Yang, Q. 2012. The PMU performance evaluation. *CPEM Digest (Conference on Precision Electromagnetic Measurements)*, 0013(1): pp.416–417.

Liu, Y., Gao, H., Gao, W., Li, N. & Xiang, M. 2011. A design scheme of line current differential protection based on IEC61850. *2011 IEEE Power Engineering and Automation Conference*, 2: pp.520–523.

Liu, Y., Wei, H. & Zhao, T. 2012. Ethernet Communication Based on the IEC61850-line Monitoring of Electrical Equipment IED. , (September): pp.778–781.

Lopez, O., Member, S. & Guardado, J.L. 2015. Modeling Transformer Differential

Protection.

Mackiewicz, R. 2004. Technical Overview and Benefits of the IEC 61850 Standard for Substation Automation. : pp.1–8.

Madani, V., Picard, S., Yin, Y. & Adamiak, M. 2015. Challenges and lessons learned from commissioning an IEC 61850-90-5 based synchrophasor system. *2015 68th Annual Conference for Protective Relay Engineers, CPRE 2015*: pp.842–849.

Mahanty, R. & Gupta, P. 2004. Application of RBF neural network to fault classification and location in transmission lines. *IEE Proceedings-Generation, Transmission and …*, 151(3): pp.201–212. http://digital-library.theiet.org/content/journals/10.1049/ip-gtd_20040098.

Mangiatordi, F. 2011. Smart Grid Cyber Security Requirements. *Technology*: 1–4.

Martin, K.E. 2006. Phasor Measurement Systems in the WECC. *2006 IEEE PES Power Systems Conference and Exposition*: pp.132–138.

Miao, S., Liu, P. & Lin, X. 2010. An adaptive operating characteristic to improve the operation stability of percentage differential protection. *IEEE Transactions on Power Delivery*, 25(3): pp.1410–1417.

Microsoft. 2015. Visual Studio - Microsoft Developer Tools. https://www.visualstudio.com/ 1 February 2016.

Miller, H., Burger, J., Fischer, N. & Kasztenny, B. 2010a. Modern line current differential protection solutions. *2010 63rd Annual Conference for Protective Relay Engineers*.

Miller, H., Burger, J., Fischer, N. & Kasztenny, B. 2010b. Modern line current differential protection solutions. *2010 63rd Annual Conference for Protective Relay Engineers*, 2(3).

Mills, D. 2014. Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI. http://tools.ietf.org/html/rfc2030 12 March 2016.

Moraes, R.M., Hu, Y., Stenbakken, G., Martin, K., Alves, J.E.R., Phadke, A.G., Volskis, H.A.R. & Centeno, V. 2012. PMU interoperability, steady-state and dynamic performance tests. *IEEE Transactions on Smart Grid*, 3(4): pp.1660–1669.

Morais Pereira, C.E.M. & Zanetta Jr., L.C. 2005. Optimization Algorithm for Fault Location in Transmission Lines Considering Current Transformers Saturation. *IEEE Transactions on Power Delivery*, 20(2): pp.603–608. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1413292.

Moxa. 2004. PT-G7509 Series. , 1613: pp.1–3.

Nengling, T., Deng, Y. & Chen, C. 2006. New generator split-phase transverse differential protection based on wavelet transform. *IEEE Transactions on Power Delivery*, 21(4): pp.1817–1823.

Nexans, S.A. & Nerc, E.A. 2012. R e a l - T i m e A p p l i c a t i o n o f S y n c h r o p h a s o r s f o r I m p r o v i n g R e l i a b i l i t y.

Park, C.W. 2010. Advanced Frequency Estimation Technique using Gain Compensation. , 866025404(2).

Pires, V.F. & Guerreiro, M. 2008. A current differential line protection using a synchronous reference frame approach. *IET 9th International Conference on Developments in Power Systems Protection (DPSP 2008)*, 2008: pp.198–203. http://digital-library.theiet.org/content/conferences/10.1049/cp_20080035.

Pires, V.F., Guerreiro, M., Fortunato, C. & Martins, L.S. 2010. Transmission lines protection based on the current eigenvalues differential concept. *10th IET International Conference on Developments in Power System Protection (DPSP 2010). Managing the Change*: P02–P02. http://digital-library.theiet.org/content/conferences/10.1049/cp.2010.0277.

Querol, X., Ayora, C., Ferna, C. & Janssen-jurkovicova, M. 2001. REAL-TIME APPLICATIONS OF PHASOR MEASUREMENT UNITS (PMU) FOR VISUALIZATION, REACTIVE POWER MONITORING AND VOLTAGE STABILITY PROTECTION. *Jornal of Enviromental Engineering*, 127(11): pp.994–1002.

Reilly, E.D. 2004. *Concise Encyclopedia of Computer Science*. John Wiley & Sons. https://books.google.com/books?id=5Jaa1BVverIC&pgis=1 16 February 2016.

Richards, G.G. & Tan, O.T. 1983. Fault location for transmission lines with current-transformer saturation. *IEE Proceedings C Generation, Transmission and Distribution*, 130(1): 22.

Rintamaki, O. & Ylinen, J. 2008. Communicating line differential protection for urban distribution networks. *2008 China International Conference on Electricity Distribution, CICED 2008*: pp.1–5.

RTDS-Technologies. 2013. RTDS-Technologies. https://www.rtds.com/wp-content/uploads/2014/09/RTDS-Technologies-Inc.-English.pdf 13 March 2016.

dos Santos, a., Martins, J.F., Monteiro, P., Kojovic, L. j. A., Bishop, M.T. & Day, T.R. 2010. Line differential protection systems based on low power stand alone current sensors. *10th IET International Conference on Developments in Power System Protection (DPSP 2010). Managing the Change*: pp.121–121. http://digital-library.theiet.org/content/conferences/10.1049/cp.2010.0259.

Sathaye, S., Ramakrishnan, K.K., Yang, H., Street, K. & Lkg-, A. 1994. FIFO Design for a High-speed Network Interface. : pp.2–11.

Singh, B., Sharma, N., Tiwari, A., Verma, K. & Singh, S. 2011. Applications of phasor measurement units (PMUs) in electric power system networks incorporated with FACTS controllers. *International Journal of Engineering, Science and Technology*, 3(3): pp.64–82.

Sivanagaraju, G., Chakrabarti, S. & Srivastava, S.C. 2014. Uncertainty in Transmission Line Parameters: Estimation and Impact on Line Current Differential Protection. *IEEE Transactions on Instrumentation and Measurement*, 63(6): pp.1496–1504. http://bibliotecavirtual.ups.edu.ec:2065/ielx7/19/6812145/06680707.pdf?tp=&arnumber=6680707&isnumber=6812145\nhttp://bibliotecavirtual.ups.edu.ec:2065/xpl/articleDetails.jsp?tp=&arnumber=6680707&queryText=Uncertainty+in+transmission+line+parameters.

Slutsker, I. & Mokhtari, S. 1995. Implementation of phasor measurements in state estimator at Sevillana de Electricidad. *Power Industry Computer Application Conference*: pp.392–398. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=515269.

Tholomier, D., Richards, S. & Apostolov, A. 2008. Which One Is Better - Line Differential or Directional Comparison? : pp.86–91. http://ieeexplore.ieee.org/ielx5/4496939/4496940/04496965.pdf?tp=&arnumber=4496965&isnumber=4496940.

Unde, S. V & Dambhare, S.S. 2011. GPS synchronized current differential protection of mutually coupled line. *2011 International Conference on Power and Energy Systems*: pp.1–6.

Voloh, I., Johnson, R., Multilin, G.E. & Canada, O. 2005. Over Pilot Wires. : 287–290.

Walter, S. & Kenrick, M. 2005. Absolute C++. http://is.ptithcm.edu.vn/~tdhuy/Programming/AbsoluteC++.pdf 16 February 2016.

Wang, Y. 2010. Research of time synchronization in digital substation based on IEEE 1588. *The 2nd International Conference on Information Science and Engineering*, (1): pp.2320–2325. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5691671.

Wheatley, J.M. 1985. MICROPROCESSOR BASED CURRENT DIFFERENTIAL PROTECTION. : pp.116–120.

Wood, A.J., Wollenberg, B.F. & B, S.G. 2013. *POWER GENERATION, OPERATION, AND CONTROL*. 3rd ed.

Wu, H. & Giri, J. 2006. PMU Impact on State Estimation Reliability for Improved Grid Security. *Transmission and Distribution Conference and Exhibition, 2005/2006 IEEE PES*: pp.1349–1351.

Wu, H., Tsakalis, K.S. & Heydt, G.T. 2004. Evaluation of Time Delay Effects to Wide-Area Power System Stabilizer Design. *IEEE Transactions on Power Systems*, 19(4): pp.1935–1941.

Xie, L., Choi, D., Member, S., Kar, S. & Poor, H.V. 2012. Wide-Area Monitoring Systems. , 3(3): pp.1154–1169.

Xingguo Wang, Zexin Zhou & Huanzhang Liu. 2015. A transmission line current differential protection based on virtual restraint current. *2015 IEEE Power & Energy Society General Meeting*: pp.1–4. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7285769.

Xu, Z.Y., Du, Z.Q., Ran, L., Wu, Y.K., Yang, Q.X., He, J.L. & Member, S. 2007. A Current Differential Relay for a 1000-kV UHV Transmission Line. , 22(3): pp.1392–1399.

Xue, Y., Kasztenny, B., Taylor, D. & Xia, Y. 2013. Series compensation, power swings, and inverter-based sources and their impact on line current differential protection. *2013 66th Annual Conference for Protective Relay Engineers*, (Section III): pp.80–91. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6822029.

Yang, C.-W. & Vyatkin, V. 2015. Modelling of IEC 61850 Message Passing for Automatic Generation of Distributed Control. : pp.1524–1529.

Yoo, Y., Lee, I., Choi, J.K. & Member, S. 2016. Experimental Performance Analysis of DER Network Communication in IEC 61850 Server. : pp.540–541.

Yuan, R., Shen, Y., Xiao, Z., Liu, W., Li, T. & Deng, X. 2015. Digital differential protection technique of transmission line using instantaneous active current: theory, simulation and experiment. *IET Generation, Transmission & Distribution*, 9(11): pp.996–1005. http://digital-library.theiet.org/content/journals/10.1049/iet-gtd.2014.0317.

Zhang, H., He, J., Li, B. & Bo, Z. 2009. An improved transformer percentage restraint method based on transmission line information. *1st International Conference on Sustainable Power Generation and Supply, SUPERGEN '09*: pp.1–4.

Zhang, J., Xue, A., Bi, T. & Member, S. PERFORMANCE ANALYSIS OF MNMR STATE ESTIMATOR CONSIDERING PMU DATA.

Zhang, J., Yuan, Z. & Member, Z.G. 2008. Study of the Entire Current Differential Protection of EHV transmission lines Based on Electronic Transducer. : pp.1–5.

Zhang, N., Dong, X.Z., Bo, Z.Q. & Klimek, A. 2008. An integrated relay for differential protection of transmission lines. *3rd International Conference on Deregulation and Restructuring and Power Technologies, DRPT 2008*, (April): pp.1787–1791.

Appendix A:

PCDAA, Software utility source code

Appendix A presents the source code for the developed PCDAA software utility. The appendix is sub-divided into five sections which present the component classes used to develop the synchrophasor-based protection application.

- A.1 : Main.cs
- A.2 : MainDesigner.cs
- A.3 : PCDAA.cs
- A.4 : Concentrator.cs
- A.5 : myData.cs

A.1 Class: Main.cs

```
using System;
using System.Linq;
using System.Windows.Forms;
using System.Threading.Tasks;
using System.Collections.Generic;

namespace pmuLive
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new pmuLive());
        }
    }}
```

A2 Class: MainDesigner.cs

```
using GSF;
using GSF.IO;
using System;
using System.IO;
using GSF.Units;
using System.Data;
using GSF.Units.EE;
using GSF.TimeSeries;
using System.Threading;
using GSF.PhasorProtocols;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.ComponentModel;
using System.Collections.Generic;
using System.Collections.Concurrent;
using GSF.PhasorProtocols.Anonymous;

using System.Threading.Tasks;

namespace pmuLive
{
    partial class pmuLive
    {
        #region Windows Form

        public void stopParse()
        {
            clock.Stop();

            sysDisplay.Clear();

            progressBar1.Value = 0;

            sysStatus.Text = "system : offline";

            MessageBox.Show("Connection Terminated", "Message", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        }

        private void progressBar()
        {
            progressBar1.Increment(4);
            if (progressBar1.Value == 100)
            {
                sysStatus.Text = "system : online";
            }
            else
            {
                sysStatus.Text = progressBar1.Value.ToString() + " %";
            }
        }

private void showUserDefinedConnectionParameters()
        {
            sysDisplay.AppendText(Environment.NewLine);
            sysDisplay.AppendText("local Pmu id             : " + locpmuid.Text);
            sysDisplay.AppendText(Environment.NewLine);
            sysDisplay.AppendText("local Pmu port          : " + locpmuport.Text);
            sysDisplay.AppendText(Environment.NewLine);
            sysDisplay.AppendText("local Pmu Server ip : " + locpmuip.Text);
            sysDisplay.AppendText(Environment.NewLine);
            sysDisplay.AppendText(Environment.NewLine);
            sysDisplay.AppendText("remote Pmu id            : " + rempmuid.Text);
            sysDisplay.AppendText(Environment.NewLine);
            sysDisplay.AppendText("remote Pmu port          : " + rempmuport.Text);
            sysDisplay.AppendText(Environment.NewLine);
```

```
            sysDisplay.AppendText("remote Pmu Server ip : " + rempmuip.Text);
            sysDisplay.AppendText(Environment.NewLine);
            sysDisplay.AppendText(Environment.NewLine);
            sysDisplay.AppendText("transport Protocol   : Tcp");
            sysDisplay.AppendText(Environment.NewLine);
            sysDisplay.AppendText("phasor Protocol       : IEEEC37 118.1 (2011)");
        }

        DifferentialCurrentAlgorithm computeCurrentDifferential = new
DifferentialCurrentAlgorithm();

        private void startThreadManager(string localID, string localConnString, string remoteID,
string remoteConnString)
        {
            Thread localPmuThread = new Thread(() => localPmuPhasorDataStream(localID,
localConnString));
            Thread remotePmuThread = new Thread(() => remotePmuPhasorDataStream(remoteID,
remoteConnString));
            localPmuThread.Start();
            remotePmuThread.Start();
        }


        #region form ui Components

        #region Components

        private void InitializeComponent()
        {
            this.components = new System.ComponentModel.Container();
            System.Windows.Forms.DataVisualization.Charting.ChartArea chartArea4 = new
System.Windows.Forms.DataVisualization.Charting.ChartArea();
            System.Windows.Forms.DataVisualization.Charting.Legend legend4 = new
System.Windows.Forms.DataVisualization.Charting.Legend();
            System.Windows.Forms.DataVisualization.Charting.Series series7 = new
System.Windows.Forms.DataVisualization.Charting.Series();
            System.Windows.Forms.DataVisualization.Charting.Series series8 = new
System.Windows.Forms.DataVisualization.Charting.Series();
            System.Windows.Forms.DataVisualization.Charting.ChartArea chartArea5 = new
System.Windows.Forms.DataVisualization.Charting.ChartArea();
            System.Windows.Forms.DataVisualization.Charting.Legend legend5 = new
System.Windows.Forms.DataVisualization.Charting.Legend();
            System.Windows.Forms.DataVisualization.Charting.Series series9 = new
System.Windows.Forms.DataVisualization.Charting.Series();
            System.Windows.Forms.DataVisualization.Charting.Series series10 = new
System.Windows.Forms.DataVisualization.Charting.Series();
            System.Windows.Forms.DataVisualization.Charting.ChartArea chartArea6 = new
System.Windows.Forms.DataVisualization.Charting.ChartArea();
            System.Windows.Forms.DataVisualization.Charting.Legend legend6 = new
System.Windows.Forms.DataVisualization.Charting.Legend();
            System.Windows.Forms.DataVisualization.Charting.Series series11 = new
System.Windows.Forms.DataVisualization.Charting.Series();
            System.Windows.Forms.DataVisualization.Charting.Series series12 = new
System.Windows.Forms.DataVisualization.Charting.Series();

            this.ConSet = new System.Windows.Forms.GroupBox();
            this.sysDisplay = new System.Windows.Forms.TextBox();
            this.comboBox1 = new System.Windows.Forms.ComboBox();
            this.radioButton3 = new System.Windows.Forms.RadioButton();
            this.radioButton2 = new System.Windows.Forms.RadioButton();
            this.radioButton1 = new System.Windows.Forms.RadioButton();
            this.button2 = new System.Windows.Forms.Button();
            this.button1 = new System.Windows.Forms.Button();
            this.tabControl2 = new System.Windows.Forms.TabControl();
            this.tabPage4 = new System.Windows.Forms.TabPage();
            this.locpmuport = new System.Windows.Forms.TextBox();
            this.locpmuip = new System.Windows.Forms.TextBox();
            this.locpmuid = new System.Windows.Forms.TextBox();
            this.label29 = new System.Windows.Forms.Label();
```

```
            this.label31 = new System.Windows.Forms.Label();
            this.label30 = new System.Windows.Forms.Label();
            this.tabPage5 = new System.Windows.Forms.TabPage();
            this.rempmuport = new System.Windows.Forms.TextBox();
            this.rempmuip = new System.Windows.Forms.TextBox();
            this.rempmuid = new System.Windows.Forms.TextBox();
            this.label32 = new System.Windows.Forms.Label();
            this.label33 = new System.Windows.Forms.Label();
            this.label34 = new System.Windows.Forms.Label();
            this.tabControl1 = new System.Windows.Forms.TabControl();
            this.tabPage1 = new System.Windows.Forms.TabPage();
            this.panel1 = new System.Windows.Forms.Panel();
            this.dataGridView3 = new System.Windows.Forms.DataGridView();
            this.tabPage2 = new System.Windows.Forms.TabPage();
            this.label4 = new System.Windows.Forms.Label();
            this.label3 = new System.Windows.Forms.Label();
            this.dataGridView2 = new System.Windows.Forms.DataGridView();
            this.idDataGridViewTextBoxColumn1 = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.tsDataGridViewTextBoxColumn1 = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.iAangDataGridViewTextBoxColumn1 = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.iAmagDataGridViewTextBoxColumn1 = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.iBangDataGridViewTextBoxColumn1 = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.iBmagDataGridViewTextBoxColumn1 = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.iCangDataGridViewTextBoxColumn1 = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.iCmagDataGridViewTextBoxColumn1 = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.remotePmuBindingSource = new System.Windows.Forms.BindingSource(this.components);
            this.database1DataSet = new Database1DataSet();
            this.dataGridView1 = new System.Windows.Forms.DataGridView();
            this.idDataGridViewTextBoxColumn = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.tsDataGridViewTextBoxColumn = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.iAangDataGridViewTextBoxColumn = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.iAmagDataGridViewTextBoxColumn = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.iBangDataGridViewTextBoxColumn = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.iBmagDataGridViewTextBoxColumn = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.iCangDataGridViewTextBoxColumn = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.iCmagDataGridViewTextBoxColumn = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.localPmuBindingSource = new System.Windows.Forms.BindingSource(this.components);
            this.tabPage3 = new System.Windows.Forms.TabPage();
            this.chart3 = new System.Windows.Forms.DataVisualization.Charting.Chart();
            this.chart2 = new System.Windows.Forms.DataVisualization.Charting.Chart();
            this.chart1 = new System.Windows.Forms.DataVisualization.Charting.Chart();
            this.groupBox2 = new System.Windows.Forms.GroupBox();
            this.label2 = new System.Windows.Forms.Label();
            this.time = new System.Windows.Forms.Label();
            this.Frequency = new System.Windows.Forms.Label();
            this.label1 = new System.Windows.Forms.Label();
            this.groupBox3 = new System.Windows.Forms.GroupBox();
            this.label17 = new System.Windows.Forms.Label();
            this.label18 = new System.Windows.Forms.Label();
            this.ICmagR = new System.Windows.Forms.Label();
            this.IBmagR = new System.Windows.Forms.Label();
            this.IAmagR = new System.Windows.Forms.Label();
            this.ICangR = new System.Windows.Forms.Label();
```

```
            this.IBangR = new System.Windows.Forms.Label();
            this.IAangR = new System.Windows.Forms.Label();
            this.label25 = new System.Windows.Forms.Label();
            this.label26 = new System.Windows.Forms.Label();
            this.label27 = new System.Windows.Forms.Label();
            this.label28 = new System.Windows.Forms.Label();
            this.groupBox1 = new System.Windows.Forms.GroupBox();
            this.label10 = new System.Windows.Forms.Label();
            this.label9 = new System.Windows.Forms.Label();
            this.ICmag = new System.Windows.Forms.Label();
            this.IBmag = new System.Windows.Forms.Label();
            this.IAmag = new System.Windows.Forms.Label();
            this.ICang = new System.Windows.Forms.Label();
            this.IBang = new System.Windows.Forms.Label();
            this.IAang = new System.Windows.Forms.Label();
            this.label8 = new System.Windows.Forms.Label();
            this.label7 = new System.Windows.Forms.Label();
            this.IAmaglab = new System.Windows.Forms.Label();
            this.label5 = new System.Windows.Forms.Label();
            this.progressBar1 = new System.Windows.Forms.ProgressBar();
            this.menuStrip1 = new System.Windows.Forms.MenuStrip();
            this.fileToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.viewToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.editToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.optionsToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.helpToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.clock = new System.Windows.Forms.Timer(this.components);
            this.sysStatus = new System.Windows.Forms.Label();
            this.localPmuTableAdapter = new Database1DataSetTableAdapters.localPmuTableAdapter();
            this.remotePmuTableAdapter = new
Database1DataSetTableAdapters.remotePmuTableAdapter();
            this.label6 = new System.Windows.Forms.Label();
            this.database2DataSet = new Database2DataSet();
            this.phasorDiffentialArchiveBindingSource = new
System.Windows.Forms.BindingSource(this.components);
            this.phasorDiffentialArchiveTableAdapter = new
Database2DataSetTableAdapters.phasorDiffentialArchiveTableAdapter();
            this.idDataGridViewTextBoxColumn2 = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.tDataGridViewTextBoxColumn = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.differerentialMagnitudeDataGridViewTextBoxColumn = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.differerentialAngleDataGridViewTextBoxColumn = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.phasorDataGridViewTextBoxColumn = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.ConSet.SuspendLayout();
            this.tabControl2.SuspendLayout();
            this.tabPage4.SuspendLayout();
            this.tabPage5.SuspendLayout();
            this.tabControl1.SuspendLayout();
            this.tabPage1.SuspendLayout();
            this.panel1.SuspendLayout();
            ((System.ComponentModel.ISupportInitialize)(this.dataGridView3)).BeginInit();
            this.tabPage2.SuspendLayout();
            ((System.ComponentModel.ISupportInitialize)(this.dataGridView2)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.remotePmuBindingSource)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.database1DataSet)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.localPmuBindingSource)).BeginInit();
            this.tabPage3.SuspendLayout();
            ((System.ComponentModel.ISupportInitialize)(this.chart3)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.chart2)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.chart1)).BeginInit();
            this.groupBox2.SuspendLayout();
            this.groupBox3.SuspendLayout();
            this.groupBox1.SuspendLayout();
            this.menuStrip1.SuspendLayout();
```

```
            ((System.ComponentModel.ISupportInitialize)(this.database2DataSet)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.phasorDiffentialArchiveBindingSource)).BeginInit(
);
            this.SuspendLayout();


            //
            // ConSet
            //
            this.ConSet.Controls.Add(this.sysDisplay);
            this.ConSet.Controls.Add(this.comboBox1);
            this.ConSet.Controls.Add(this.radioButton3);
            this.ConSet.Controls.Add(this.radioButton2);
            this.ConSet.Controls.Add(this.radioButton1);
            this.ConSet.Controls.Add(this.button2);
            this.ConSet.Controls.Add(this.button1);
            this.ConSet.Controls.Add(this.tabControl2);
            this.ConSet.Location = new System.Drawing.Point(12, 82);
            this.ConSet.Name = "ConSet";
            this.ConSet.Size = new System.Drawing.Size(228, 541);
            this.ConSet.TabIndex = 0;
            this.ConSet.TabStop = false;
            this.ConSet.Text = "Connection Settings";
            //
            // sysDisplay
            //
            this.sysDisplay.BackColor = System.Drawing.Color.Black;
            this.sysDisplay.ForeColor = System.Drawing.Color.Lime;
            this.sysDisplay.Location = new System.Drawing.Point(6, 329);
            this.sysDisplay.Multiline = true;
            this.sysDisplay.Name = "sysDisplay";
            this.sysDisplay.Size = new System.Drawing.Size(216, 206);
            this.sysDisplay.TabIndex = 5;
            //
            // comboBox1
            //
            this.comboBox1.FormattingEnabled = true;
            this.comboBox1.Items.AddRange(new object[] {
            "UTK F-Net",
            "IEEE 1344-1995",
            "SEL Fast Message",
            "IEEE C37.118.2-2011",
            "IEC 61850-90-5 (starting with version 4.3)",
            "IEEE C37.118-2005 (Version 1/Draft 7, Draft 6)",
            "Macrodyne (M and G starting with version 4.3)",
            "BPA PDCstream (Revisions 0, 1, and 2, including PDCxchng formatted data)"});
            this.comboBox1.Location = new System.Drawing.Point(6, 187);
            this.comboBox1.Name = "comboBox1";
            this.comboBox1.Size = new System.Drawing.Size(212, 21);
            this.comboBox1.TabIndex = 4;
            this.comboBox1.Click += new System.EventHandler(this.standardSelect);
            //
            // radioButton3
            //
            this.radioButton3.AutoSize = true;
            this.radioButton3.Location = new System.Drawing.Point(10, 266);
            this.radioButton3.Name = "radioButton3";
            this.radioButton3.Size = new System.Drawing.Size(54, 17);
            this.radioButton3.TabIndex = 4;
            this.radioButton3.TabStop = true;
            this.radioButton3.Text = "Serial ";
            this.radioButton3.UseVisualStyleBackColor = true;
            //
            // radioButton2
            //
            this.radioButton2.AutoSize = true;
            this.radioButton2.Location = new System.Drawing.Point(10, 243);
            this.radioButton2.Name = "radioButton2";
```

```
this.radioButton2.Size = new System.Drawing.Size(44, 17);
this.radioButton2.TabIndex = 4;
this.radioButton2.TabStop = true;
this.radioButton2.Text = "Tcp";
this.radioButton2.UseVisualStyleBackColor = true;
//
// radioButton1
//
this.radioButton1.AutoSize = true;
this.radioButton1.Location = new System.Drawing.Point(10, 220);
this.radioButton1.Name = "radioButton1";
this.radioButton1.Size = new System.Drawing.Size(45, 17);
this.radioButton1.TabIndex = 4;
this.radioButton1.TabStop = true;
this.radioButton1.Text = "Udp";
this.radioButton1.UseVisualStyleBackColor = true;
//
// button2
//
this.button2.Location = new System.Drawing.Point(114, 296);
this.button2.Name = "button2";
this.button2.Size = new System.Drawing.Size(104, 22);
this.button2.TabIndex = 4;
this.button2.Text = "discconnect ";
this.button2.UseVisualStyleBackColor = true;
this.button2.Click += new System.EventHandler(this.DisconnectBttn);
//
// button1
//
this.button1.Location = new System.Drawing.Point(6, 296);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(104, 22);
this.button1.TabIndex = 4;
this.button1.Text = "connect ";
this.button1.UseVisualStyleBackColor = true;
this.button1.Click += new System.EventHandler(this.ConnectionButtn);
//
// tabControl2
//
this.tabControl2.Controls.Add(this.tabPage4);
this.tabControl2.Controls.Add(this.tabPage5);
this.tabControl2.Location = new System.Drawing.Point(6, 33);
this.tabControl2.Name = "tabControl2";
this.tabControl2.SelectedIndex = 0;
this.tabControl2.Size = new System.Drawing.Size(216, 142);
this.tabControl2.TabIndex = 0;
//
// tabPage4
//
this.tabPage4.Controls.Add(this.locpmuport);
this.tabPage4.Controls.Add(this.locpmuip);
this.tabPage4.Controls.Add(this.locpmuid);
this.tabPage4.Controls.Add(this.label29);
this.tabPage4.Controls.Add(this.label31);
this.tabPage4.Controls.Add(this.label30);
this.tabPage4.Location = new System.Drawing.Point(4, 22);
this.tabPage4.Name = "tabPage4";
this.tabPage4.Padding = new System.Windows.Forms.Padding(3);
this.tabPage4.Size = new System.Drawing.Size(208, 116);
this.tabPage4.TabIndex = 0;
this.tabPage4.Text = "local Pmu";
this.tabPage4.UseVisualStyleBackColor = true;
//
// locpmuport
//
this.locpmuport.Location = new System.Drawing.Point(76, 75);
this.locpmuport.Name = "locpmuport";
this.locpmuport.Size = new System.Drawing.Size(100, 20);
this.locpmuport.TabIndex = 10;
```

```
this.locpmuport.Text = "4831";


//
// locpmuip
//
this.locpmuip.Location = new System.Drawing.Point(76, 48);
this.locpmuip.Name = "locpmuip";
this.locpmuip.Size = new System.Drawing.Size(100, 20);
this.locpmuip.TabIndex = 11;
this.locpmuip.Text = "192.168.1.204";
//
// locpmuid
//
this.locpmuid.Location = new System.Drawing.Point(76, 21);
this.locpmuid.Name = "locpmuid";
this.locpmuid.Size = new System.Drawing.Size(100, 20);
this.locpmuid.TabIndex = 12;
this.locpmuid.Text = "31";
//
// label29
//
this.label29.AutoSize = true;
this.label29.Location = new System.Drawing.Point(19, 23);
this.label29.Name = "label29";
this.label29.Size = new System.Drawing.Size(38, 13);
this.label29.TabIndex = 7;
this.label29.Text = "pmu id";
//
// label31
//
this.label31.AutoSize = true;
this.label31.Location = new System.Drawing.Point(19, 51);
this.label31.Name = "label31";
this.label31.Size = new System.Drawing.Size(38, 13);
this.label31.TabIndex = 8;
this.label31.Text = "pmu ip";
//
// label30
//
this.label30.AutoSize = true;
this.label30.Location = new System.Drawing.Point(9, 79);
this.label30.Name = "label30";
this.label30.Size = new System.Drawing.Size(48, 13);
this.label30.TabIndex = 9;
this.label30.Text = "pmu port";
//
// tabPage5
//
this.tabPage5.Controls.Add(this.rempmuport);
this.tabPage5.Controls.Add(this.rempmuip);
this.tabPage5.Controls.Add(this.rempmuid);
this.tabPage5.Controls.Add(this.label32);
this.tabPage5.Controls.Add(this.label33);
this.tabPage5.Controls.Add(this.label34);
this.tabPage5.Location = new System.Drawing.Point(4, 22);
this.tabPage5.Name = "tabPage5";
this.tabPage5.Padding = new System.Windows.Forms.Padding(3);
this.tabPage5.Size = new System.Drawing.Size(208, 116);
this.tabPage5.TabIndex = 1;
this.tabPage5.Text = "remote Pmu";
this.tabPage5.UseVisualStyleBackColor = true;
//
// rempmuport
//
this.rempmuport.Location = new System.Drawing.Point(77, 75);
this.rempmuport.Name = "rempmuport";
this.rempmuport.Size = new System.Drawing.Size(100, 20);
this.rempmuport.TabIndex = 16;
```

```
this.rempmuport.Text = "4830";


//
// rempmuip
//
this.rempmuip.Location = new System.Drawing.Point(77, 48);
this.rempmuip.Name = "rempmuip";
this.rempmuip.Size = new System.Drawing.Size(100, 20);
this.rempmuip.TabIndex = 17;
this.rempmuip.Text = "192.168.1.204";
//
// rempmuid
//
this.rempmuid.Location = new System.Drawing.Point(77, 21);
this.rempmuid.Name = "rempmuid";
this.rempmuid.Size = new System.Drawing.Size(100, 20);
this.rempmuid.TabIndex = 18;
this.rempmuid.Text = "30";
//
// label32
//
this.label32.AutoSize = true;
this.label32.Location = new System.Drawing.Point(20, 23);
this.label32.Name = "label32";
this.label32.Size = new System.Drawing.Size(38, 13);
this.label32.TabIndex = 13;
this.label32.Text = "pmu id";
//
// label33
//
this.label33.AutoSize = true;
this.label33.Location = new System.Drawing.Point(20, 51);
this.label33.Name = "label33";
this.label33.Size = new System.Drawing.Size(38, 13);
this.label33.TabIndex = 14;
this.label33.Text = "pmu ip";
//
// label34
//
this.label34.AutoSize = true;
this.label34.Location = new System.Drawing.Point(10, 79);
this.label34.Name = "label34";
this.label34.Size = new System.Drawing.Size(48, 13);
this.label34.TabIndex = 15;
this.label34.Text = "pmu port";
//
// tabControl1
//
this.tabControl1.Controls.Add(this.tabPage1);
this.tabControl1.Controls.Add(this.tabPage2);
this.tabControl1.Controls.Add(this.tabPage3);
this.tabControl1.Location = new System.Drawing.Point(271, 82);
this.tabControl1.Name = "tabControl1";
this.tabControl1.SelectedIndex = 0;
this.tabControl1.Size = new System.Drawing.Size(507, 541);
this.tabControl1.TabIndex = 1;
//
// tabPage1
//
this.tabPage1.Controls.Add(this.panel1);
this.tabPage1.Location = new System.Drawing.Point(4, 22);
this.tabPage1.Name = "tabPage1";
this.tabPage1.Padding = new System.Windows.Forms.Padding(3);
this.tabPage1.Size = new System.Drawing.Size(499, 515);
this.tabPage1.TabIndex = 0;
this.tabPage1.Text = "Grid Monitor";
this.tabPage1.UseVisualStyleBackColor = true;
//
```

```
            // panel1
            //
            this.panel1.BackColor = System.Drawing.Color.Transparent;
            this.panel1.Controls.Add(this.dataGridView3);
            this.panel1.Location = new System.Drawing.Point(-4, 0);
            this.panel1.Name = "panel1";
            this.panel1.Size = new System.Drawing.Size(503, 513);
            this.panel1.TabIndex = 0;
            //
            // dataGridView3
            //
            this.dataGridView3.AutoGenerateColumns = false;
            this.dataGridView3.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
            this.dataGridView3.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[] {
            this.idDataGridViewTextBoxColumn2,
            this.tDataGridViewTextBoxColumn,
            this.differerentialMagnitudeDataGridViewTextBoxColumn,
            this.differerentialAngleDataGridViewTextBoxColumn,
            this.phasorDataGridViewTextBoxColumn});
            this.dataGridView3.DataSource = this.phasorDiffentialArchiveBindingSource;
            this.dataGridView3.Location = new System.Drawing.Point(3, 0);
            this.dataGridView3.Name = "dataGridView3";
            this.dataGridView3.Size = new System.Drawing.Size(499, 512);
            this.dataGridView3.TabIndex = 0;
            //
            // tabPage2
            //
            this.tabPage2.Controls.Add(this.label4);
            this.tabPage2.Controls.Add(this.label3);
            this.tabPage2.Controls.Add(this.dataGridView2);
            this.tabPage2.Controls.Add(this.dataGridView1);
            this.tabPage2.Location = new System.Drawing.Point(4, 22);
            this.tabPage2.Name = "tabPage2";
            this.tabPage2.Padding = new System.Windows.Forms.Padding(3);
            this.tabPage2.Size = new System.Drawing.Size(499, 515);
            this.tabPage2.TabIndex = 1;
            this.tabPage2.Text = "Phasor Data Archive";
            this.tabPage2.UseVisualStyleBackColor = true;
            //
            // label4
            //
            this.label4.AutoSize = true;
            this.label4.Location = new System.Drawing.Point(3, 298);
            this.label4.Name = "label4";
            this.label4.Size = new System.Drawing.Size(169, 13);
            this.label4.TabIndex = 2;
            this.label4.Text = "Remote Pmu Phasor Data Archive";
            //
            // label3
            //
            this.label3.AutoSize = true;
            this.label3.Location = new System.Drawing.Point(6, 11);
            this.label3.Name = "label3";
            this.label3.Size = new System.Drawing.Size(158, 13);
            this.label3.TabIndex = 2;
            this.label3.Text = "Local Pmu Phasor Data Archive";
            //
            // dataGridView2
            //
            this.dataGridView2.AutoGenerateColumns = false;
            this.dataGridView2.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
            this.dataGridView2.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[] {
            this.idDataGridViewTextBoxColumn1,
            this.tsDataGridViewTextBoxColumn1,
            this.iAangDataGridViewTextBoxColumn1,
            this.iAmagDataGridViewTextBoxColumn1,
            this.iBangDataGridViewTextBoxColumn1,
```

```
            this.iBmagDataGridViewTextBoxColumn1,
            this.iCangDataGridViewTextBoxColumn1,
            this.iCmagDataGridViewTextBoxColumn1});
            this.dataGridView2.DataSource = this.remotePmuBindingSource;
            this.dataGridView2.Location = new System.Drawing.Point(1, 314);
            this.dataGridView2.Name = "dataGridView2";
            this.dataGridView2.Size = new System.Drawing.Size(502, 205);
            this.dataGridView2.TabIndex = 1;
            // 
            // idDataGridViewTextBoxColumn1
            // 
            this.idDataGridViewTextBoxColumn1.DataPropertyName = "Id";
            this.idDataGridViewTextBoxColumn1.HeaderText = "Id";
            this.idDataGridViewTextBoxColumn1.Name = "idDataGridViewTextBoxColumn1";
            this.idDataGridViewTextBoxColumn1.ReadOnly = true;
            this.idDataGridViewTextBoxColumn1.Width = 30;
            // 
            // tsDataGridViewTextBoxColumn1
            // 
            this.tsDataGridViewTextBoxColumn1.DataPropertyName = "ts";
            this.tsDataGridViewTextBoxColumn1.HeaderText = "ts";
            this.tsDataGridViewTextBoxColumn1.Name = "tsDataGridViewTextBoxColumn1";
            // 
            // iAangDataGridViewTextBoxColumn1
            // 
            this.iAangDataGridViewTextBoxColumn1.DataPropertyName = "IAang";
            this.iAangDataGridViewTextBoxColumn1.HeaderText = "IAang";
            this.iAangDataGridViewTextBoxColumn1.Name = "iAangDataGridViewTextBoxColumn1";
            // 
            // iAmagDataGridViewTextBoxColumn1
            // 
            this.iAmagDataGridViewTextBoxColumn1.DataPropertyName = "IAmag";
            this.iAmagDataGridViewTextBoxColumn1.HeaderText = "IAmag";
            this.iAmagDataGridViewTextBoxColumn1.Name = "iAmagDataGridViewTextBoxColumn1";
            // 
            // iBangDataGridViewTextBoxColumn1
            // 
            this.iBangDataGridViewTextBoxColumn1.DataPropertyName = "IBang";
            this.iBangDataGridViewTextBoxColumn1.HeaderText = "IBang";
            this.iBangDataGridViewTextBoxColumn1.Name = "iBangDataGridViewTextBoxColumn1";
            // 
            // iBmagDataGridViewTextBoxColumn1
            // 
            this.iBmagDataGridViewTextBoxColumn1.DataPropertyName = "IBmag";
            this.iBmagDataGridViewTextBoxColumn1.HeaderText = "IBmag";
            this.iBmagDataGridViewTextBoxColumn1.Name = "iBmagDataGridViewTextBoxColumn1";
            // 
            // iCangDataGridViewTextBoxColumn1
            // 
            this.iCangDataGridViewTextBoxColumn1.DataPropertyName = "ICang";
            this.iCangDataGridViewTextBoxColumn1.HeaderText = "ICang";
            this.iCangDataGridViewTextBoxColumn1.Name = "iCangDataGridViewTextBoxColumn1";
            // 
            // iCmagDataGridViewTextBoxColumn1
            // 
            this.iCmagDataGridViewTextBoxColumn1.DataPropertyName = "ICmag";
            this.iCmagDataGridViewTextBoxColumn1.HeaderText = "ICmag";
            this.iCmagDataGridViewTextBoxColumn1.Name = "iCmagDataGridViewTextBoxColumn1";
            // 
            // remotePmuBindingSource
            // 
            this.remotePmuBindingSource.DataMember = "remotePmu";
            this.remotePmuBindingSource.DataSource = this.database1DataSet;
            // 
            // database1DataSet
            // 
            this.database1DataSet.DataSetName = "Database1DataSet";
            this.database1DataSet.SchemaSerializationMode = 
System.Data.SchemaSerializationMode.IncludeSchema;
```

```
            //
            // dataGridView1
            //
            this.dataGridView1.AutoGenerateColumns = false;
            this.dataGridView1.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D;
            this.dataGridView1.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
            this.dataGridView1.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[] {
            this.idDataGridViewTextBoxColumn,
            this.tsDataGridViewTextBoxColumn,
            this.iAangDataGridViewTextBoxColumn,
            this.iAmagDataGridViewTextBoxColumn,
            this.iBangDataGridViewTextBoxColumn,
            this.iBmagDataGridViewTextBoxColumn,
            this.iCangDataGridViewTextBoxColumn,
            this.iCmagDataGridViewTextBoxColumn});
            this.dataGridView1.DataSource = this.localPmuBindingSource;
            this.dataGridView1.Location = new System.Drawing.Point(0, 28);
            this.dataGridView1.Name = "dataGridView1";
            this.dataGridView1.Size = new System.Drawing.Size(498, 258);
            this.dataGridView1.TabIndex = 0;
            //
            // idDataGridViewTextBoxColumn
            //
            this.idDataGridViewTextBoxColumn.DataPropertyName = "Id";
            this.idDataGridViewTextBoxColumn.HeaderText = "Id";
            this.idDataGridViewTextBoxColumn.Name = "idDataGridViewTextBoxColumn";
            this.idDataGridViewTextBoxColumn.ReadOnly = true;
            this.idDataGridViewTextBoxColumn.Width = 30;
            //
            // tsDataGridViewTextBoxColumn
            //
            this.tsDataGridViewTextBoxColumn.DataPropertyName = "ts";
            this.tsDataGridViewTextBoxColumn.HeaderText = "ts";
            this.tsDataGridViewTextBoxColumn.Name = "tsDataGridViewTextBoxColumn";
            //
            // iAangDataGridViewTextBoxColumn
            //
            this.iAangDataGridViewTextBoxColumn.DataPropertyName = "IAang";
            this.iAangDataGridViewTextBoxColumn.HeaderText = "IAang";
            this.iAangDataGridViewTextBoxColumn.Name = "iAangDataGridViewTextBoxColumn";
            //
            // iAmagDataGridViewTextBoxColumn
            //
            this.iAmagDataGridViewTextBoxColumn.DataPropertyName = "IAmag";
            this.iAmagDataGridViewTextBoxColumn.HeaderText = "IAmag";
            this.iAmagDataGridViewTextBoxColumn.Name = "iAmagDataGridViewTextBoxColumn";
            //
            // iBangDataGridViewTextBoxColumn
            //
            this.iBangDataGridViewTextBoxColumn.DataPropertyName = "IBang";
            this.iBangDataGridViewTextBoxColumn.HeaderText = "IBang";
            this.iBangDataGridViewTextBoxColumn.Name = "iBangDataGridViewTextBoxColumn";
            //
            // iBmagDataGridViewTextBoxColumn
            //
            this.iBmagDataGridViewTextBoxColumn.DataPropertyName = "IBmag";
            this.iBmagDataGridViewTextBoxColumn.HeaderText = "IBmag";
            this.iBmagDataGridViewTextBoxColumn.Name = "iBmagDataGridViewTextBoxColumn";
            //
            // iCangDataGridViewTextBoxColumn
            //
            this.iCangDataGridViewTextBoxColumn.DataPropertyName = "ICang";
            this.iCangDataGridViewTextBoxColumn.HeaderText = "ICang";
            this.iCangDataGridViewTextBoxColumn.Name = "iCangDataGridViewTextBoxColumn";
            //
            // iCmagDataGridViewTextBoxColumn
            //
            this.iCmagDataGridViewTextBoxColumn.DataPropertyName = "ICmag";
```

```
            this.iCmagDataGridViewTextBoxColumn.HeaderText = "ICmag";
            this.iCmagDataGridViewTextBoxColumn.Name = "iCmagDataGridViewTextBoxColumn";
            //
            // localPmuBindingSource
            //
            this.localPmuBindingSource.DataMember = "localPmu";
            this.localPmuBindingSource.DataSource = this.database1DataSet;
            //
            // tabPage3
            //
            this.tabPage3.Controls.Add(this.chart3);
            this.tabPage3.Controls.Add(this.chart2);
            this.tabPage3.Controls.Add(this.chart1);
            this.tabPage3.Location = new System.Drawing.Point(4, 22);
            this.tabPage3.Name = "tabPage3";
            this.tabPage3.Padding = new System.Windows.Forms.Padding(3);
            this.tabPage3.Size = new System.Drawing.Size(499, 515);
            this.tabPage3.TabIndex = 2;
            this.tabPage3.Text = "realTime Correlationa & Analysis";
            this.tabPage3.UseVisualStyleBackColor = true;
            //
            // chart3
            //
            chartArea4.AxisX.MajorGrid.LineColor = System.Drawing.Color.Lime;
            chartArea4.AxisY.MajorGrid.LineColor = System.Drawing.Color.Lime;
            chartArea4.BackColor = System.Drawing.Color.Black;
            chartArea4.Name = "ChartArea1";
            this.chart3.ChartAreas.Add(chartArea4);
            legend4.IsDockedInsideChartArea = false;
            legend4.MaximumAutoSize = 20F;
            legend4.Name = "Legend1";
            this.chart3.Legends.Add(legend4);
            this.chart3.Location = new System.Drawing.Point(-29, 345);
            this.chart3.Name = "chart3";
            series7.ChartArea = "ChartArea1";
            series7.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Spline;
            series7.Legend = "Legend1";
            series7.Name = "loc  :IC";
            series8.ChartArea = "ChartArea1";
            series8.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Spline;
            series8.Legend = "Legend1";
            series8.Name = "rem :IC";
            this.chart3.Series.Add(series7);
            this.chart3.Series.Add(series8);
            this.chart3.Size = new System.Drawing.Size(542, 146);
            this.chart3.TabIndex = 0;
            this.chart3.Text = "chart1";
            //
            // chart2
            //
            chartArea5.AxisX.MajorGrid.LineColor = System.Drawing.Color.Lime;
            chartArea5.AxisY.MajorGrid.LineColor = System.Drawing.Color.Lime;
            chartArea5.BackColor = System.Drawing.Color.Black;
            chartArea5.Name = "ChartArea1";
            this.chart2.ChartAreas.Add(chartArea5);
            legend5.IsDockedInsideChartArea = false;
            legend5.MaximumAutoSize = 20F;
            legend5.Name = "Legend1";
            this.chart2.Legends.Add(legend5);
            this.chart2.Location = new System.Drawing.Point(-23, 184);
            this.chart2.Name = "chart2";
            series9.ChartArea = "ChartArea1";
            series9.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Spline;
            series9.Legend = "Legend1";
            series9.Name = "loc  :IB";
            series10.ChartArea = "ChartArea1";
```

```
            series10.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Spline;
            series10.Legend = "Legend1";
            series10.Name = "rem :IB";
            this.chart2.Series.Add(series9);
            this.chart2.Series.Add(series10);
            this.chart2.Size = new System.Drawing.Size(542, 146);
            this.chart2.TabIndex = 0;
            this.chart2.Text = "chart1";
            //
            // chart1
            //
            chartArea6.AxisX.MajorGrid.LineColor = System.Drawing.Color.Lime;
            chartArea6.AxisY.MajorGrid.LineColor = System.Drawing.Color.Lime;
            chartArea6.BackColor = System.Drawing.Color.Black;
            chartArea6.Name = "ChartArea1";
            this.chart1.ChartAreas.Add(chartArea6);
            legend6.IsDockedInsideChartArea = false;
            legend6.MaximumAutoSize = 20F;
            legend6.Name = "Legend1";
            this.chart1.Legends.Add(legend6);
            this.chart1.Location = new System.Drawing.Point(-23, 23);
            this.chart1.Name = "chart1";
            series11.ChartArea = "ChartArea1";
            series11.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Spline;
            series11.Legend = "Legend1";
            series11.Name = "loc  :IA";
            series12.ChartArea = "ChartArea1";
            series12.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Spline;
            series12.Legend = "Legend1";
            series12.Name = "rem :IA";
            this.chart1.Series.Add(series11);
            this.chart1.Series.Add(series12);
            this.chart1.Size = new System.Drawing.Size(542, 146);
            this.chart1.TabIndex = 0;
            this.chart1.Text = "chart1";
            //
            // groupBox2
            //
            this.groupBox2.Controls.Add(this.label2);
            this.groupBox2.Controls.Add(this.time);
            this.groupBox2.Controls.Add(this.Frequency);
            this.groupBox2.Controls.Add(this.label1);
            this.groupBox2.Controls.Add(this.groupBox3);
            this.groupBox2.Controls.Add(this.groupBox1);
            this.groupBox2.Location = new System.Drawing.Point(784, 82);
            this.groupBox2.Name = "groupBox2";
            this.groupBox2.Size = new System.Drawing.Size(248, 541);
            this.groupBox2.TabIndex = 0;
            this.groupBox2.TabStop = false;
            this.groupBox2.Text = "Phasors";
            //
            // label2
            //
            this.label2.AutoSize = true;
            this.label2.Location = new System.Drawing.Point(6, 501);
            this.label2.Name = "label2";
            this.label2.Size = new System.Drawing.Size(90, 13);
            this.label2.TabIndex = 1;
            this.label2.Text = "time stamp (GMT)";
            //
            // time
            //
            this.time.AutoSize = true;
            this.time.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.time.Location = new System.Drawing.Point(94, 501);
```

```
            this.time.Name = "time";
            this.time.Size = new System.Drawing.Size(10, 15);
            this.time.TabIndex = 1;
            this.time.Text = ":";
            //
            // Frequency
            //
            this.Frequency.AutoSize = true;
            this.Frequency.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.Frequency.Location = new System.Drawing.Point(94, 466);
            this.Frequency.Name = "Frequency";
            this.Frequency.Size = new System.Drawing.Size(10, 15);
            this.Frequency.TabIndex = 1;
            this.Frequency.Text = ":";
            //
            // label1
            //
            this.label1.AutoSize = true;
            this.label1.Location = new System.Drawing.Point(34, 466);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(54, 13);
            this.label1.TabIndex = 1;
            this.label1.Text = "frequency";
            //
            // groupBox3
            //
            this.groupBox3.Controls.Add(this.label17);
            this.groupBox3.Controls.Add(this.label18);
            this.groupBox3.Controls.Add(this.ICmagR);
            this.groupBox3.Controls.Add(this.IBmagR);
            this.groupBox3.Controls.Add(this.IAmagR);
            this.groupBox3.Controls.Add(this.ICangR);
            this.groupBox3.Controls.Add(this.IBangR);
            this.groupBox3.Controls.Add(this.IAangR);
            this.groupBox3.Controls.Add(this.label25);
            this.groupBox3.Controls.Add(this.label26);
            this.groupBox3.Controls.Add(this.label27);
            this.groupBox3.Controls.Add(this.label28);
            this.groupBox3.Location = new System.Drawing.Point(16, 243);
            this.groupBox3.Name = "groupBox3";
            this.groupBox3.RightToLeft = System.Windows.Forms.RightToLeft.Yes;
            this.groupBox3.Size = new System.Drawing.Size(215, 206);
            this.groupBox3.TabIndex = 0;
            this.groupBox3.TabStop = false;
            this.groupBox3.Text = "remote Pmu ";
            //
            // label17
            //
            this.label17.AutoSize = true;
            this.label17.Location = new System.Drawing.Point(6, 170);
            this.label17.Name = "label17";
            this.label17.Size = new System.Drawing.Size(69, 13);
            this.label17.TabIndex = 1;
            this.label17.Text = "IC magnitude";
            //
            // label18
            //
            this.label18.AutoSize = true;
            this.label18.Location = new System.Drawing.Point(26, 86);
            this.label18.Name = "label18";
            this.label18.Size = new System.Drawing.Size(49, 13);
            this.label18.TabIndex = 1;
            this.label18.Text = "IC angle ";
            //
            // ICmagR
            //
            this.ICmagR.AutoSize = true;
```

```
            this.ICmagR.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.ICmagR.Location = new System.Drawing.Point(81, 170);
            this.ICmagR.Name = "ICmagR";
            this.ICmagR.Size = new System.Drawing.Size(10, 13);
            this.ICmagR.TabIndex = 1;
            this.ICmagR.Text = ":";
            //
            // IBmagR
            //
            this.IBmagR.AutoSize = true;
            this.IBmagR.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.IBmagR.Location = new System.Drawing.Point(81, 142);
            this.IBmagR.Name = "IBmagR";
            this.IBmagR.Size = new System.Drawing.Size(10, 13);
            this.IBmagR.TabIndex = 1;
            this.IBmagR.Text = ":";
            //
            // IAmagR
            //
            this.IAmagR.AutoSize = true;
            this.IAmagR.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.IAmagR.Location = new System.Drawing.Point(81, 114);
            this.IAmagR.Name = "IAmagR";
            this.IAmagR.Size = new System.Drawing.Size(10, 13);
            this.IAmagR.TabIndex = 1;
            this.IAmagR.Text = ":";
            //
            // ICangR
            //
            this.ICangR.AutoSize = true;
            this.ICangR.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.ICangR.Location = new System.Drawing.Point(81, 86);
            this.ICangR.Name = "ICangR";
            this.ICangR.Size = new System.Drawing.Size(10, 13);
            this.ICangR.TabIndex = 1;
            this.ICangR.Text = ":";
            //
            // IBangR
            //
            this.IBangR.AutoSize = true;
            this.IBangR.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.IBangR.Location = new System.Drawing.Point(81, 58);
            this.IBangR.Name = "IBangR";
            this.IBangR.Size = new System.Drawing.Size(10, 13);
            this.IBangR.TabIndex = 1;
            this.IBangR.Text = ":";
            //
            // IAangR
            //
            this.IAangR.AutoSize = true;
            this.IAangR.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.IAangR.Location = new System.Drawing.Point(81, 30);
            this.IAangR.Name = "IAangR";
            this.IAangR.Size = new System.Drawing.Size(10, 13);
            this.IAangR.TabIndex = 1;
            this.IAangR.Text = ":";
            //
            // label25
            //
            this.label25.AutoSize = true;
            this.label25.Location = new System.Drawing.Point(6, 142);
            this.label25.Name = "label25";
            this.label25.Size = new System.Drawing.Size(69, 13);
```

```
this.label25.TabIndex = 1;
this.label25.Text = "IB magnitude";
//
// label26
//
this.label26.AutoSize = true;
this.label26.Location = new System.Drawing.Point(26, 58);
this.label26.Name = "label26";
this.label26.Size = new System.Drawing.Size(49, 13);
this.label26.TabIndex = 1;
this.label26.Text = "IB angle ";
//
// label27
//
this.label27.AutoSize = true;
this.label27.Location = new System.Drawing.Point(6, 114);
this.label27.Name = "label27";
this.label27.Size = new System.Drawing.Size(69, 13);
this.label27.TabIndex = 1;
this.label27.Text = "IA magnitude";
//
// label28
//
this.label28.AutoSize = true;
this.label28.Location = new System.Drawing.Point(26, 30);
this.label28.Name = "label28";
this.label28.Size = new System.Drawing.Size(49, 13);
this.label28.TabIndex = 1;
this.label28.Text = "IA angle ";
//
// groupBox1
//
this.groupBox1.Controls.Add(this.label10);
this.groupBox1.Controls.Add(this.label9);
this.groupBox1.Controls.Add(this.ICmag);
this.groupBox1.Controls.Add(this.IBmag);
this.groupBox1.Controls.Add(this.IAmag);
this.groupBox1.Controls.Add(this.ICang);
this.groupBox1.Controls.Add(this.IBang);
this.groupBox1.Controls.Add(this.IAang);
this.groupBox1.Controls.Add(this.label8);
this.groupBox1.Controls.Add(this.label7);
this.groupBox1.Controls.Add(this.IAmaglab);
this.groupBox1.Controls.Add(this.label5);
this.groupBox1.Location = new System.Drawing.Point(16, 20);
this.groupBox1.Name = "groupBox1";
this.groupBox1.RightToLeft = System.Windows.Forms.RightToLeft.Yes;
this.groupBox1.Size = new System.Drawing.Size(215, 206);
this.groupBox1.TabIndex = 0;
this.groupBox1.TabStop = false;
this.groupBox1.Text = "local Pmu ";
//
// label10
//
this.label10.AutoSize = true;
this.label10.Location = new System.Drawing.Point(6, 170);
this.label10.Name = "label10";
this.label10.Size = new System.Drawing.Size(69, 13);
this.label10.TabIndex = 1;
this.label10.Text = "IC magnitude";
//
// label9
//
this.label9.AutoSize = true;
this.label9.Location = new System.Drawing.Point(26, 86);
this.label9.Name = "label9";
this.label9.Size = new System.Drawing.Size(49, 13);
this.label9.TabIndex = 1;
this.label9.Text = "IC angle ";
```

```
            //
            // ICmag
            //
            this.ICmag.AutoSize = true;
            this.ICmag.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.ICmag.Location = new System.Drawing.Point(81, 170);
            this.ICmag.Name = "ICmag";
            this.ICmag.Size = new System.Drawing.Size(10, 13);
            this.ICmag.TabIndex = 1;
            this.ICmag.Text = ":";
            //
            // IBmag
            //
            this.IBmag.AutoSize = true;
            this.IBmag.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.IBmag.Location = new System.Drawing.Point(81, 142);
            this.IBmag.Name = "IBmag";
            this.IBmag.Size = new System.Drawing.Size(10, 13);
            this.IBmag.TabIndex = 1;
            this.IBmag.Text = ":";
            //
            // IAmag
            //
            this.IAmag.AutoSize = true;
            this.IAmag.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.IAmag.Location = new System.Drawing.Point(81, 114);
            this.IAmag.Name = "IAmag";
            this.IAmag.Size = new System.Drawing.Size(10, 13);
            this.IAmag.TabIndex = 1;
            this.IAmag.Text = ":";
            //
            // ICang
            //
            this.ICang.AutoSize = true;
            this.ICang.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.ICang.Location = new System.Drawing.Point(81, 86);
            this.ICang.Name = "ICang";
            this.ICang.Size = new System.Drawing.Size(10, 13);
            this.ICang.TabIndex = 1;
            this.ICang.Text = ":";
            //
            // IBang
            //
            this.IBang.AutoSize = true;
            this.IBang.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.IBang.Location = new System.Drawing.Point(81, 58);
            this.IBang.Name = "IBang";
            this.IBang.Size = new System.Drawing.Size(10, 13);
            this.IBang.TabIndex = 1;
            this.IBang.Text = ":";
            //
            // IAang
            //
            this.IAang.AutoSize = true;
            this.IAang.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.IAang.Location = new System.Drawing.Point(81, 30);
            this.IAang.Name = "IAang";
            this.IAang.Size = new System.Drawing.Size(10, 13);
            this.IAang.TabIndex = 1;
            this.IAang.Text = ":";
            //
            // label8
            //
```

```
this.label8.AutoSize = true;
this.label8.Location = new System.Drawing.Point(6, 142);
this.label8.Name = "label8";
this.label8.Size = new System.Drawing.Size(69, 13);
this.label8.TabIndex = 1;
this.label8.Text = "IB magnitude";
//
// label7
//
this.label7.AutoSize = true;
this.label7.Location = new System.Drawing.Point(26, 58);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(49, 13);
this.label7.TabIndex = 1;
this.label7.Text = "IB angle ";
//
// IAmaglab
//
this.IAmaglab.AutoSize = true;
this.IAmaglab.Location = new System.Drawing.Point(6, 114);
this.IAmaglab.Name = "IAmaglab";
this.IAmaglab.Size = new System.Drawing.Size(69, 13);
this.IAmaglab.TabIndex = 1;
this.IAmaglab.Text = "IA magnitude";
//
// label5
//
this.label5.AutoSize = true;
this.label5.Location = new System.Drawing.Point(26, 30);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(49, 13);
this.label5.TabIndex = 1;
this.label5.Text = "IA angle ";
//
// progressBar1
//
this.progressBar1.Location = new System.Drawing.Point(0, 645);
this.progressBar1.Name = "progressBar1";
this.progressBar1.Size = new System.Drawing.Size(1046, 5);
this.progressBar1.TabIndex = 2;
//
// menuStrip1
//
this.menuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
this.fileToolStripMenuItem,
this.viewToolStripMenuItem,
this.editToolStripMenuItem,
this.optionsToolStripMenuItem,
this.helpToolStripMenuItem});
this.menuStrip1.Location = new System.Drawing.Point(0, 0);
this.menuStrip1.Name = "menuStrip1";
this.menuStrip1.Size = new System.Drawing.Size(1044, 24);
this.menuStrip1.TabIndex = 3;
this.menuStrip1.Text = "menuStrip1";
//
// fileToolStripMenuItem
//
this.fileToolStripMenuItem.Name = "fileToolStripMenuItem";
this.fileToolStripMenuItem.Size = new System.Drawing.Size(37, 20);
this.fileToolStripMenuItem.Text = "File";
//
// viewToolStripMenuItem
//
this.viewToolStripMenuItem.Name = "viewToolStripMenuItem";
this.viewToolStripMenuItem.Size = new System.Drawing.Size(44, 20);
this.viewToolStripMenuItem.Text = "View";
//
// editToolStripMenuItem
//
```

```
            this.editToolStripMenuItem.Name = "editToolStripMenuItem";
            this.editToolStripMenuItem.Size = new System.Drawing.Size(42, 20);
            this.editToolStripMenuItem.Text = "Edit ";
            //
            // optionsToolStripMenuItem
            //
            this.optionsToolStripMenuItem.Name = "optionsToolStripMenuItem";
            this.optionsToolStripMenuItem.Size = new System.Drawing.Size(61, 20);
            this.optionsToolStripMenuItem.Text = "Options";
            //
            // helpToolStripMenuItem
            //
            this.helpToolStripMenuItem.Name = "helpToolStripMenuItem";
            this.helpToolStripMenuItem.Size = new System.Drawing.Size(44, 20);
            this.helpToolStripMenuItem.Text = "Help";
            //
            // clock
            //
            this.clock.Tick += new System.EventHandler(this.clock_Tick);
            //
            // sysStatus
            //
            this.sysStatus.AutoSize = true;
            this.sysStatus.Location = new System.Drawing.Point(-3, 626);
            this.sysStatus.Name = "sysStatus";
            this.sysStatus.Size = new System.Drawing.Size(76, 13);
            this.sysStatus.TabIndex = 4;
            this.sysStatus.Text = "system : offline";
            this.sysStatus.Click += new System.EventHandler(this.sysStatus_Click);

            //
            // localPmuTableAdapter
            //
            this.localPmuTableAdapter.ClearBeforeFill = true;
            //
            // remotePmuTableAdapter
            //
            this.remotePmuTableAdapter.ClearBeforeFill = true;
            //
            // label6
            //
            this.label6.AutoSize = true;
            this.label6.ForeColor = System.Drawing.SystemColors.ActiveCaption;
            this.label6.Location = new System.Drawing.Point(898, 630);
            this.label6.Name = "label6";
            this.label6.Size = new System.Drawing.Size(134, 13);
            this.label6.TabIndex = 5;
            this.label6.Text = "PmuLive Version 1.0.0.250";
            //
            // database2DataSet
            //
            this.database2DataSet.DataSetName = "Database2DataSet";
            this.database2DataSet.SchemaSerializationMode =
System.Data.SchemaSerializationMode.IncludeSchema;
            //
            // phasorDiffentialArchiveBindingSource
            //
            this.phasorDiffentialArchiveBindingSource.DataMember = "phasorDiffentialArchive";
            this.phasorDiffentialArchiveBindingSource.DataSource = this.database2DataSet;
            //
            // phasorDiffentialArchiveTableAdapter
            //
            this.phasorDiffentialArchiveTableAdapter.ClearBeforeFill = true;
            //
            // idDataGridViewTextBoxColumn2
            //
            this.idDataGridViewTextBoxColumn2.DataPropertyName = "Id";
            this.idDataGridViewTextBoxColumn2.HeaderText = "Id";
            this.idDataGridViewTextBoxColumn2.Name = "idDataGridViewTextBoxColumn2";
```

```
            this.idDataGridViewTextBoxColumn2.ReadOnly = true;
            this.idDataGridViewTextBoxColumn2.Width = 70;
            //
            // tDataGridViewTextBoxColumn
            //
            this.tDataGridViewTextBoxColumn.DataPropertyName = "t";
            this.tDataGridViewTextBoxColumn.HeaderText = "t";
            this.tDataGridViewTextBoxColumn.Name = "tDataGridViewTextBoxColumn";
            //
            // differerentialMagnitudeDataGridViewTextBoxColumn
            //
            this.differerentialMagnitudeDataGridViewTextBoxColumn.DataPropertyName =
"differerentialMagnitude";
            this.differerentialMagnitudeDataGridViewTextBoxColumn.HeaderText =
"differerentialMagnitude";
            this.differerentialMagnitudeDataGridViewTextBoxColumn.Name =
"differerentialMagnitudeDataGridViewTextBoxColumn";
            //
            // differerentialAngleDataGridViewTextBoxColumn
            //
            this.differerentialAngleDataGridViewTextBoxColumn.DataPropertyName =
"differerentialAngle";
            this.differerentialAngleDataGridViewTextBoxColumn.HeaderText = "differerentialAngle";
            this.differerentialAngleDataGridViewTextBoxColumn.Name =
"differerentialAngleDataGridViewTextBoxColumn";
            //
            // phasorDataGridViewTextBoxColumn
            //
            this.phasorDataGridViewTextBoxColumn.DataPropertyName = "phasor";
            this.phasorDataGridViewTextBoxColumn.HeaderText = "phasor";
            this.phasorDataGridViewTextBoxColumn.Name = "phasorDataGridViewTextBoxColumn";

            //
            // pmuLive
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(1044, 652);
            this.Controls.Add(this.label6);
            this.Controls.Add(this.sysStatus);
            this.Controls.Add(this.progressBar1);
            this.Controls.Add(this.tabControl1);
            this.Controls.Add(this.groupBox2);
            this.Controls.Add(this.ConSet);
            this.Controls.Add(this.menuStrip1);
            this.MainMenuStrip = this.menuStrip1;
            this.MaximizeBox = false;
            this.Name = "pmuLive";
            this.ShowIcon = false;
            this.Text = "Form1";
            this.Load += new System.EventHandler(this.pmuLive_Load);
            this.ConSet.ResumeLayout(false);
            this.ConSet.PerformLayout();
            this.tabControl2.ResumeLayout(false);
            this.tabPage4.ResumeLayout(false);
            this.tabPage4.PerformLayout();
            this.tabPage5.ResumeLayout(false);
            this.tabPage5.PerformLayout();
            this.tabControl1.ResumeLayout(false);
            this.tabPage1.ResumeLayout(false);
            this.panel1.ResumeLayout(false);
            ((System.ComponentModel.ISupportInitialize)(this.dataGridView3)).EndInit();
            this.tabPage2.ResumeLayout(false);
            this.tabPage2.PerformLayout();
            ((System.ComponentModel.ISupportInitialize)(this.dataGridView2)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.remotePmuBindingSource)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.database1DataSet)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.localPmuBindingSource)).EndInit();
```

```
            this.tabPage3.ResumeLayout(false);
            ((System.ComponentModel.ISupportInitialize)(this.chart3)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.chart2)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.chart1)).EndInit();
            this.groupBox2.ResumeLayout(false);
            this.groupBox2.PerformLayout();
            this.groupBox3.ResumeLayout(false);
            this.groupBox3.PerformLayout();
            this.groupBox1.ResumeLayout(false);
            this.groupBox1.PerformLayout();
            this.menuStrip1.ResumeLayout(false);
            this.menuStrip1.PerformLayout();
            ((System.ComponentModel.ISupportInitialize)(this.database2DataSet)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.phasorDiffentialArchiveBindingSource)).EndInit();
            this.ResumeLayout(false);
            this.PerformLayout();

        }

        #endregion

        #region UI Components

        delegate void SetTextCallback(string text);
        private System.Windows.Forms.GroupBox ConSet;
        private System.Windows.Forms.TabControl tabControl1;
        private System.Windows.Forms.TabPage tabPage1;
        private System.Windows.Forms.TabPage tabPage2;
        private System.Windows.Forms.TabPage tabPage3;
        private System.Windows.Forms.GroupBox groupBox2;
        private System.Windows.Forms.ProgressBar progressBar1;
        private System.Windows.Forms.TabControl tabControl2;
        private System.Windows.Forms.TabPage tabPage4;
        private System.Windows.Forms.TabPage tabPage5;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Label time;
        private System.Windows.Forms.Label Frequency;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.GroupBox groupBox3;
        private System.Windows.Forms.Label label17;
        private System.Windows.Forms.Label label18;
        private System.Windows.Forms.Label ICmagR;
        private System.Windows.Forms.Label IBmagR;
        private System.Windows.Forms.Label IAmagR;
        private System.Windows.Forms.Label ICangR;
        private System.Windows.Forms.Label IBangR;
        private System.Windows.Forms.Label IAangR;
        private System.Windows.Forms.Label label25;
        private System.Windows.Forms.Label label26;
        private System.Windows.Forms.Label label27;
        private System.Windows.Forms.Label label28;
        private System.Windows.Forms.GroupBox groupBox1;
        private System.Windows.Forms.Label label10;
        private System.Windows.Forms.Label label9;
        private System.Windows.Forms.Label ICmag;
        private System.Windows.Forms.Label IBmag;
        private System.Windows.Forms.Label IAmag;
        private System.Windows.Forms.Label ICang;
        private System.Windows.Forms.Label IBang;
        private System.Windows.Forms.Label IAang;
        private System.Windows.Forms.Label label8;
        private System.Windows.Forms.Label label7;
        private System.Windows.Forms.Label IAmaglab;
        private System.Windows.Forms.Label label5;
        private System.Windows.Forms.TextBox locpmuport;
        private System.Windows.Forms.TextBox locpmuip;
        private System.Windows.Forms.TextBox locpmuid;
        private System.Windows.Forms.Label label29;
```

```
        private System.Windows.Forms.Label label31;
        private System.Windows.Forms.Label label30;
        private System.Windows.Forms.TextBox rempmuport;
        private System.Windows.Forms.TextBox rempmuip;
        private System.Windows.Forms.TextBox rempmuid;
        private System.Windows.Forms.Label label32;
        private System.Windows.Forms.Label label33;
        private System.Windows.Forms.Label label34;
        private System.Windows.Forms.MenuStrip menuStrip1;
        private System.Windows.Forms.ToolStripMenuItem fileToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem viewToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem editToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem optionsToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem helpToolStripMenuItem;
        private System.Windows.Forms.ComboBox comboBox1;
        private System.Windows.Forms.RadioButton radioButton3;
        private System.Windows.Forms.RadioButton radioButton2;
        private System.Windows.Forms.RadioButton radioButton1;
        private System.Windows.Forms.Button button2;
        private System.Windows.Forms.Button button1;
        private System.Windows.Forms.Timer timer1 = new System.Windows.Forms.Timer();
        private Label sysStatus;
        private TextBox sysDisplay;
        private DataGridView dataGridView1;
        private Database1DataSet database1DataSet;
        private BindingSource localPmuBindingSource;
        private Database1DataSetTableAdapters.localPmuTableAdapter localPmuTableAdapter;
        private Label label4;
        private Label label3;
        private DataGridView dataGridView2;
        private BindingSource remotePmuBindingSource;
        private Database1DataSetTableAdapters.remotePmuTableAdapter remotePmuTableAdapter;
        private System.Windows.Forms.DataVisualization.Charting.Chart chart1;
        private System.Windows.Forms.DataVisualization.Charting.Chart chart3;
        private System.Windows.Forms.DataVisualization.Charting.Chart chart2;
        private System.Windows.Forms.Timer clock;
        private Label label6;
        private Panel panel1;
        private DataGridView dataGridView3;
        private DataGridViewTextBoxColumn idDataGridViewTextBoxColumn1;
        private DataGridViewTextBoxColumn tsDataGridViewTextBoxColumn1;
        private DataGridViewTextBoxColumn iAangDataGridViewTextBoxColumn1;
        private DataGridViewTextBoxColumn iAmagDataGridViewTextBoxColumn1;
        private DataGridViewTextBoxColumn iBangDataGridViewTextBoxColumn1;
        private DataGridViewTextBoxColumn iBmagDataGridViewTextBoxColumn1;
        private DataGridViewTextBoxColumn iCangDataGridViewTextBoxColumn1;
        private DataGridViewTextBoxColumn iCmagDataGridViewTextBoxColumn1;
        private DataGridViewTextBoxColumn idDataGridViewTextBoxColumn;
        private DataGridViewTextBoxColumn tsDataGridViewTextBoxColumn;
        private DataGridViewTextBoxColumn iAangDataGridViewTextBoxColumn;
        private DataGridViewTextBoxColumn iAmagDataGridViewTextBoxColumn;
        private DataGridViewTextBoxColumn iBangDataGridViewTextBoxColumn;
        private DataGridViewTextBoxColumn iBmagDataGridViewTextBoxColumn;
        private DataGridViewTextBoxColumn iCangDataGridViewTextBoxColumn;
        private DataGridViewTextBoxColumn iCmagDataGridViewTextBoxColumn;
        private Database2DataSet database2DataSet;
        private BindingSource phasorDiffentialArchiveBindingSource;
        private Database2DataSetTableAdapters.phasorDiffentialArchiveTableAdapter
phasorDiffentialArchiveTableAdapter;
        private DataGridViewTextBoxColumn idDataGridViewTextBoxColumn2;
        private DataGridViewTextBoxColumn tDataGridViewTextBoxColumn;
        private DataGridViewTextBoxColumn differerentialMagnitudeDataGridViewTextBoxColumn;
        private DataGridViewTextBoxColumn differerentialAngleDataGridViewTextBoxColumn;
        private DataGridViewTextBoxColumn phasorDataGridViewTextBoxColumn;

        #endregion

        #region Component Model
```

```
        private System.ComponentModel.IContainer components = null;

        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #endregion

        #endregion

        #region cross thread workers

        #region phasor time domain plotting

        #region phasor plot variables

        DateTime startTime;

        double tval_ia, tval_Ria;
        double yval_ia, yval_Ria;

        double tval_ib, tval_Rib;
        double yval_ib, yval_Rib;

        double tval_ic, tval_Ric;
        double yval_ic, yval_Ric;

        int Valcount_ia = 0;
        int Valcount_ib = 0;
        int Valcount_ic = 0;

        int Valcount_Ria = 0;
        int Valcount_Rib = 0;
        int Valcount_Ric = 0;

        int minVal_ia = 0;
        int maxVal_ia = 20;

        int minVal_ib = 0;
        int maxVal_ib = 20;

        int minVal_ic = 0;
        int maxVal_ic = 20;

        int minVal_Ria = 0;
        int maxVal_Ria = 20;

        int minVal_Rib = 0;
        int maxVal_Rib = 20;

        int minVal_Ric = 0;
        int maxVal_Ric = 20;

        #endregion

        #region local IA phasor plot

        private void timeDomainPlot_ia(double t, double y)
        {
            tval_ia = t;
            yval_ia = y;

            new Thread(new ParameterizedThreadStart(worker_ia)).Start(new
Action<myData>(this.AddDataPoint_ia));
```

```
        }

        private void worker_ia(object obj)
        {
            var updateData = (Action<myData>)obj;
            updateData(new myData { X = tval_ia, Y = yval_ia });

        }

        public void AddDataPoint_ia(myData points)
        {
            if (this.InvokeRequired)
            {
                this.Invoke(new Action<myData>(AddDataPoint_ia), new object[] { points });
            }

            else
            {
                Valcount_ia++;

                if (Valcount_ia == maxVal_ia)
                {
                    minVal_ia = maxVal_ia;
                    maxVal_ia = maxVal_ia + 20;

                    chart1.ChartAreas[0].AxisX.Minimum = minVal_ia;
                    chart1.ChartAreas[0].AxisX.Maximum = maxVal_ia;
                }
                else
                {
                    chart1.ChartAreas[0].AxisX.Minimum = minVal_ia;
                    chart1.ChartAreas[0].AxisX.Maximum = maxVal_ia;
                }

                chart1.Series["loc  :IA"].Points.AddXY(points.X, points.Y);
            }
        }

        #endregion

        #region local IB phasor plot

        private void timeDomainPlot_ib(double t, double y)
        {
            tval_ib = t;
            yval_ib = y;

            new Thread(new ParameterizedThreadStart(worker_ib)).Start(new
Action<myData>(this.AddDataPoint_ib));
        }



        private void worker_ib(object obj)
        {
            var updateData = (Action<myData>)obj;
            updateData(new myData { X = tval_ib, Y = yval_ib });

        }


        public void AddDataPoint_ib(myData points)
        {
            if (this.InvokeRequired)
            {
                this.Invoke(new Action<myData>(AddDataPoint_ib), new object[] { points });
            }

            else
```

```
            {
                Valcount_ib++;

                if (Valcount_ib == maxVal_ib)
                {
                    minVal_ib = maxVal_ib;
                    maxVal_ib = maxVal_ib + 20;

                    chart2.ChartAreas[0].AxisX.Minimum = minVal_ib;
                    chart2.ChartAreas[0].AxisX.Maximum = maxVal_ib;
                }
                else
                {
                    chart2.ChartAreas[0].AxisX.Minimum = minVal_ib;
                    chart2.ChartAreas[0].AxisX.Maximum = maxVal_ib;
                }

                chart2.Series["loc  :IB"].Points.AddXY(points.X, points.Y);
            }
        }
        #endregion

        #region local IC phasor plot

        private void timeDomainPlot_ic(double t, double y)
        {
            tval_ic = t;
            yval_ic = y;

            new Thread(new ParameterizedThreadStart(worker_ic)).Start(new
Action<myData>(this.AddDataPoint_ic));
        }



        private void worker_ic(object obj)
        {
            var updateData = (Action<myData>)obj;
            updateData(new myData { X = tval_ic, Y = yval_ic });

        }


        public void AddDataPoint_ic(myData points)
        {
            if (this.InvokeRequired)
            {
                this.Invoke(new Action<myData>(AddDataPoint_ic), new object[] { points });
            }

            else
            {
                Valcount_ic++;

                if (Valcount_ic == maxVal_ic)
                {
                    minVal_ic = maxVal_ic;
                    maxVal_ic = maxVal_ic + 20;

                    chart3.ChartAreas[0].AxisX.Minimum = minVal_ic;
                    chart3.ChartAreas[0].AxisX.Maximum = maxVal_ic;
                }
                else
                {
                    chart3.ChartAreas[0].AxisX.Minimum = minVal_ic;
                    chart3.ChartAreas[0].AxisX.Maximum = maxVal_ic;
                }

                chart3.Series["loc  :IC"].Points.AddXY(points.X, points.Y);
```

```
            }
        }
        #endregion

        #region remote IA phasor plot

        private void timeDomainPlot_Ria(double t, double y)
        {
            tval_Ria = t;
            yval_Ria = y;

            new Thread(new ParameterizedThreadStart(worker_Ria)).Start(new
Action<myData>(this.AddDataPoint_Ria));
        }

        private void worker_Ria(object obj)
        {
            var updateData = (Action<myData>)obj;
            updateData(new myData { X = tval_Ria, Y = yval_Ria });

        }

        public void AddDataPoint_Ria(myData points)
        {
            if (this.InvokeRequired)
            {
                this.Invoke(new Action<myData>(AddDataPoint_Ria), new object[] { points });
            }

            else
            {
                Valcount_Ria++;

                if (Valcount_Ria == maxVal_Ria)
                {
                    minVal_Ria = maxVal_Ria;
                    maxVal_Ria = maxVal_Ria + 20;

                    chart1.ChartAreas[0].AxisX.Minimum = minVal_Ria;
                    chart1.ChartAreas[0].AxisX.Maximum = maxVal_Ria;
                }
                else
                {
                    chart1.ChartAreas[0].AxisX.Minimum = minVal_Ria;
                    chart1.ChartAreas[0].AxisX.Maximum = maxVal_Ria;
                }

                chart1.Series["rem :IA"].Points.AddXY(points.X, points.Y);
            }
        }

        #endregion

        #region remote IB phasor plot

        private void timeDomainPlot_Rib(double t, double y)
        {
            tval_Rib = t;
            yval_Rib = y;

            new Thread(new ParameterizedThreadStart(worker_Rib)).Start(new
Action<myData>(this.AddDataPoint_Rib));
        }

        private void worker_Rib(object obj)
        {
            var updateData = (Action<myData>)obj;
            updateData(new myData { X = tval_Rib, Y = yval_Rib });
```

```
        }

        public void AddDataPoint_Rib(myData points)
        {
            if (this.InvokeRequired)
            {
                this.Invoke(new Action<myData>(AddDataPoint_Rib), new object[] { points });
            }

            else
            {
                Valcount_Rib++;

                if (Valcount_Rib == maxVal_Rib)
                {
                    minVal_Rib = maxVal_Rib;
                    maxVal_Rib = maxVal_Rib + 20;

                    chart2.ChartAreas[0].AxisX.Minimum = minVal_Rib;
                    chart2.ChartAreas[0].AxisX.Maximum = maxVal_Rib;
                }
                else
                {
                    chart2.ChartAreas[0].AxisX.Minimum = minVal_Rib;
                    chart2.ChartAreas[0].AxisX.Maximum = maxVal_Rib;
                }

                chart2.Series["rem :IB"].Points.AddXY(points.X, points.Y);
            }
        }

        #endregion

        #region remote IC phasor plot

        private void timeDomainPlot_Ric(double t, double y)
        {
            tval_Ric = t;
            yval_Ric = y;

            new Thread(new ParameterizedThreadStart(worker_Ric)).Start(new
Action<myData>(this.AddDataPoint_Ric));
        }



        private void worker_Ric(object obj)
        {
            var updateData = (Action<myData>)obj;
            updateData(new myData { X = tval_Ric, Y = yval_Ric });

        }


        public void AddDataPoint_Ric(myData points)
        {
            if (this.InvokeRequired)
            {
                this.Invoke(new Action<myData>(AddDataPoint_Ric), new object[] { points });
            }

            else
            {
                Valcount_Ric++;

                if (Valcount_Ric == maxVal_Ric)
                {
                    minVal_Ric = maxVal_Ric;
                    maxVal_Ric = maxVal_Ric + 20;
```

```
                chart3.ChartAreas[0].AxisX.Minimum = minVal_Ric;
                chart3.ChartAreas[0].AxisX.Maximum = maxVal_Ric;
            }
            else
            {
                chart3.ChartAreas[0].AxisX.Minimum = minVal_Ric;
                chart3.ChartAreas[0].AxisX.Maximum = maxVal_Ric;
            }

            chart3.Series["rem :IC"].Points.AddXY(points.X, points.Y);
        }
    }

    #endregion

    #endregion

    #region real time Phasor Magnitude & Angle Display

    #region ui

    private void infoDisplay(string text)
    {
        if (this.sysDisplay.InvokeRequired)
        {
            SetTextCallback sc = new SetTextCallback(infoDisplay);
            this.Invoke(sc, new object[] { text });
        }
        else
        {
            sysDisplay.AppendText(Environment.NewLine);
            this.sysDisplay.Text = text;
        }
    }

    private void frequencyDisplay(string text)
    {
        if (this.Frequency.InvokeRequired)
        {
            SetTextCallback sc = new SetTextCallback(frequencyDisplay);
            this.Invoke(sc, new object[] { text });
        }
        else
        {
this.phasorDiffentialArchiveTableAdapter.Fill(this.database2DataSet.phasorDiffentialArchive);
            this.Frequency.Text = text;
        }
    }

    private void timeStampDisplay(string text)
    {
        if (this.time.InvokeRequired)
        {
            SetTextCallback sc = new SetTextCallback(timeStampDisplay);
            this.Invoke(sc, new object[] { text });
        }
        else { this.time.Text = text; }
    }

    #endregion

    #region local Pmu

    private void IaAngleDisplay(string text)
    {
        if (this.IAang.InvokeRequired)
        {
```

```
            SetTextCallback sc = new SetTextCallback(IaAngleDisplay);
            this.Invoke(sc, new object[] { text });
        }
        else
        {
            this.IAang.Text = text;
            this.localPmuTableAdapter.Fill(this.database1DataSet.localPmu);
        }
    }

    private void IaMagnitudeDisplay(string text)
    {
        if (this.IAmag.InvokeRequired)
        {
            SetTextCallback sc = new SetTextCallback(IaMagnitudeDisplay);
            this.Invoke(sc, new object[] { text });
        }
        else
        {
            this.IAmag.Text = text;
            this.localPmuTableAdapter.Fill(this.database1DataSet.localPmu);
        }
    }

    private void IbAngleDisplay(string text)
    {
        if (this.IBang.InvokeRequired)
        {
            SetTextCallback sc = new SetTextCallback(IbAngleDisplay);
            this.Invoke(sc, new object[] { text });
        }
        else
        {
            this.IBang.Text = text;
            this.localPmuTableAdapter.Fill(this.database1DataSet.localPmu);
        }
    }

    private void IbMagnitudeDisplay(string text)
    {
        if (this.IBmag.InvokeRequired)
        {
            SetTextCallback sc = new SetTextCallback(IbMagnitudeDisplay);
            this.Invoke(sc, new object[] { text });
        }
        else
        {
            this.IBmag.Text = text;
            this.localPmuTableAdapter.Fill(this.database1DataSet.localPmu);
        }
    }

    private void IcAngleDisplay(string text)
    {
        if (this.ICang.InvokeRequired)
        {
            SetTextCallback sc = new SetTextCallback(IcAngleDisplay);
            this.Invoke(sc, new object[] { text });
        }
        else
        {
            this.ICang.Text = text;
            this.localPmuTableAdapter.Fill(this.database1DataSet.localPmu);
        }
    }

    private void IcMagnitudeDisplay(string text)
    {
        if (this.ICmag.InvokeRequired)
```

```
    {
        SetTextCallback sc = new SetTextCallback(IcMagnitudeDisplay);
        this.Invoke(sc, new object[] { text });
    }
    else
    {
        this.ICmag.Text = text;
        this.localPmuTableAdapter.Fill(this.database1DataSet.localPmu);
    }
}

#endregion

#region remote Pmu

private void RIaAngleDisplay(string text)
{
    if (this.IAangR.InvokeRequired)
    {
        SetTextCallback sc = new SetTextCallback(RIaAngleDisplay);
        this.Invoke(sc, new object[] { text });
    }
    else
    {
        this.IAangR.Text = text;
        this.remotePmuTableAdapter.Fill(this.database1DataSet.remotePmu);
    }
}

private void RIaMagnitudeDisplay(string text)
{
    if (this.IAmagR.InvokeRequired)
    {
        SetTextCallback sc = new SetTextCallback(RIaMagnitudeDisplay);
        this.Invoke(sc, new object[] { text });
    }
    else
    {
        this.IAmagR.Text = text;
        this.remotePmuTableAdapter.Fill(this.database1DataSet.remotePmu);
    }
}

private void RIbAngleDisplay(string text)
{
    if (this.IBangR.InvokeRequired)
    {
        SetTextCallback sc = new SetTextCallback(RIbAngleDisplay);
        this.Invoke(sc, new object[] { text });
    }
    else
    {
        this.IBangR.Text = text;
        this.remotePmuTableAdapter.Fill(this.database1DataSet.remotePmu);
    }
}

private void RIbMagnitudeDisplay(string text)
{
    if (this.IBmagR.InvokeRequired)
    {
        SetTextCallback sc = new SetTextCallback(RIbMagnitudeDisplay);
        this.Invoke(sc, new object[] { text });
    }
    else
    {
        this.IBmagR.Text = text;
        this.remotePmuTableAdapter.Fill(this.database1DataSet.remotePmu);
    }
```

```
        }

        private void RIcAngleDisplay(string text)
        {
            if (this.ICangR.InvokeRequired)
            {
                SetTextCallback sc = new SetTextCallback(RIcAngleDisplay);
                this.Invoke(sc, new object[] { text });
            }
            else
            {
                this.ICangR.Text = text;
                this.remotePmuTableAdapter.Fill(this.database1DataSet.remotePmu);
            }
        }

        private void RIcMagnitudeDisplay(string text)
        {
            if (this.ICmagR.InvokeRequired)
            {
                SetTextCallback sc = new SetTextCallback(RIcMagnitudeDisplay);
                this.Invoke(sc, new object[] { text });
            }
            else
            {
                this.ICmagR.Text = text;
                this.remotePmuTableAdapter.Fill(this.database1DataSet.remotePmu);
            }
        }

        #endregion

        #endregion

        #endregion

        #endregion

        #region Local Phasor Protocol Parse

        #region Local  Phasor Objects

        private static long localPmuByteCount;
        private static long localPmuFrameCount;

        private static Concentrator localPmuConcentrator;
        private static StreamWriter localPmuM_exportFile;
        private static MultiProtocolFrameParser localPmuParser;

        private static readonly bool localPmuWriteLogs = false;
        private static readonly bool localPmuTestConcentrator = false;

        private static ConcurrentDictionary<string, IMeasurement> localPmuM_definedMeasurements;
        private static ConcurrentDictionary<ushort, ConfigurationCell> localPmuM_definedDevices;

        #endregion

        #region Local  Phasor Data Stream

        public void localPmuPhasorDataStream(string localpmuid, string localConnectionString)
        {
            localPmuM_definedDevices = new ConcurrentDictionary<ushort, ConfigurationCell>();
            localPmuM_definedMeasurements = new ConcurrentDictionary<string, IMeasurement>();
            localPmuM_definedDevices = new ConcurrentDictionary<ushort, ConfigurationCell>();
            if (localPmuWriteLogs)
                localPmuM_exportFile = new
StreamWriter(FilePath.GetAbsolutePath("InputTimestamps.csv"));

            if (localPmuTestConcentrator)
```

```
            {
                localPmuConcentrator = new Concentrator(localPmuWriteLogs,
FilePath.GetAbsolutePath("OutputTimestamps.csv"));
                localPmuConcentrator.TimeResolution = 333000;
                localPmuConcentrator.FramesPerSecond = 30;
                localPmuConcentrator.LagTime = 3.0D;
                localPmuConcentrator.LeadTime = 9.0D;
                localPmuConcentrator.PerformTimestampReasonabilityCheck = false;
                localPmuConcentrator.ProcessByReceivedTimestamp = true;
                localPmuConcentrator.Start();
            }
            localPmuParser = new MultiProtocolFrameParser();

            #region parser event handlers

            localPmuParser.AllowedParsingExceptions = 500;
            localPmuParser.ParsingExceptionWindow = 5;
            localPmuParser.ConnectionAttempt += localPmuParser_ConnectionAttempt;
            localPmuParser.ConnectionEstablished += localPmuParser_ConnectionEstablished;
            localPmuParser.ConnectionException += localPmuParser_ConnectionException;
            localPmuParser.ParsingException += localPmuParser_ParsingException;
            localPmuParser.ReceivedConfigurationFrame +=
localPmuParser_ReceivedConfigurationFrame;
            localPmuParser.ReceivedDataFrame += parser_ReceivedDataFrame;
            localPmuParser.ReceivedFrameBufferImage += localPmuParser_ReceivedFrameBufferImage;
            localPmuParser.ConnectionTerminated += localPmuParser_ConnectionTerminated;

            #endregion

            localPmuParser.ConnectionString = @"phasorProtocol=IEEEC37_118V1;
transportProtocol=tcp; accessID=" + localpmuid + "; server=" + localConnectionString + ";
interface=0.0.0.0; isListener=false";

            Dictionary<string, string> localPmuSettings =
localPmuParser.ConnectionString.ParseKeyValuePairs();

            #region local & remote pmu settings

            string setting;

            if (localPmuSettings.TryGetValue("accessID", out setting)) localPmuParser.DeviceID =
ushort.Parse(setting);

            if (localPmuSettings.TryGetValue("simulateTimestamp", out setting))
localPmuParser.InjectSimulatedTimestamp = setting.ParseBoolean();

            if (localPmuSettings.TryGetValue("allowedParsingExceptions", out setting))
localPmuParser.AllowedParsingExceptions = int.Parse(setting);

            if (localPmuSettings.TryGetValue("parsingExceptionWindow", out setting))
localPmuParser.ParsingExceptionWindow = Ticks.FromSeconds(double.Parse(setting));

            if (localPmuSettings.TryGetValue("autoStartDataParsingSequence", out setting))
localPmuParser.AutoStartDataParsingSequence = setting.ParseBoolean();

            if (localPmuSettings.TryGetValue("skipDisableRealTimeData", out setting))
localPmuParser.SkipDisableRealTimeData = setting.ParseBoolean();

            #endregion

            localPmuParser.AutoRepeatCapturedPlayback = true;
            localPmuParser.AutoStartDataParsingSequence = true;
            localPmuParser.Start();

            if (localPmuTestConcentrator) localPmuConcentrator.Stop();
            if (localPmuWriteLogs) localPmuM_exportFile.Close();
        }

        #endregion
```

```
        #region Local  Phasor Data Stream Parse

        private void parser_ReceivedDataFrame(object sender, EventArgs<IDataFrame> e)
        {
            localPmuDataframe(sender, e);
        }

        #endregion

        #region Local  Phasor Data Parsing Methods

        #region Parser .Method >> parsing exception

        private static void localPmuParser_ParsingException(object sender, EventArgs<Exception> e)
        {
            MessageBox.Show(e.Argument.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }

        #endregion

        #region Parser .Method >> connection expeption

        private void localPmuParser_ConnectionException(object sender, EventArgs<Exception, int>
e)
        {
            MessageBox.Show(e.Argument1.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }

        #endregion

        #region Parser .Method >> attempting connection

        private static void localPmuParser_ConnectionAttempt(object sender, EventArgs e)
        {
        }

        #endregion

        #region Parser .Method >> terminating connection

        static void localPmuParser_ConnectionTerminated(object sender, EventArgs e)
        {
            MessageBox.Show("Connection terminated.", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }

        #endregion

        #region Parser .Method >> establishing connection

        private void localPmuParser_ConnectionEstablished(object sender, EventArgs e)
        {
            //string connected = "Connected";
            //MessageBox.Show(connected, "Message", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        }

        #endregion

        #region Parser .Method >> recieving frame buffer image

        static void localPmuParser_ReceivedFrameBufferImage(object sender,
EventArgs<FundamentalFrameType, byte[], int, int> e)
        {
            const int interval = (int)SI.Kilo * 2;
```

```
            bool showMessage = localPmuByteCount + e.Argument4 >= (localPmuByteCount / interval +
1) * interval;
            localPmuByteCount += e.Argument4;
        }

        #endregion

        #region Parser .Method >> recieving configuration frame

        private static void localPmuParser_ReceivedConfigurationFrame(object sender,
EventArgs<IConfigurationFrame> e)
        {
        }

        #endregion

        #endregion

        #region Local Phasor Data-Frame reading Method

        private void localPmuDataframe(object sender, EventArgs<IDataFrame> e)
        {
            #region method members

            double t;       //time domain (t)
            double y;       //time domain function
            string globalTimeStamp = System.String.Empty;

            string localPhasorAngle_IA = System.String.Empty;
            string localPhasorAngle_IB = System.String.Empty;
            string localPhasorAngle_IC = System.String.Empty;
            string localPhasorMagnitude_IA = System.String.Empty;
            string localPhasorMagnitude_IB = System.String.Empty;
            string localPhasorMagnitude_IC = System.String.Empty;

            IDataFrame dataFrame = e.Argument;
            SqlConnection dataBaseConnection = new
SqlConnection(global::pmuLive.Properties.Settings.Default.Database1ConnectionString);

            #endregion

            localPmuFrameCount++;

            if (dataFrame.Cells.Count > 0)
            {

                IDataCell device = dataFrame.Cells[0];
                globalTimeStamp = device.GetStatusFlagsMeasurement().Timestamp.ToString("yyyy-MM-
dd HH:mm:ss.fff").ToString();

                timeStampDisplay(globalTimeStamp);
                frequencyDisplay(device.FrequencyValue.Frequency.ToString());

                #region local pmu data stream

                for (int x = 0; x < device.PhasorValues.Count; x++)
                {
                    #region ia phasor

                    if (x == 3)
                    {
                        #region phasor [t]

                        DateTime currentTime = DateTime.Parse(globalTimeStamp);
                        localPmuMarkStarttime(currentTime);
                        System.TimeSpan timedifference = currentTime.Subtract(startTime);
                        t = timedifference.TotalSeconds;

                        #endregion
```

```
                        #region phasor data Read

                        localPhasorAngle_IA = device.PhasorValues[x].Angle.ToString();
                        localPhasorMagnitude_IA = device.PhasorValues[x].Magnitude.ToString();

                        IaAngleDisplay(localPhasorAngle_IA);
                        IaMagnitudeDisplay(localPhasorMagnitude_IA);

                        #endregion

                        #region phasor plot function (time domain)

                        y = plotFunction(device.FrequencyValue.Frequency,
device.PhasorValues[x].Angle, device.PhasorValues[x].Magnitude, t);
                        timeDomainPlot_ia(t, y);

                        #endregion
                    }

                    #endregion

                    #region ib phasor

                    if (x == 4)
                    {
                        #region phasor [t]

                        DateTime currentTime = DateTime.Parse(globalTimeStamp);
                        localPmuMarkStarttime(currentTime);
                        System.TimeSpan timedifference = currentTime.Subtract(startTime);
                        t = timedifference.TotalSeconds;

                        #endregion

                        #region phasor data Read

                        localPhasorAngle_IB = device.PhasorValues[x].Angle.ToString();
                        localPhasorMagnitude_IB = device.PhasorValues[x].Magnitude.ToString();

                        IbAngleDisplay(localPhasorAngle_IB);
                        IbMagnitudeDisplay(localPhasorMagnitude_IB);

                        #endregion

                        #region phasor plot function (time domain)

                        y = plotFunction(device.FrequencyValue.Frequency,
device.PhasorValues[x].Angle, device.PhasorValues[x].Magnitude, t);
                        timeDomainPlot_ib(t, y);

                        #endregion
                    }

                    #endregion

                    #region ic phasor

                    if (x == 5)
                    {
                        #region phasor [t]

                        DateTime currentTime = DateTime.Parse(globalTimeStamp);
                        localPmuMarkStarttime(currentTime);
                        System.TimeSpan timedifference = currentTime.Subtract(startTime);
                        t = timedifference.TotalSeconds;

                        #endregion
```

```
                        #region phasor data Read

                        localPhasorAngle_IC = device.PhasorValues[x].Angle.ToString();
                        localPhasorMagnitude_IC = device.PhasorValues[x].Magnitude.ToString();

                        IcAngleDisplay(localPhasorAngle_IC);
                        IcMagnitudeDisplay(localPhasorMagnitude_IC);

                        #endregion

                        #region phasor plot function (time domain)

                        y = plotFunction(device.FrequencyValue.Frequency,
device.PhasorValues[x].Angle, device.PhasorValues[x].Magnitude, t);
                        timeDomainPlot_ic(t, y);

                        #endregion

                        #region phasor data stream, data-base archive

                        try
                        {
                            string databaseString = "INSERT INTO localPmu (ts, IAmag, IAang,IBmag,
IBang,ICmag, ICang) values('" + globalTimeStamp + "','" + localPhasorMagnitude_IA + "','" +
localPhasorAngle_IA + "','" + localPhasorMagnitude_IB + "','" + localPhasorAngle_IB + "','" +
localPhasorMagnitude_IC + "','" + localPhasorAngle_IC + "' )";
                            SqlCommand exeSql = new SqlCommand(databaseString,
dataBaseConnection);

                            dataBaseConnection.Open();
                            exeSql.ExecuteNonQuery();

                            dataBaseConnection.Close();
                        }
                        catch (Exception ex)
                        {
                            MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                        }

                        #endregion

                    }

                    #endregion
                }

                #endregion

                Thread.Sleep(100);
                computeCurrentDifferential.setTime(globalTimeStamp);
            }
            Thread.Sleep(10);
            computeCurrentDifferential.localCurrent(Convert.ToDouble(localPhasorMagnitude_IA),
Convert.ToDouble(localPhasorAngle_IA), Convert.ToDouble(localPhasorMagnitude_IB),
Convert.ToDouble(localPhasorAngle_IB), Convert.ToDouble(localPhasorMagnitude_IC),
Convert.ToDouble(localPhasorAngle_IC));
        }

        #endregion

        #endregion

        #region Remote Phasor Protocol Parse

        #region Remote  Phasor Objects

        private static long remotePmuByteCount;
        private static long remotePmuFrameCount;
```

```csharp
        private static Concentrator remotePmuConcentrator;
        private static StreamWriter remotePmuM_exportFile;
        private static MultiProtocolFrameParser remotePmuParser;

        private static readonly bool remotePmuWriteLogs = false;
        private static readonly bool remotePmuTestConcentrator = false;

        private static ConcurrentDictionary<string, IMeasurement> remotePmuM_definedMeasurements;
        private static ConcurrentDictionary<ushort, ConfigurationCell> remotePmuM_definedDevices;

        #endregion

        #region Remote  Phasor Data Stream

        public void remotePmuPhasorDataStream(string remotepmuid, string remoteConnectionString)
        {
            remotePmuM_definedDevices = new ConcurrentDictionary<ushort, ConfigurationCell>();
            remotePmuM_definedMeasurements = new ConcurrentDictionary<string, IMeasurement>();
            remotePmuM_definedDevices = new ConcurrentDictionary<ushort, ConfigurationCell>();
            if (remotePmuWriteLogs)
                remotePmuM_exportFile = new
StreamWriter(FilePath.GetAbsolutePath("InputTimestamps.csv"));

            if (remotePmuTestConcentrator)
            {
                remotePmuConcentrator = new Concentrator(remotePmuWriteLogs,
FilePath.GetAbsolutePath("OutputTimestamps.csv"));
                remotePmuConcentrator.TimeResolution = 333000;
                remotePmuConcentrator.FramesPerSecond = 30;
                remotePmuConcentrator.LagTime = 3.0D;
                remotePmuConcentrator.LeadTime = 9.0D;
                remotePmuConcentrator.PerformTimestampReasonabilityCheck = false;
                remotePmuConcentrator.ProcessByReceivedTimestamp = true;
                remotePmuConcentrator.Start();
            }
            remotePmuParser = new MultiProtocolFrameParser();

            #region parser event handlers

            remotePmuParser.AllowedParsingExceptions = 500;
            remotePmuParser.ParsingExceptionWindow = 5;
            remotePmuParser.ConnectionAttempt += remotePmuParser_ConnectionAttempt;
            remotePmuParser.ConnectionEstablished += remotePmuParser_ConnectionEstablished;
            remotePmuParser.ConnectionException += remotePmuParser_ConnectionException;
            remotePmuParser.ParsingException += remotePmuParser_ParsingException;
            remotePmuParser.ReceivedConfigurationFrame +=
remotePmuParser_ReceivedConfigurationFrame;
            remotePmuParser.ReceivedDataFrame += remotePmuParser_ReceivedDataFrame;
            remotePmuParser.ReceivedFrameBufferImage += remotePmuParser_ReceivedFrameBufferImage;
            remotePmuParser.ConnectionTerminated += remotePmuParser_ConnectionTerminated;

            #endregion

            remotePmuParser.ConnectionString = @"phasorProtocol=IEEEC37_118V1;
transportProtocol=tcp; accessID=" + remotepmuid + "; server=" + remoteConnectionString + ";
interface=0.0.0.0; isListener=false";

            Dictionary<string, string> remotePmuSettings =
remotePmuParser.ConnectionString.ParseKeyValuePairs();

            #region local & remote pmu settings

            string setting;

            if (remotePmuSettings.TryGetValue("accessID", out setting)) remotePmuParser.DeviceID =
ushort.Parse(setting);
```

```
            if (remotePmuSettings.TryGetValue("simulateTimestamp", out setting))
remotePmuParser.InjectSimulatedTimestamp = setting.ParseBoolean();

            if (remotePmuSettings.TryGetValue("allowedParsingExceptions", out setting))
remotePmuParser.AllowedParsingExceptions = int.Parse(setting);

            if (remotePmuSettings.TryGetValue("parsingExceptionWindow", out setting))
remotePmuParser.ParsingExceptionWindow = Ticks.FromSeconds(double.Parse(setting));

            if (remotePmuSettings.TryGetValue("autoStartDataParsingSequence", out setting))
remotePmuParser.AutoStartDataParsingSequence = setting.ParseBoolean();

            if (remotePmuSettings.TryGetValue("skipDisableRealTimeData", out setting))
remotePmuParser.SkipDisableRealTimeData = setting.ParseBoolean();

            #endregion

            remotePmuParser.AutoRepeatCapturedPlayback = true;
            remotePmuParser.AutoStartDataParsingSequence = true;
            remotePmuParser.Start();

            if (remotePmuTestConcentrator) remotePmuConcentrator.Stop();
            if (remotePmuWriteLogs) remotePmuM_exportFile.Close();
        }

        #endregion

        #region Remote  Phasor Data Stream Parse

        private void remotePmuParser_ReceivedDataFrame(object sender, EventArgs<IDataFrame> e)
        {
            remotePmuDataframe(sender, e);
        }

        #endregion

        #region Remote  Phasor Data Parsing Methods

        #region Parser .Method >> parsing exception

        private static void remotePmuParser_ParsingException(object sender, EventArgs<Exception>
e)
        {
            MessageBox.Show(e.Argument.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }

        #endregion

        #region Parser .Method >> connection expeption

        private void remotePmuParser_ConnectionException(object sender, EventArgs<Exception, int>
e)
        {
            MessageBox.Show(e.Argument1.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }

        #endregion

        #region Parser .Method >> attempting connection

        private static void remotePmuParser_ConnectionAttempt(object sender, EventArgs e)
        {
        }

        #endregion

        #region Parser .Method >> terminating connection
```

```
        static void remotePmuParser_ConnectionTerminated(object sender, EventArgs e)
        {
            MessageBox.Show("Connection terminated.", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }

        #endregion

        #region Parser .Method >> establishing connection

        private void remotePmuParser_ConnectionEstablished(object sender, EventArgs e)
        {
            //string connected = "Connected";
            //MessageBox.Show(connected, "Message", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        }

        #endregion

        #region Parser .Method >> recieving frame buffer image

        static void remotePmuParser_ReceivedFrameBufferImage(object sender,
EventArgs<FundamentalFrameType, byte[], int, int> e)
        {
            const int interval = (int)SI.Kilo * 2;
            bool showMessage = remotePmuByteCount + e.Argument4 >= (remotePmuByteCount / interval
+ 1) * interval;
            remotePmuByteCount += e.Argument4;
        }

        #endregion

        #region Parser .Method >> recieving configuration frame

        private static void remotePmuParser_ReceivedConfigurationFrame(object sender,
EventArgs<IConfigurationFrame> e)
        {
        }

        #endregion

        #endregion

        #region Remote Phasor Data-Frame reading Method

        private void remotePmuDataframe(object sender, EventArgs<IDataFrame> e)
        {
            #region method members
            double t;       //time domain (t)
            double y;       //time domain function
            string globalTimeStamp = System.String.Empty;

            string remotePhasorAngle_IA = System.String.Empty;
            string remotePhasorAngle_IB = System.String.Empty;
            string remotePhasorAngle_IC = System.String.Empty;
            string remotePhasorMagnitude_IA = System.String.Empty;
            string remotePhasorMagnitude_IB = System.String.Empty;
            string remotePhasorMagnitude_IC = System.String.Empty;

            IDataFrame dataFrame = e.Argument;
            SqlConnection dataBaseConnection = new
SqlConnection(global::pmuLive.Properties.Settings.Default.Database1ConnectionString);

            #endregion

            localPmuFrameCount++;

            if (dataFrame.Cells.Count > 0)
```

```
            {

                IDataCell device = dataFrame.Cells[0];
                globalTimeStamp = device.GetStatusFlagsMeasurement().Timestamp.ToString("yyyy-MM-
dd HH:mm:ss.fff").ToString();

                timeStampDisplay(globalTimeStamp);
                frequencyDisplay(device.FrequencyValue.Frequency.ToString());

                #region remote pmu data stream

                for (int x = 0; x < device.PhasorValues.Count; x++)
                {
                    #region ia phasor

                    if (x == 3)
                    {
                        #region phasor [t]

                        DateTime currentTime = DateTime.Parse(globalTimeStamp);
                        remotePmuMarkStarttime(currentTime);
                        System.TimeSpan timedifference = currentTime.Subtract(startTime);
                        t = timedifference.TotalSeconds;

                        #endregion

                        #region phasor data Read

                        remotePhasorAngle_IA = device.PhasorValues[x].Angle.ToString();
                        remotePhasorMagnitude_IA = device.PhasorValues[x].Magnitude.ToString();

                        RIaAngleDisplay(remotePhasorAngle_IA);
                        RIaMagnitudeDisplay(remotePhasorMagnitude_IA);

                        #endregion

                        #region phasor plot function (time domain)

                        y = plotFunction(device.FrequencyValue.Frequency,
device.PhasorValues[x].Angle, device.PhasorValues[x].Magnitude, t);
                        timeDomainPlot_Ria(t, y);

                        #endregion
                    }

                    #endregion

                    #region ib phasor

                    if (x == 4)
                    {
                        #region phasor [t]

                        DateTime currentTime = DateTime.Parse(globalTimeStamp);
                        remotePmuMarkStarttime(currentTime);
                        System.TimeSpan timedifference = currentTime.Subtract(startTime);
                        t = timedifference.TotalSeconds;

                        #endregion

                        #region phasor data Read

                        remotePhasorAngle_IB = device.PhasorValues[x].Angle.ToString();
                        remotePhasorMagnitude_IB = device.PhasorValues[x].Magnitude.ToString();

                        RIbAngleDisplay(remotePhasorAngle_IB);
                        RIbMagnitudeDisplay(remotePhasorMagnitude_IB);

                        #endregion
```

```
                        #region phasor plot function (time domain)

                        y = plotFunction(device.FrequencyValue.Frequency,
device.PhasorValues[x].Angle, device.PhasorValues[x].Magnitude, t);
                        timeDomainPlot_Rib(t, y);

                        #endregion
                    }
                    #endregion

                    #region ic phasor

                    if (x == 5)
                    {

                        #region phasor [t]

                        DateTime currentTime = DateTime.Parse(globalTimeStamp);
                        remotePmuMarkStarttime(currentTime);
                        System.TimeSpan timedifference = currentTime.Subtract(startTime);
                        t = timedifference.TotalSeconds;

                        #endregion

                        #region phasor data Read

                        remotePhasorAngle_IC = device.PhasorValues[x].Angle.ToString();
                        remotePhasorMagnitude_IC = device.PhasorValues[x].Magnitude.ToString();

                        RIcAngleDisplay(remotePhasorAngle_IC);
                        RIcMagnitudeDisplay(remotePhasorMagnitude_IC);

                        #endregion

                        #region phasor plot function (time domain)

                        y = plotFunction(device.FrequencyValue.Frequency,
device.PhasorValues[x].Angle, device.PhasorValues[x].Magnitude, t);
                        timeDomainPlot_Ric(t, y);

                        #endregion

                        #region phasor data stream, data-base archive

                        try
                        {
                            string databaseString = "INSERT INTO remotePmu (ts, IAmag,
IAang,IBmag, IBang,ICmag, ICang) values('" + globalTimeStamp + "','" + remotePhasorMagnitude_IA +
"','" + remotePhasorAngle_IA + "','" + remotePhasorMagnitude_IB + "','" + remotePhasorAngle_IB +
"','" + remotePhasorMagnitude_IC + "','" + remotePhasorAngle_IC + "' )";
                            SqlCommand exeSql = new SqlCommand(databaseString,
dataBaseConnection);

                            dataBaseConnection.Open();
                            exeSql.ExecuteNonQuery();

                            dataBaseConnection.Close();
                        }
                        catch (Exception ex)
                        {
                            MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                        }
                        #endregion

                    }
                    #endregion
                }
```

```
                #endregion

                Thread.Sleep(100);
                computeCurrentDifferential.setTime(globalTimeStamp);
            }
            Thread.Sleep(10);
            computeCurrentDifferential.remoteCurrent(Convert.ToDouble(remotePhasorMagnitude_IA),
Convert.ToDouble(remotePhasorAngle_IA), Convert.ToDouble(remotePhasorMagnitude_IB),
Convert.ToDouble(remotePhasorAngle_IB), Convert.ToDouble(remotePhasorMagnitude_IC),
Convert.ToDouble(remotePhasorAngle_IC));
        }

        #endregion

        #endregion

        #region Local & Remote Phasor Plotting Methods

        public double plotFunction(double frequency, double phasorAngle, double phasorMagnitude,
double time)
        {
            double yValue;
            double phasorAngularVelocity;

            phasorAngularVelocity = 2 * Math.PI * frequency;
            yValue = phasorMagnitude * Math.Cos((phasorAngularVelocity * time) + phasorAngle);

            return yValue;
        }

        private void localPmuMarkStarttime(DateTime currentTime)
        {
            bool systemStartTimeStampFlag = false;

            if (systemStartTimeStampFlag == false)
            {
                startTime = currentTime;
                systemStartTimeStampFlag = true;
            }
        }

        private void remotePmuMarkStarttime(DateTime currentTime)
        {
            bool systemStartTimeStampFlag = false;

            if (systemStartTimeStampFlag == false)
            {
                startTime = currentTime;
                systemStartTimeStampFlag = true;
```

A.3 Class: PCDAA.cs

```
using System;
using System.Data;
using System.Linq;
using System.Text;
using System.Drawing;
using System.Threading;
using System.Windows.Forms;
using System.ComponentModel;
using System.Threading.Tasks;
using System.Collections.Generic;

namespace pmuLive
{
    public partial class pmuLive : Form
    {
```

```
        static AutoResetEvent localThread = new AutoResetEvent(false);
        static AutoResetEvent remoteThread = new AutoResetEvent(false);

        public pmuLive()
        {
            InitializeComponent();
        }

        private void clock_Tick(object sender, EventArgs e)
        {
            progressBar();
        }

        private void pmuLive_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
'database2DataSet.phasorDiffentialArchive' table. You can move, or remove it, as needed.

this.phasorDiffentialArchiveTableAdapter.Fill(this.database2DataSet.phasorDiffentialArchive);
            this.remotePmuTableAdapter.Fill(this.database1DataSet.remotePmu);
            this.localPmuTableAdapter.Fill(this.database1DataSet.localPmu);
        }

        private void standardSelect(object sender, EventArgs e)
        {

        }

        private void DisconnectBttn(object sender, EventArgs e)
        {
            stopParse();
        }

        private void ConnectionButtn(object sender, EventArgs e)
        {
            clock.Start();

            string localpmuid = locpmuid.Text;
            string remotepmuid = rempmuid.Text;
            string localpmuconnString = locpmuip.Text + ":" + locpmuport.Text;
            string remotepmuconnString = rempmuip.Text + ":" + rempmuport.Text;

            showUserDefinedConnectionParameters();
            startThreadManager(localpmuid, localpmuconnString, remotepmuid, remotepmuconnString);
        }

        private void sysStatus_Click(object sender, EventArgs e)
        {
        }

    }
}
```

A.4 Class: Concentrator.cs

```
using System;
using System.IO;
using System.Linq;
using System.Text;
using GSF.TimeSeries;
using System.Threading.Tasks;
using System.Collections.Generic;

namespace pmuLive
{
    class Concentrator : ConcentratorBase
    {
        private readonly StreamWriter m_exportFile;
```

```
        private readonly bool m_writeLogs;

        public Concentrator(bool writeLogs, string exportFileName)
        {
            m_writeLogs = writeLogs;

            if (m_writeLogs)
                m_exportFile = new StreamWriter(exportFileName);
        }

        public Concentrator()
        {
        }

        public override void Stop()
        {
            base.Stop();

            if (m_exportFile != null)
                m_exportFile.Close();
        }

        protected override void PublishFrame(IFrame frame, int index)
        {
            if (m_writeLogs)
                m_exportFile.WriteLine(frame.Timestamp.ToString("yyyy-MM-dd HH:mm:ss.fff") +
string.Concat(frame.Measurements.Values.Select(measurement => "," +
measurement.AdjustedValue.ToString())));
        }
    }
}
```

A.5 Class: myData.cs

```
using System;
using System.Linq;
using System.Text;
using System.Collections.Generic;

namespace pmuLive
{
    public class myData
    {
        public double X;
        public double Y;
    }
}
```

Appendix B:

CSAEMS Test-bench, configuration

B.1 Communication network configuration

This section details setup and network configuration of the CSAEMS Test-bench communication network.

B1.1 Control station configuration

Net-work configuration of the control station computer is detailed here. Assignment of static network is illustrates in Figure B.1 and Figure B.2.

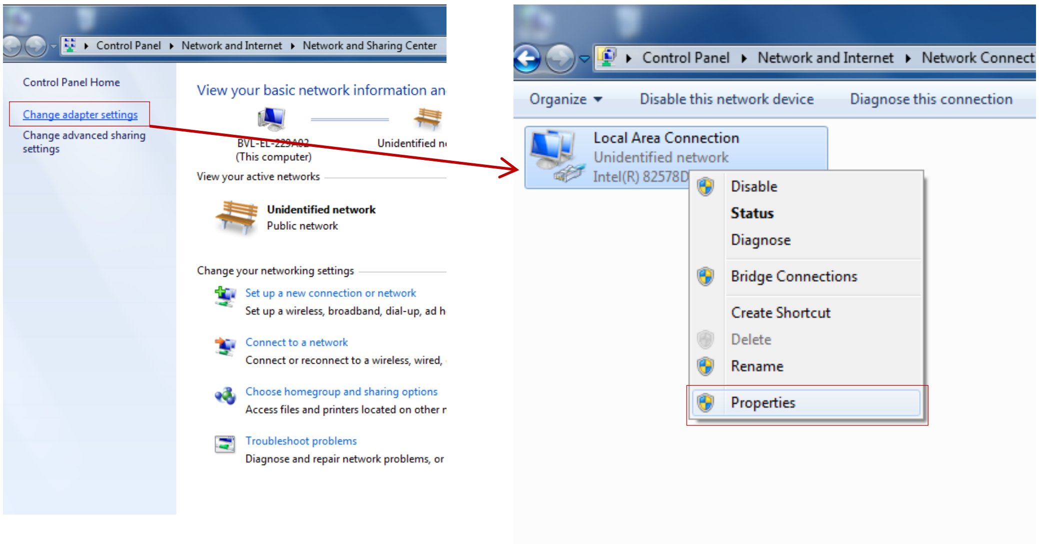


Figure B.1: Adapter settings, Local area connection properties

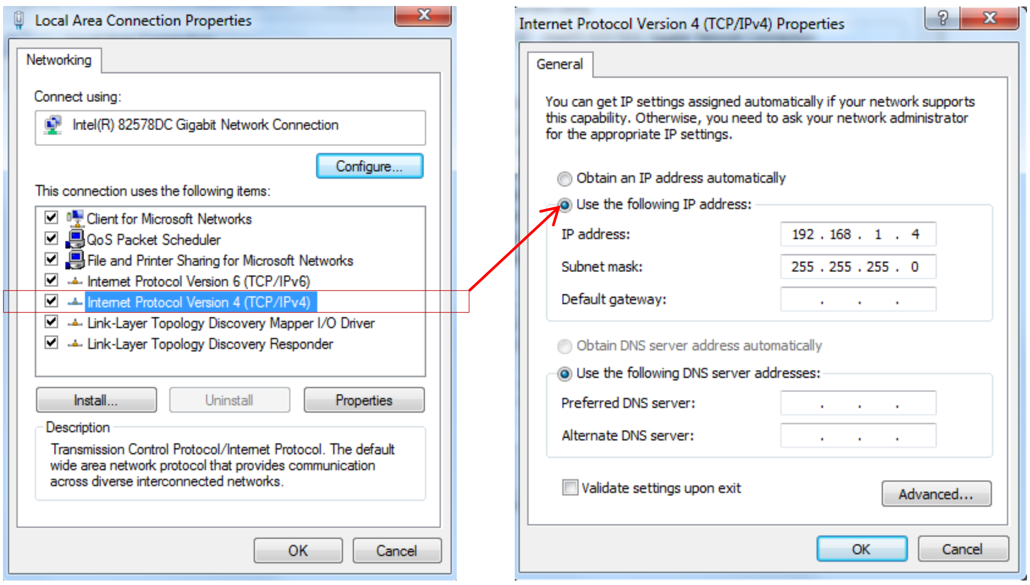
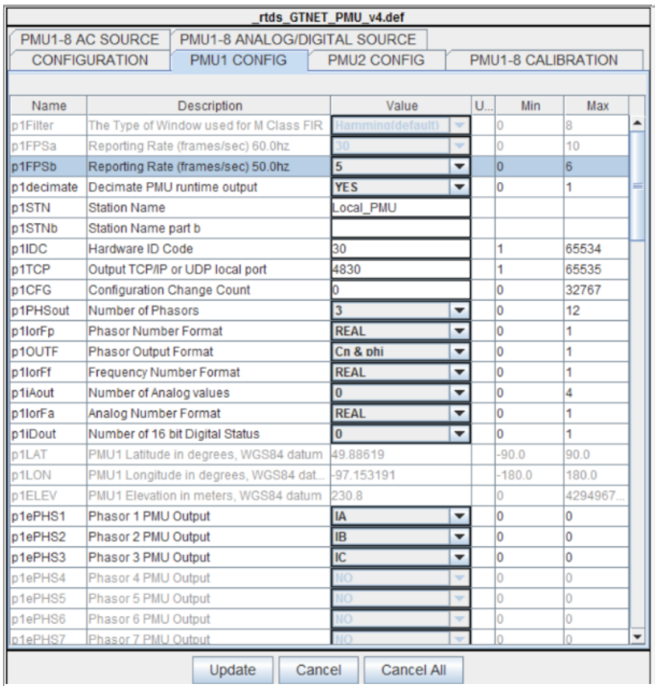


Figure B.2: Static address assignment

B1.2 RSCAD configuration

The Test-bench control station is assigned a static address of 192.168.1.4. This enables to be a part of the CSAEMS RTDS closed network. The two RTDS GTNET cards have the static addresses 192.168.1.202 and 192.168.1.204 but connection to them still needs to be established through specifying the port and ID specifications of the PMUs using the RSCAD interface. Figure B.3 and Figure B.4 illustrate port and ID specification for the local and remote terminal PMUs in the real-time RTDS simulation

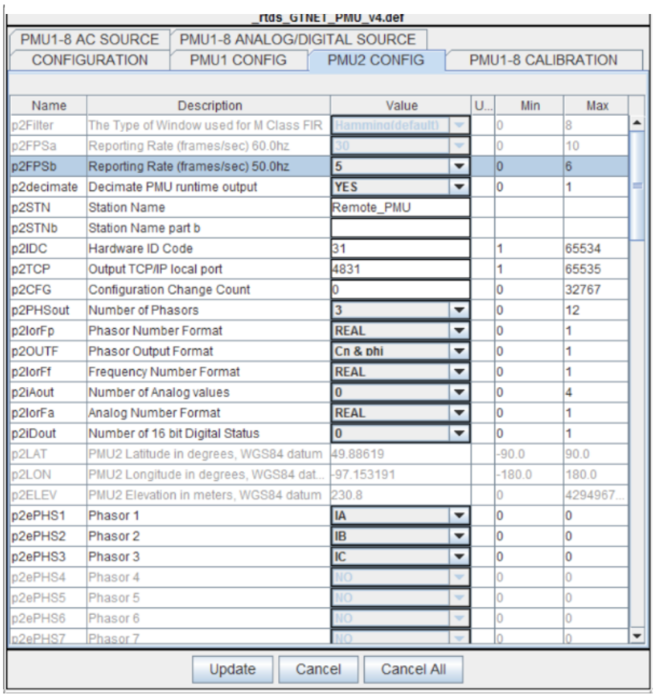

_rtds_GTNET_PMU_v4.def

PMU1-8 AC SOURCE | PMU1-8 ANALOG/DIGITAL SOURCE | CONFIGURATION | PMU1 CONFIG | PMU2 CONFIG | PMU1-8 CALIBRATION

Name	Description	Value	U...	Min	Max
p1Filter	The Type of Window used for M Class FIR	Hamming(default)		0	8
p1FPSa	Reporting Rate (frames/sec) 60.0hz	30		0	10
p1FPSb	Reporting Rate (frames/sec) 50.0hz	5		0	6
p1decimate	Decimate PMU runtime output	YES		0	1
p1STN	Station Name	Local_PMU			
p1STNb	Station Name part b				
p1IDC	Hardware ID Code	30		1	65534
p1TCP	Output TCP/IP or UDP local port	4830		1	65535
p1CFG	Configuration Change Count	0		0	32767
p1PHSout	Number of Phasors	3		0	12
p1lorFp	Phasor Number Format	REAL		0	1
p1OUTF	Phasor Output Format	Cn & phi		0	1
p1lorFf	Frequency Number Format	REAL		0	1
p1iAout	Number of Analog values	0		0	4
p1lorFa	Analog Number Format	REAL		0	1
p1iDout	Number of 16 bit Digital Status	0		0	1
p1LAT	PMU1 Latitude in degrees, WGS84 datum	49.88619		-90.0	90.0
p1LON	PMU1 Longitude in degrees, WGS84 dat...	-97.153191		-180.0	180.0
p1ELEV	PMU1 Elevation in meters, WGS84 datum	230.8		0	4294967...
p1ePHS1	Phasor 1 PMU Output	IA		0	0
p1ePHS2	Phasor 2 PMU Output	IB		0	0
p1ePHS3	Phasor 3 PMU Output	IC		0	0
p1ePHS4	Phasor 4 PMU Output	NO		0	0
p1ePHS5	Phasor 5 PMU Output	NO		0	0
p1ePHS6	Phasor 6 PMU Output	NO		0	0
p1ePHS7	Phasor 7 PMU Output	NO		0	0

Update | Cancel | Cancel All

Figure B.1: Local PMU specifications


_rtds_GTNET_PMU_v4.def

PMU1-8 AC SOURCE | PMU1-8 ANALOG/DIGITAL SOURCE | CONFIGURATION | PMU1 CONFIG | PMU2 CONFIG | PMU1-8 CALIBRATION

Name	Description	Value	U...	Min	Max
p2Filter	The Type of Window used for M Class FIR	Hamming(default)		0	8
p2FPSa	Reporting Rate (frames/sec) 60.0hz	30		0	10
p2FPSb	Reporting Rate (frames/sec) 50.0hz	5		0	6
p2decimate	Decimate PMU runtime output	YES		0	1
p2STN	Station Name	Remote_PMU			
p2STNb	Station Name part b				
p2IDC	Hardware ID Code	31		1	65534
p2TCP	Output TCP/IP local port	4831		1	65535
p2CFG	Configuration Change Count	0		0	32767
p2PHSout	Number of Phasors	3		0	12
p2lorFp	Phasor Number Format	REAL		0	1
p2OUTF	Phasor Output Format	Cn & phi		0	1
p2lorFf	Frequency Number Format	REAL		0	1
p2iAout	Number of Analog values	0		0	4
p2lorFa	Analog Number Format	REAL		0	1
p2iDout	Number of 16 bit Digital Status	0		0	1
p2LAT	PMU2 Latitude in degrees, WGS84 datum	49.88619		-90.0	90.0
p2LON	PMU2 Longitude in degrees, WGS84 dat...	-97.153191		-180.0	180.0
p2ELEV	PMU2 Elevation in meters, WGS84 datum	230.8		0	4294967...
p2ePHS1	Phasor 1	IA		0	0
p2ePHS2	Phasor 2	IB		0	0
p2ePHS3	Phasor 3	IC		0	0
p2ePHS4	Phasor 4	NO		0	0
p2ePHS5	Phasor 5	NO		0	0
p2ePHS6	Phasor 6	NO		0	0
p2ePHS7	Phasor 7	NO		0	0

Update | Cancel | Cancel All

Figure B.2: Remote PMU specifications

B1.3 Serial Communication

The Control station employs the developed PCDAA which transfers TTL control logic to the RTDS to implement protection and control function in the PMU-based differential protection scheme Figure B.5 and Figure B.6 illustrate serial communication configuration between the developed PCDAA and the Atmega 2560 to establish a serial communication link to transfer the TTL trip logic.

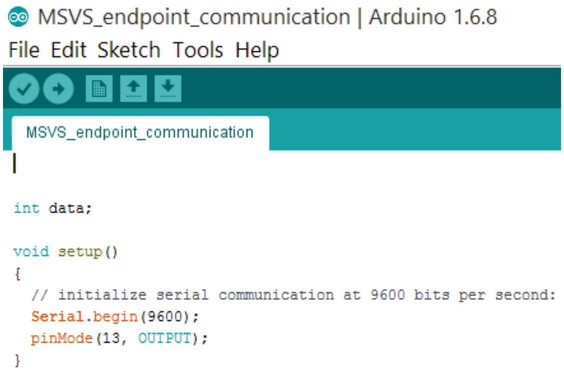


Figure B.3: Atmega 2560 serial communication configuration

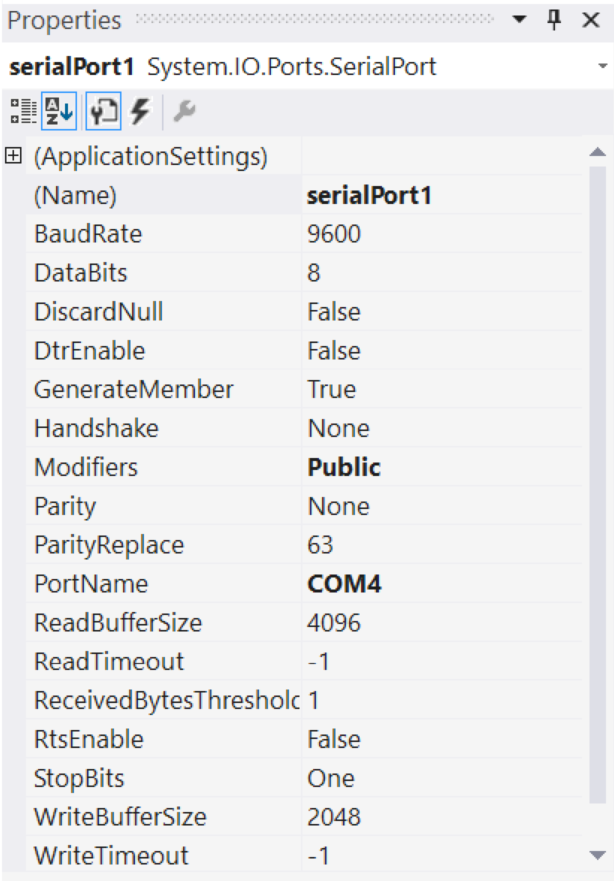


Figure B.4: PCDAA serial interface communication configuration