SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

Dynamic Phasor Estimation in Electrical Power Systems Based on IEC61850 Process-Bus

By

Ahmed Abdolkhalig

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

September 2014

Statement of Originality

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I give consent to this copy of my thesis, when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

The author acknowledges that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library catalogue and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Signed _____ Date _____

Abstract

State estimators in electrical power systems are implemented based on measurements that are provided by the SCADA systems or more recently, by dynamic phasor measurement units. The shortages of traditional SCADA systems such as the asynchronicity, the lack of using measurements with high sampling rates and complexity make them ineffective in capturing the dynamic response of a power system. Enhancing the performance of the state estimation by adding more dedicated phasor measurement units is costly. For reducing the cost and complexity of delivering measurements, IEC 61850 part-9-2 has been proposed using technology of Ethernet network communication. It uses a Process-Bus to link the equipment at switchyard and devices of protection and control at bay level. To enhance the phasor estimation and move from static to dynamic estimation, the high sampling rate of raw data provided by the Merging Unit as defined in IEC 61850 part-9-2 in combination with a Unscented Kalman filter as a dynamic estimator is proposed in the thesis. The major technical challenge that arises in applying Ethernet network communication is the impact of traffic performance on the estimated phasors and hence the substation protection and control reliability.

This thesis work is to study the performance of proposed dynamic phasor estimator influenced by delay and/or loss of the Sampled Measured Value time critical messages over the Ethernet network communication. Detailed modelling of the Sampled Measured Value packet format and the IEC 61850 based digital communication are carried out using C programming language and TrueTime simulation tool is presented. To benefit from the raw Sampled Measured Values streamed with high sampling rate from Merging Units at substation process level, Unscented Kalman Filter Model-Based for dynamic phasor estimation is developed and proposed to be used at substation bay level in this thesis. To show how some protective relays can react to the proposed dynamic phasor estimation model under the IEC 61850-9-2 communication, an overcurrent protective relay is implemented based on the proposed dynamic phasor estimation and tested against different operation conditions of the IEC 61850 Process-Bus.

I dedicate this thesis work to my Wife.

Acknowledgment

First and foremost, I would like to thank and express my deep sincere to my supervisor, Dr. Rastko Zivanovic, School of Electrical and Electronic Engineering, University of Adelaide for his guidance and his personal advices which made the progress and quality of this research ensured. The wide knowledge and logical way of thinking he has helped in realizing this research work and gave it a great value. Also, I would like to thank my co-supervisor Dr. Said Al-Sarawi for encouraging me and giving many advices and help on many parts of this research work. All thanks go to my loving wife for her support and endless love. She takes care of me, starting from what I eat to what I wear. Despite her studies, she always took care of the kids whether I was working on this research work or not. I am very lucky to have you in my life. I would like to thank and appreciate all my current and earlier colleagues at school that I worked with over the years. From each one of you, I had learned something new which also helped in some stages of this work. I extend my special thanks to all my relatives and friends in Libya for their honest support. I could not conclude without thanking all the members of IT staff, electronics shop staff and administrative staff at the school for their support at various stages of work.

Apologies if I have forgotten anyone, but if, it just an inattention.

Thank you all.

Publications

Journals:

1. A. Abdolkhalig and R. Zivanovic, "Simulation and Testing of the Overcurrent Protection System Based on IEC 61850 Process-Buses and Dynamic Estimator," *Sustainable Energy, Grids and Networks*, Under Review.
2. A. Abdolkhalig and R. Zivanovic, "Performance of Total Vector Error of an Estimated Phasor within Local Area Networks," *International Journal of Electrical, Robotics, Electronics and Communications Engineering* , vol. 8, No:2, 2014.
3. A. Abdolkhalig and R. Zivanovic, "Phasor measurement based on IEC 61850-9-2 and Kalman–Filtering," *Measurement,* vol. 50, pp. 126-134, 2014.
4. A. Abdolkhalig and R. Zivanovic, "Phasor Measurements within Local Area Networks," *International Journal of Information Technology & Computer Science (IJITCS)* vol. 12, 2013.

Conferences:

1. A. Abdolkhalig and R. Zivanovic, "Evaluation of IEC 61850-9-2 Samples Loss on Total Vector Error of an Estimated Phasor," presented at the IEEE SCOReD2013, 2013.
2. A. Abdolkhalig and R. Zivanovic, "Performance Evaluation of Phasor Estimator within IEC 61850-9-2 Communication Network," in *The International Conference on Electrical and Electronics Engineering, Clean Energy and Green Computing (EEECEGC 2013)*, Dubai, 2013, pp. 113-119.
3. A. Abdolkhalig and R. Zivanovic, "Towards dynamic on-line state estimator based on IEC61850–9–2 Process Bus," in *Electrotechnical Conference (MELECON), 2012 16th IEEE Mediterranean*, 2012, pp. 261-264.
4. Abdolkhalig, "Optimized Calculation of Hourly Price Forward Curve (HPFC)," *Proceedings of World Academy of Science: Engineering & Technology,* vol. 47, 2008.

Table of Contents

Table of Figures

List of Abbreviations Used

ADC:	Analog to Digital Converter
APDU:	Application Protocol Data Unit
ASDU:	Application Service Data Unit
CSMA/CD:	Carrier Sense Multiple Access with Collision Detection
CT:	Current Transformer
DFT:	Discrete Fourier Transform
DNP:	Distributed Network Protocol
EMS:	Energy Management System
EMTDC:	Electro Magnetic Transient Direct Current Analysis
ESW:	Ethernet Switch
FIR:	Finite Impulse Response
GOOSE:	Generic Object Oriented Substation Event
GPS:	Global Positioning System
GSSE:	Generic Substation Status Event
HMI:	Human Machine Interface
IEC:	International Electro technical Commission
IED:	Intelligent Electronic Device
IEEE:	The Institute of Electrical and Electronics Engineers
IP:	Internet Protocol
IRIG-B:	Inter Range Instrumentation Group -B
KF:	Kalman Filter
LAN:	Local Area Network
MAC:	Medium Access Control
MMS:	Manufacturing Message Specification
MU:	Merging Unit
OSI:	Open Systems Interconnection
P&C:	Protection and Control
Ph-Ph:	Phase-to-Phase
PMU:	Phasor Measurement Unit
PSRC:	Power System Relaying Committee
PTP:	Precision Time Protocol
RTP:	Real Time Playback
SAS:	Substation Automation System
SIR:	Source Impedance Ratios
SMV:	Sampled Measured Values

TC:	Technical Committee
TCI:	Tag Control Identifier
TCP:	Transmission Control Protocol
TPID:	Tag Protocol Identifier
TS:	Time Synchronization source
TVE:	Total Vector Error
UDP:	User Datagram Protocol
UKF:	Unscented Kalman Filter
VLAN:	Virtual Local Area Network
XML:	Extensible Mark-up Language

Chapter 1

1. Introduction: Overview of Phasor Estimation, IEC 61850 and Research Objectives

1.1 Introduction

This chapter is a review of the concept of the dynamic phasor estimation and the IEC 61850 standard, specifically the concept of the substation Process-Bus IEC which is IEC 61850-9-2 based. The major benefits and the significant features as well as the technical challenges related to the IEC 61850-9-2 for Process-Bus based substation monitoring and protection systems are also presented based on a detailed literature review. Finally, objectives of the research presented in this thesis (as technical challenges) motivated by the IEC 61850-9-2 as well as the research work methodology are discussed.

1.1.1 Concept of Dynamic Phasor Estimation

In electrical power engineering, the purpose of phasor estimator is to determine the best match to the state of the system depending on the accuracy of the measurements. To achieve such an aim, the system operators require accurate real-time measurements and a computationally efficient tool to keep the system reliable, in order to deliver energy to consumers, regardless of any dynamic change in the power flow and the plant availability conditions that can be found in the contemporary power grid [1].

The massive widespread of the power outage that occurred on August 14, 2003 throughout the Northeastern and Midwestern parts of the United States and Ontario was the second most widespread electrical blackout in history [2]. The Northeast Blackout of 2003 affected an estimated 50 million people and contributed to at least 11 deaths and cost an estimated $6 billion. Some reports [3-6] identified what happened was the result of a triggering event in which incorrect direct telemetry data causes the phasor estimators to become inoperative.

Any modern state estimator has to be able to monitor and analyze dynamic phenomenon that is crucial to investigate. This processing will estimate the state or the voltage phasor (polar coordinates) by direct means which could involve accurate synchronized phasor measurement units (PMUs) that are installed at all buses [1].

Enhancing the performance of the power system state estimation by using PMUs has been the ambition of most utilities in the resent years. If the power utilities are interested in how to improve their existing measurement, this can be done by increasing the number of PMU or re-designing the whole system to choose the optimum places for the PMUs deployed. In [7], it has been indicated that enhancing the state estimation by deploying more PMUs or re-designing the whole system will be very expensive, so the cost benefit will be a crucial factor for many utilities that are considering financial issues. Also, some other researches [8-13] have suggested algorithms to determine the optimum placement of each PMU with different aims, like develop the observability of the network, detection of measurement and how to maximize the information contained in the captured signal. In [10], it has been suggested a method to situate the PMUs while the station is refurbished. This method may be more cost effective than other methods.

Moreover, most utilities that are responsible for the substations today have few accurate synchronized PMUs deployed and their supervision is mostly done based on a telemetry system controlled by SCADA. Such an approach can have problems of measurement errors or telemetry failures [1]. It was reported in the examinations of the 2003 blackout [14] that, some PMU types were well protected against abnormal inputs and parasitic oscillations. The operation of PMUs relies on steady-state conditions and only slow oscillation changes in the power flow can be considered.

Classical phasor estimators rely on the Discrete Fourier Transform (DFT) algorithm [15, 16], which can have large errors. Another technique known as Kalman filter (KF) has been considered as one of the most robust tools for dynamic phasors estimation [17-23]. This dynamic estimation technique dates back to 1970’s when it was applied to enhance the performance of the static state estimation [24]. Kalman filter was used in the steady state models of the power systems to estimate the bus voltages and phase angles.

The goal of the dynamic estimator is to estimate the variables in the dynamic state that is represented by liner differential equation which represents the power system. Because the dynamic measurements include some noise, the traditional KF cannot be used directly. Therefore, the Unscented Kalman Filter (UKF) [25] is proposed in this research to incorporate with the non-liner state equation so that they can be applied to state estimation of the power systems.

From all above-mentioned, there is a need to design a complete measurement system for substation level that can compromise between cost and accuracy and supersede/complementing the existing measurements provided by SCADA. The objective is to use the IEC 61850 standard for such system [26]. The IEC 61850 standard is a world-wide accepted standard for the communication within the Substation Automation System (SAS). In general, the IEC 61850 standard consists of two kinds of communication Buses: Station-Bus and the Process-Bus. Both are Ethernet technology supported. The Station-Bus (IEC 61850, part 8) connects the bay level IEDs for protection, control and monitoring (i.e., bay units) with station level devices (i.e., the station computer with HMI and the gateway to the network communication center (NCC)) using whatever services are required by the applications. The Process-Bus (IEC 61850, part 9) was intended for the communication between bay and process level. One of the typical applications for the Process-Bus communication is to transfer the information between instrument transformers and protection devices. The information transferred is time critical messages and have high sampling rate. These time critical messages named Sampled Measured Values (SMVs). Introducing measurements with high sampling rate into power systems makes it possible to capture the dynamic behavior in state estimation if it is fed to UKF.

1.1.2 Protection and Automation in Substation Based on IEC 61850 Standard

Realizing a successful protection and automation system in a substation relies on the usage of available communication system to link the various monitoring devices for the protection and control within the substation. The biggest challenge that design engineers

can face is to provide interoperability among all monitoring devices that come from different manufacturers. Up until a short time ago, all manufacturers of monitoring devices were using communication protocols with their own proprietary for the different applications of the protection and automation in the electric power substation. Even though many protocols have been used for interlinking devices from different manufacturers in substations, enormous investment were needed to develop complicated and costly converters of protocols for interfacing these monitoring devices in electric substation [27]. In 2003, this issue was addressed by the Technical Committee (TC)-57 of the International Electrotechnical Commission (IEC) which published as IEC 61850 standard titled "Communication Networks and Systems in Substation" [26]. The standard covers all aspects of how to communicate and what to communicate in electrical substation automation design. Capabilities of IEC 61850 clearly offered what most former branded protocols had offered [28] like IEC 60870-5-101 [29], DNP 3.0 [30] and MODBUS [31].

The interoperability provided by IEC 61850 communication protocol is accomplished by defining special data format and the configuration language. This communication standard specified the OSI-7 layer-2 Ethernet-based communication systems [32]. The advantage of using Ethernet-based communication is to provide higher communication architectures and the incorporation with the fast growing of the Ethernet-based communication technologies [33]. The time critical and high-speed property of present Ethernet-based communication technology makes it an appropriate communication technology for applications of automation in substation [34]. Use of Ethernet-based communication in substation automation has been proposed in part-8 and part-9 of IEC 61850 standard at station-level and process-level, respectively. In the following subsection, description of the entire IEC 61850 standard scope is reviewed.

1.1.2.1 IEC 61850 Scope

The recent developments and fast growing in communication and networking technologies have presented new range of opportunities to utilities for improving their electric

operations and process automations. In SAS, the interoperability in a substation among the Intelligent Electronic Devices (IEDs) from different manufacturers is one of the major challenges that can arise. This issue is addressed by the IEC61850 development as a universal standard for substation automation. It's a framework for SAS that had defined the following aspects [35]:

I. Standardized meaning of services, data and object models

II. Common configuration language and naming conventions

III. Device behaviour models and self-describing devices

The development of IEC 61850 has been conducted in the early 1990s by the Institute of Electrical and Electronics Engineers (IEEE) and the Electric Power Research Institute (EPRI). The collaboration between the two Institutes was to define Utility Communications Architecture (UCA). Their efforts were to find appropriate architecture for the inter-control communications between the substation and control centre. Three years later, both Institutes started working together again for the next stage of the UCA (Station-Bus) [36]. All concepts and fundamentals of the work done by IEEE and EPRI in UCA became the groundwork for the IEC Technical Committee Number 57 (TC57) Working Group 10 (WG10) which yielded later in 1997 the IEC 61850 standard for communication networks and systems in substations.

This standard is more than a subset of features of UCA 2.0. The standard defined the different aspects of the communications in substation in subdivided ten major parts as shown in Table 1.1. Part 3 through 5 are identifying the general and the functional requirements of the communication systems in substations, as well as discussing the project managing. In part 6, the Substation Configuration Language (SCL) is defined based on the Extensible Mark-up Language (XML) for configuring the devices from different vendors with minimal human errors and less complexity. This SCL is bridging the relations between any SAS system and the switchyard in substations. Sub-part 7-1 is introducing the common data concept and the services modelling which have been adopted in the IEC 61850 standard. Sub-part 7-2 is defining the abstract services whereas the

various common data classes have been defined in sub-part 7-3. In sub-part 7-4, the abstract data objects (i.e. Logical Nodes) have been defined. These defined data objects are containing information, such as measurements, status, etc.

The common data classes were introduced to build the concept of the larger data objects. The data and service have been adapted to abstraction, which can allow the data and services to be mapped to any implementation of protocol stack that can meet the requirements [37]. In sub-part 8-1, the abstract data objects as well as the services have been defined and mapped into the Manufacturing Messaging Specifications (MMS). Sub-parts 9-1 and 9-2 are defining the signals digitization from the instrument transformers and how to map them into Sampled Measured Values (SMVs) packets over an Ethernet layer. The Ethernet based Local Area Network (LAN) has been proposed in IEC 61850-9-2 for the communication in the substations to send and receive the protection and automation messages between devices in process level and switchyard like CTs/VTs, CBs, and other field sensors and control and protection IEDs at bay level. This Ethernet based communication network should enable the time critical messages communication, such as raw SMVs and Generic Substation Events (GSE) which include the Generic Object Oriented Substation Event (GOOSE), within the allowable standardized time defined in the standard [38].

Table 1.1. Structure of the IEC 61850 Standard

Part#	**Title**
1	Introduction and Overview
2	Glossary of terms
3	General Requirements
4	System and Project Management
5	Communication Requirements for Functions and Device Models
6	Configuration Description Language for Communication in Electrical Substations Related to IEDs
7	Basic Communication Structure for Substation and Feeder Equipment

7.1	- Principles and Models
7.2	- Abstract Communication Service Interface (ACSI)
7.3	- Common Data Classes (CDC)
7.4	- Compatible logical node classes and data classes
8	Specific Communication Service Mapping (SCSM)
8.1	- Mappings to MMS(ISO/IEC 9506 – Part 1 and Part 2) and to ISO/IEC 8802-3
9	Specific Communication Service Mapping (SCSM)
9.1	- Sampled Values over Serial Unidirectional Multi-drop Point-to-Point Link
9.2	- Sampled Values over ISO/IEC 8802-3
10	Conformance Testing

1.1.2.2 IEC 61850 Based Substation and Function interfaces

According to IEC61850 standard, the recording, monitoring, protection and control system in substation have the hierarchical structure shown in Figure 1.1. The hierarchical structure consists of three levels, which are discussed below:

Level 0 (Process): This level includes all primary equipments in the switchyard of the substation, such as instrument transformers or binary status signals (binary control signals from sensors/actuators).

Level 1 (Bay): The bay level is the level between the Process-Bus and Station-Bus. It includes IEDs of the protection and the control for the different bays of the substation. The IEDs of protection and control at the bay level are connected to the Ethernet based Process-Bus network. The segments of the Process-Bus network of different bays can be connected by using a switch.

Level 2 (Substation): At this level, the overall equipment operations are related to its functions in a substation. The data used by the functions for one bay/s or the entire

substation can be implemented at this level of hierarchy structure. As an example, the breaker triggering by a protection relay, multiple breaker trippings by a failure breaker relay through using the trip output signals from a bus differential protection. The used numbers for interface functions in Figure 1.1 are described in Table 1.2. From Figure 1.1, it can be noted that the Substation-Bus network is establishing the communication between the bay level and the substation level. For these interfaces of communication application, IF1, IF3, IF6, IF8 and IF9 are supported as they are shown in Figure 1.1. Similarly, the purpose of Process-Bus network is to exchange the data between the bay level and the process level with interfaces IF4 and IF5 supported at this level.

Table 1.2. Functions and their Interface

Interface Types	**Function**
IF1	To exchange the protection data between station and bay levels
IF2	To exchange the protection data between the bay and remote protection levels
IF3	To exchange the data within the bay level
IF4	To exchange the sampled instantaneous data of current and voltage transformers between the process and the bay levels
IF5	To exchange the control data between the process and the bay level
IF6	To exchange the control data between the bay and the station level
IF7	To exchange the data between the substation level and the remote workplace
IF8	For fast exchange of data among the bays (interlocking functions)
IF9	To exchange the data within the substation level
IF10	To exchange the control data between the devices in substation and the remote control center

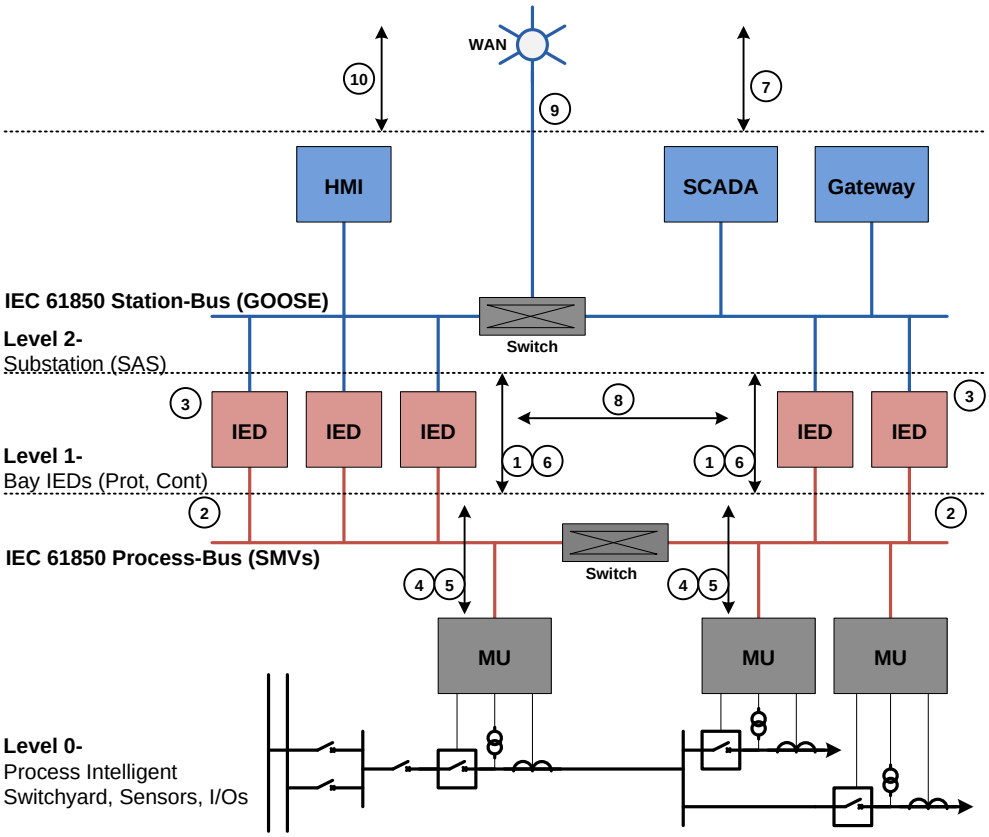


Figure 1.1. IEC 61850 Process-Bus Based SAS

1.1.2.3 IEC 61850 communication based on OSI-7 layer

The communication in IEC 61850 standard is done based on OSI-7 layer stack with three different groups and seven types of messages for the communications as shown in Figure 1.2. The seven types of messaging can be mapped into different communication stacks. The raw Sampled Measured Values data (type 4) and the GOOSE messages (type1) are time critical messages which are required to map directly to the low–level Ethernet layer. The performance of these time critical messages can be improved by shortening the Ethernet frame (no upper layer protocols are required) and reducing the time of processing. The type 2 (the medium speed message), the type 3 (the low speed message), type 5 (the file transfer functions) and type 7 (the command message with access control) are all mapped to MMS protocol suits which are a TCP/IP based stacks. The timing of messages

is synchronized based on the Simple Network Time Protocol (SNTP) and broadcasted by using UDP/IP to all IEDs in the substation.

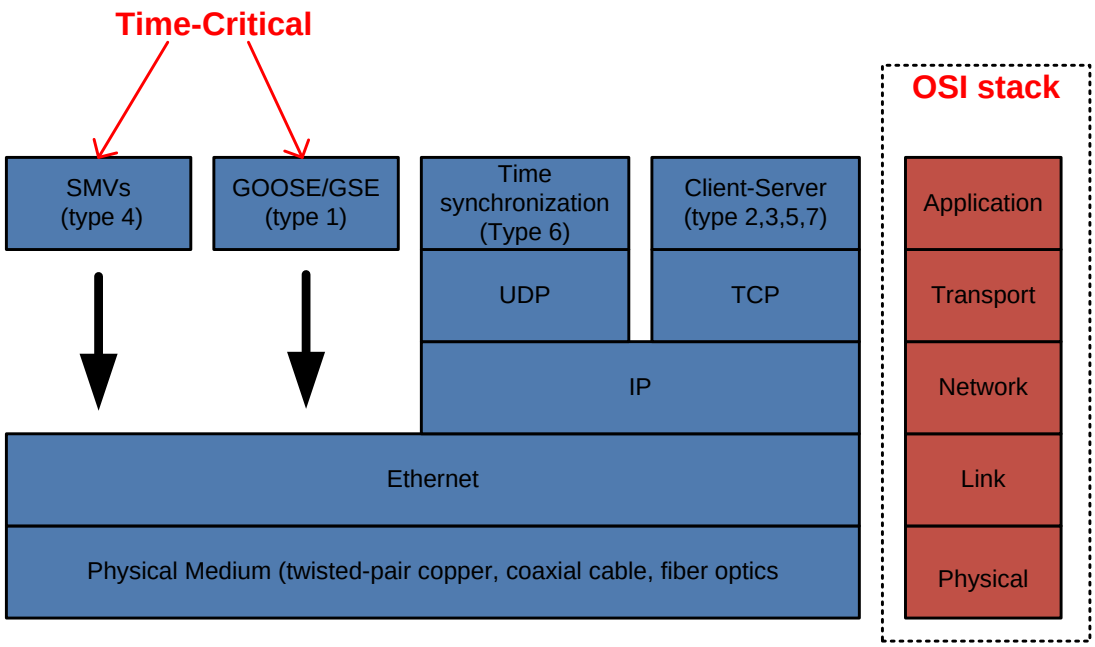


Figure 1.2. Communication Services as Defined in IEC 61850 Standard [26]

The GOOSE, SMVs and all other (client/server) message features are discussed below:

1. Features of GOOSE/GSE:
   A. Time critical data (e.g. trips, blocks and interlocks are using these messages).
   B. It does not use the TCP/IP services (i.e. it is the less reliable mechanism for the data transfer).
   C. The loss of some GOOSE messages can be compensated by repeating the message transmission.
2. Features of SMVs

A. The Sampled Measured Values of current/voltage signals from the nonconventional instrument transformers (IEDs) are Time critical data.

B. Sample loss can be compensated by the continuous data stream (redundancy) which its rate can be determined by the standardized sampling frequencies.

3. Features of Client-Server communication

A. Exchanging of information, such as the records of fault, the records of event, Sampled Measured Values are predominantly.

B. It is reliable for data transferring (all services of the OSI -7 layer stack are used).

C. Data transfer is not time critical.

1.1.2.4 IEC 61850 based communication Advantages

The IEC 61850 can offer major advantages as listed below:

A. **Interoperability:** Communication among different devices from multi-vendors.

B. **Free Configuration:** For any number of functions for protection and control, the integration and configuration is possible using the Substation Configuration Language (SCL).

C. **Simplicity for Future-proof Architecture:** As numerous point to point copper wiring are reduced to one OSI-7 based Ethernet communication link, many critical functions in an IED are possible to be integrated to have one OSI-7 layer stack.

1.2 The Concept of IEC 61850-9-2 Process-Bus

For reducing the cost of complex and long copper wires, IEC 61850 and optical cabling have been proposed to link between the devices in the processing switchyard and the bay

IEDs for protection and control. This Ethernet based communication link is referred to as IEC 61850-9 Process-Bus. The analog inputs to conventional IEDs are replaced by integrated Analog-to-Digital Converters (ADCs) and also with binary Inputs-Outputs I/Os modules. The IEC 61850 has specified the architecture of the distributed IEDs, which is installing the ADCs and binary I/Os modules near to the signal sources into the switchyard. The analog signals received from current or voltage transformers are digitized by using Merging Units (MUs). The digitized signals are casted over the Ethernet based Process-Bus to the protection and control IEDs in the bay level. The MU is the key element of the IEC 61850 Process-Bus [39].

As shown in Figure 1.3, the MU gathers voltage and current samples from the instrument transformers and the status information from the transducers. The gathered analogue values are digitized and then merged into the standardized SMV data packet format. This way, a single Ethernet communication based Process-Bus network can carry many digitized signals. For protection functions to be able to use these digitized signals, the stream of SMVs from various MUs should be time synchronized. The synchronization source of MUs can be either an external time signal (e.g. IRIG-B, GPS clock) or internal time signal like precision time synchronization protocol (IEEE 1588/IEEE C37.238) [40]. The synchronization provides a time stamp on each casted data packet. The data packets are sent to each corresponding IEDs for protection and control at bay level using the standardized Ethernet based communication links. Two sub-parts of a Specific Communication Service Mapping (SCSM) had been defined in IEC 61850-9 for the purpose of Sampled Measured Values transmission.

The sub-part 9-1 of the IEC 61850 had specified a serial unidirectional and multi-drop point to point link to carry non-configurable (fixed) data set in accordance with IEC 60044-8 [41], while the sub-part 9-2 of IEC 61850 [39] had proposed a configurable (user defined) bidirectional and multicast communication of dataset using an Ethernet LAN (defined in IEEE 802.3 [42]). Advantages like the data rates, the zero-collision, and flexible architecture of the ANSI/IEEE 802.3 based Ethernet switched communication which is standardized in IEC 61850-9-2 made it preferable over the serial point to point

standard links (standardized in IEC 61850-9-1). The implementation guideline [38] had defined two standardized sample rates with one sample rate equals to 80 samples per cycle for basic protection and control, and the other has a high rate of 256 samples per cycle for power quality and measurements.

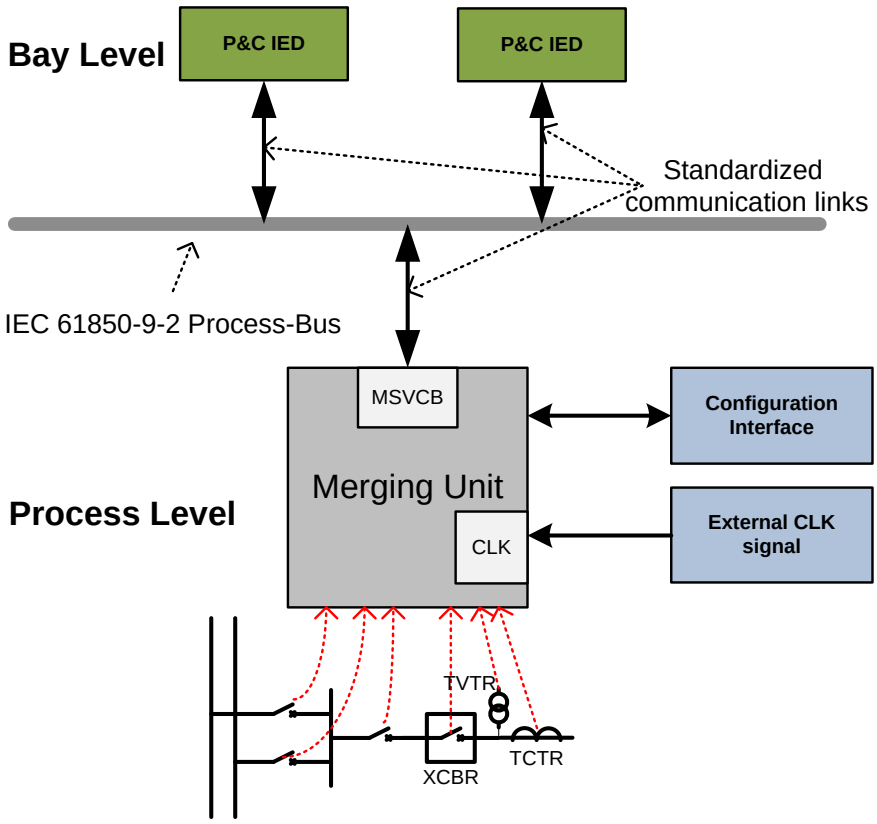


Figure 1.3. Concept of Process-Bus

1.3 Benefits of IEC 61850 Process-Bus

Ethernet Process-Bus communication network based on IEC 61850-9-2 can offer many advantages and benefits as discussed next.

1.3.1 Flexible and Adaptable Architecture with Simple Wiring

The Process-Bus based protective and control applications have many major benefits over any conventional copper wiring analog circuits at substation switchyard. The more important one is the large number of point to point copper cables are replaced with a few fiber optic cables which lead to a significant reduction in the cost of the system for the substation protection and control. In any conventional protection and control system, the protection relay can have access to few measured signals and therefore, the protective function can perform the task with a minimum number of used input signals.

1.3.2 Cost Savings

IEC 61850 Process-Bus provides a cost-effective solution in the long run as seen on Figure 1.4. Analysis of the cost for Ethernet Process-Bus communication network based IEC 61850-9-2 has already been done in [43-45]. In [45], a business case study showed that the total cost after an implementation of Process-Bus in a typical substation was reduced to approximately 25% of the classical design. Even though, the cost of the material that is associated with the Process-Bus implementation has been increased (from 27% to 42%), due to more communication network devices that are need, the labour cost and engineering operations are expected to be reduced from about 73% to 34% of the total project cost . This is because Process-Bus implementation requires less engineering hours and assembling hours, per project.

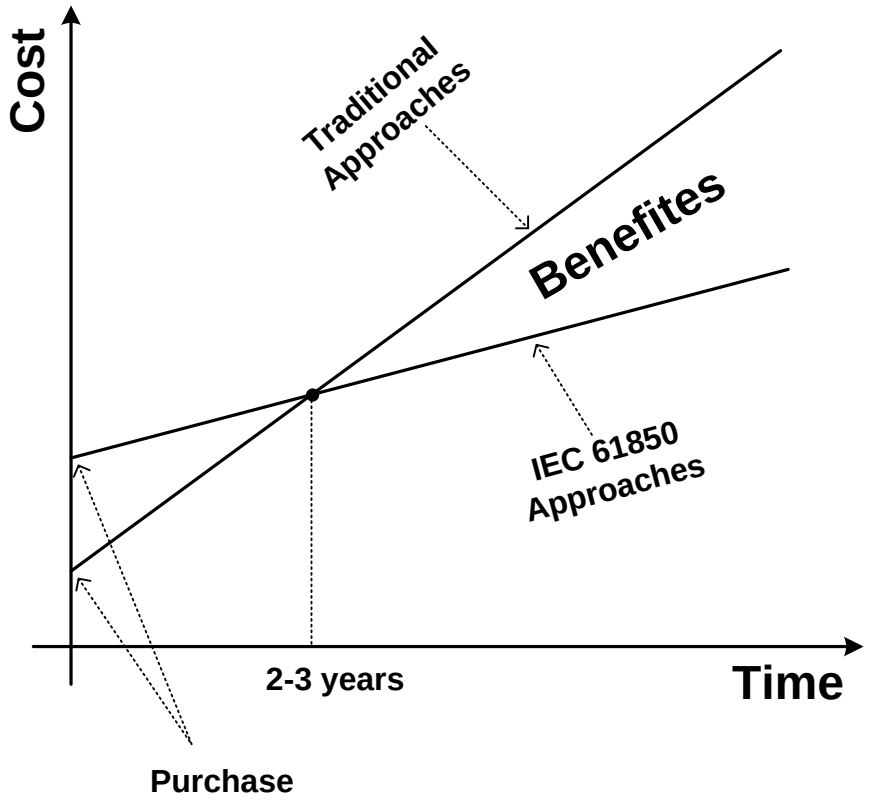


Figure 1.4. Justification for IEC 61850 Cost Saving [46]

1.3.3 Interoperability

The complete implementation of protection functions can be provided by two or more devices at bay level and MUs at process level and they can be from different vendors. Correct co-operation and information exchange between devices is required. The receiving device has to understand the data structure (syntax) and its meaning (i.e. semantics based on attributes of received data). A simple demonstration presentation of the IEC 61850 Process-Bus interoperability is shown in Figure 1.5. This figure shows the interoperability of IEC 61850-9-2 based Process-Bus; the vision and its capability to interface seamlessly devices from different vendors for a common application.

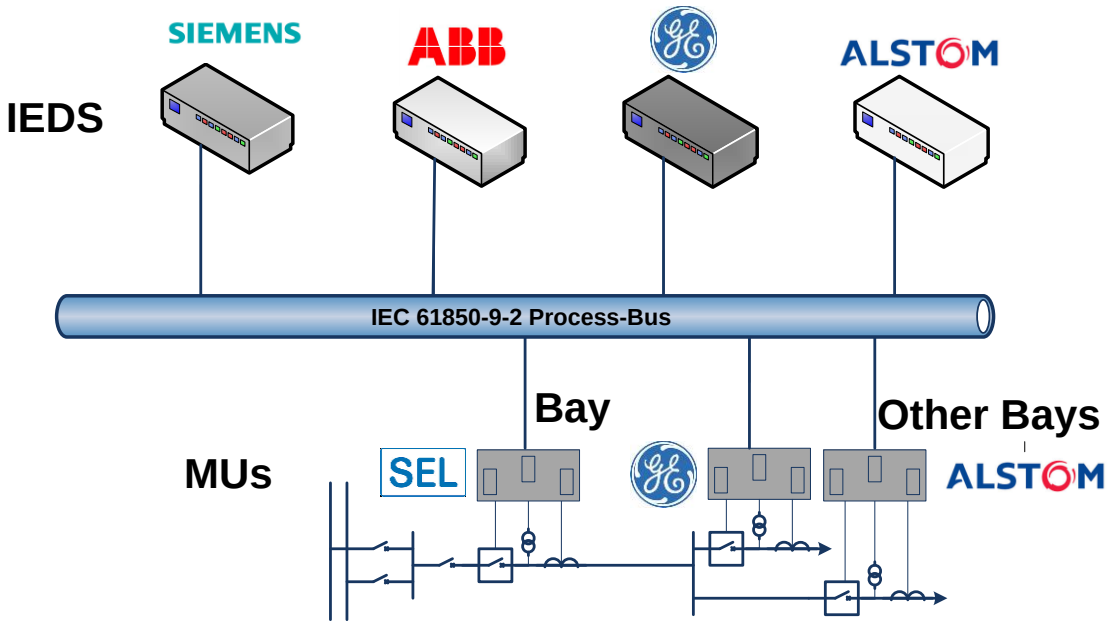


Figure 1.5. IEC 61850-9-2 Based Process-Bus Interoperability

1.4 Issues with Protection Based on Process-Bus

Nowadays, many major manufacturers are working on the development of interoperable devices for the application of the IEC 61850-9-2 Process-Bus based time critical SMVs over the Ethernet networks [47-49] after the successful implementation and testing of IEC 61850 standard based GOOSE at several testing facilities and substations worldwide [50, 51]. The goal is to reduce the complexity of copper wiring between the switchyard and the control room which is the major contributor of the labor cost [52-54]. However, technical issues such as the performance of time critical SMVs messages over Ethernet based Process-Bus, as well as the communication availability and reliability need to be analyzed [55, 56] for the successful implementation of protection application in substation based on Process-Bus network. The technical issues of the implementation of Process-Bus have been identified in literature and analyzed in this section.

1.4.1 Performance of Ethernet Network Based IEC 61850-9-2 Process-Bus

The end-to-end delay and loss of packets of important pieces of information are the main factors that can affect the dynamic performance of any communication network. The part-5 of IEC 61850 standard [57] had specified the allowable time delay of message transmission and the approaches for studying the network performance of SAS. This maximum possible time delay is for both time critical messages (i.e. GOOSE and SMVs) and also should be independent from the traffic load of the Process-Bus communication network.

There is no guidance in the standard for characterizing the performance of the message delivery across the whole Ethernet communication network of the power substation [58] . Furthermore, the end-to-end delay and loss limits of packets cannot be predicted even though the problem of collision can be eliminated by using communication full duplex switched Ethernet. This is due to the queuing delay which is not constant and the packet buffers have limited storage size [59] which will result into non-deterministic delays and/or loss of packets. Therefore, the performance of the physical IEC 61850-9-2 Process-Bus based SAS network should be evaluated.

In [34, 60], preliminary simulations of Ethernet network for substation automation without any modeling for communication stack based on IEC 61850. References [58, 61, 62] have been proposed models based on IEC 61850 standard for IED of a protection and control and for a MU in OPNET software [63] and in RTDS [64]. However, it is very important to consider and evaluate the impact of different parameters that affect the communication network performance, such as the band-width (BW) assigned, the cable length and the nodes number.

1.4.2 Loss/delay of Time Critical Messages

According to the IEC 61860 standard, the time critical messages (i.e. GOOSE and SMVs) had mapped directly on the Ethernet link layer (layer 2 of OSI-7). This is to reduce any additional time delay in the use of TCP/IP (Transmission Control Protocol/Internet

Protocol) layers. However, the reliability of these time critical messages communication might be reduced due to this elimination of TCP/IP layer [65-67]. To enhance the reliability of transmission, IEC 61850-8-1 proposed repeating the send of the same GOOSE message several times.

In general, GOOSE is sent fewer times in a second to the network as an event triggered message while SMV messages are transmitted as time triggered messages at the sampling frequency rate. The transmission of the SMV messages is not repeated, and this will reduce the reliability of their transmission over the Process-Bus. Even though, Virtual Local Area Network (VLAN) with priority tagging can be used, this cannot assure the communication delays and packet loss determinism during the worst case operation conditions [68, 69]. Therefore, it is very important to analyze and study the IEC 618509-2 based Process-Bus communication network dynamic in order to evaluate the SMV loss rate or their delay on the phasor estimation and protection functions performance. The loss or delay of SMVs is applied to the entire SMV packet, which includes samples of all voltage and current signals that are obtained at the same time-stamp from the corresponding voltage and current transformers. Therefore, the delay or loss of SMV can have a deep impact on any protective relay. In [70, 71], the loss of SMVs has been demonstrated on the digital protective relays performance, and an adaptive filtering for loss of single SMV has been presented. The currently deployed digital protection relays in the substations are not designed to handle the Process-Bus contingencies (delay or loss of SMV packets). In Reference [54], it has been stated that the loss of SMV can cause the measuring window in a protection relay to has some kind of measuring blackout. In order to consider IEC 61850-9-2 based Process-Bus communication networks for any future digital protection and automation systems, the impact of SMV loss or delay on the phasor estimation and digital protection of substation should be alleviated by developing proper techniques. The need for smart algorithms that are able to treat the loss or the delay of SMVs for the next generation of IEDs have been identified by the developers of Process-Bus products [56].

1.4.3 Other Issues

There are many other technical and non-technical issues that can be considered for Process-Bus implementation and are discussed below[52].

1.4.3.1 Issues of Time Synchronization

The time synchronization has been proposed in IEC 61850 for the time critical messages based on Simple Network Time Protocol (SNTP). The accuracy required for the raw data sampled measured values time synchronization is less than 1ms and the SNTP is not able to provide such accuracy. IRIG-B has been suggested as one of the solutions to synchronize the signals [72]. One of the major problems when using IRIG-B is the needs for an external source of time synchronization, and accuracy required is depending on the quality and the availability of the time synchronization. The third potential suggested method is using the IEEE 1588 standard [40] time synchronization technique over Process-Bus Ethernet network. It is planned to include the IEEE 1588 based time synchronization in the next revision of IEC 61850 in order to achieve a higher accuracy in the range of 1μs.

1.4.3.2 Issues with Data Security

In power substations, any interloper can cause swamped packet, port reservation, etc. This would introduce some loss of SMVs and a large time delay even if high priority messages technique is applied which could cause damage to many devices in substation. The security of data is considered more important when data is exchanging with the control center or the other substations. Some security issues can be addressed for example by the encryption of data at the sending and receiving ends with secured key or by the authentication by using security password [73]. The auditing for the security of the substation automation system based on IEC 61850 has been carried out in [74, 75]. However, the infrastructures of the substation communication need to be studied to get further security measures which have been addressed in [76].

1.4.3.3 Issue of EMI/RFI Noise Immunity

Different electromagnetic interferences (EMI) such as, lightning strike, electrostatic discharge, switching surge, and Radio frequency interference (RFI) from wireless equipment are commonly encountered in Air Insulated Substation (AIS) which can affect the functioning of equipments and devices (such as IEDs, MUs, Ethernet switches) in SAS substation. Currently, the general EMI/RFI immunity used in industry is not sufficient for insulating substation equipments and devices whereas IEC 61850 part-3 had specified only the EMI/RFI immunity outline. The requirements are given in parts of IEC 6100 series (IEC 61000-6-5 and IEC 61000-4-x) [77] or in IEEE C37.90.2 [78]. All devices in SAS must be in compliance with these standards for EMI/RFI immunity especially at the substation Process-Bus level.

1.4.3.4 Issues with Version Upgrade

It is mentioned earlier that IEC 61850 standard provides free configuration that can lead to isolation of functions for single zone and hence, the operation of single zone can be dependent on proper configuration of hardware and software of devices from different vendors. Furthermore, co-ordination with the existing versions from different vendors may not be supported by the update in any software or hardware of installed IEDs which would need to update all functions in a zone that is including devices from multi-vendors [56].

1.4.3.5 Training of Manpower Issues

IEC 61850-9-2 Communication network Based Process-Bus will be the backbone of the substation protection and control and hence, the utilities should train substation engineers to acquire enough knowledge and keep updating them with the growing communication technology for the operation of the complete networking [56].

1.5 Motivation

IEC 61850-9-2 Ethernet communication network based Process-Bus can offer many benefits such as, the flexible architecture, the interoperability and the overall cost savings. However, from the above literature review, it is obvious that there are some major technical issues, such as the Process-Bus communication network performance, especially for the time critical messages (SMVs) for phasor estimation and protection applications in substation. Also, the impacts of Process-Bus communication architectures reliability on the substation protection devices are still unanswered and need to be studied and evaluated. Even if, the IEC 61850-9-2 based Process-Bus has been proved to be an attractive technology for protection systems, the examination of the substation protection systems feasibility and reliability over the Ethernet switched Process-Bus network in terms of the performance of the time critical messages (SMVs) is very important.

1.6 Research Objectives and Methodology

The main objectives of the research presented in this thesis are to enhance the phasor estimation by developing dynamic state estimator using sampled measured values provided by the Merging Unit that is defined in IEC 61850-9-2 and investigate the impact of IEC 61850-9-2 based Process-Bus communication network on the performance and the reliability of the phasor estimation and hence on the over-current protection in power substation. The research objectives can be highlighted below:

1. Developing a Merging Unit function in real-time Linux operating system which is able to construct and cast Sampled Measured Value packets.
2. Developing code to read and reconstruct the transmitted data from the received Sampled Measured Value packets.
3. Developing code for phasor estimation based on received packets of SMV at bay level to supersede numerous traditional PMU installed at switchyard.

4. Analyzing the performance of the estimated phasors under Process-Bus communication networks operation
5. Developing the over-current protection function based on the developed phasor estimation function.
6. Evaluating the reliability and availability of over-current protection function based on the IEC 61850-9-2 Process-Bus communication networks for a typical power substation layout.

1.7 Thesis Structure

This thesis is organized in six chapters and two appendices (A and B). Chapter 2 discusses the features of substation Process-Bus communication network based on IEC 61850-9-2 for protection and control systems and presents the software environment and modeling setup. It will discuss the time critical messages (GOOSE and SMVs) for protection functions, mechanism of GOOSE retransmission, multi-castings of SMVs, and also the architectures of flexible and fast Ethernet communication. A detailed overview of Ethernet architectures as it was reported by The Power System Relaying Committee (PSRC) for Process-Bus will be also provided in this Chapter. The detailed implementation of the Process-Bus and C programming of functions to write and read SMVs under Linux will be discussed in the same Chapter.

Chapter 3 will present the Unsented Kalman Filter based model for phasor estimation and how it can be used as a tool for central phasor estimation. Studying the frequency response of the proposed estimator and comparison with the classical one is also presented in the same Chapter.

Chapter 4 presents the performance evaluation of the proposed phasor estimator within IEC 61850 substation Process-Bus. Effects of IEC 61850-9-2 communication network and the time errors due to the Asynchrony between Merging Units on the performance of the estimated Phasor are evaluated and presented in this Chapter. The phasor estimation is also

evaluated for the Total Vector Error in the same Chapter. The Total Vector Error is analyzed for the effects of the IEC 61850-9-2 samples loss, operating conditions at different node numbers and IEC 61850-9-2 frame size. The evaluation tests in this Chapter include test of the phasor estimator for both stationary and transient phenomena in power systems, to determine the accuracy of the measurements obtained at various conditions.

Chapter 5 presents the development of the over-current protection function and how the traditional peak values estimator of current relays found in substations is replaced with the developed phasor estimator. The effects of the different factors of Process-Bus communication network based on IEC 61850-9-2 like BW, cable length and node numbers on the proposed over-current digital protective relay function will also be analyzed and presented in this Chapter.

Chapter 6 will highlight the major outcomes from the research and the conclusions from the entire work. The possible further work from this thesis will be also discussed in this chapter.

Appendices A and B contain the codes of C programming of the write and read functions of IEC 61850 SMVs with different layers.

Chapter 2

2. Overview of the IEC 61850-9-2 Process-Bus Features and Implementation of SMV Write/Read Function

This chapter will introduce the main features of IEC 61850-9-2 based Process-Bus, such as the time critical messages (SMVs and GOOSE), the multicasting of SMVs, the GOOSE retransmission, the different Ethernet configurations and the time synchronization techniques. Also, the implementation procedures that have been used to create the SMVs publisher/subscriber functions are explained in details. The programming for the encapsulation of SMV with different layers and how the Process-Bus interface is embedded into Matlab environment for real-time simulation are also documented in this Chapter.

2.1 Main features of IEC 61850-9-2 Based Process-Bus

The IEC working group (TC57) has published in 2003 the IEC 61850 standard [26] which is for defining the communication networks and systems in power substations. The standard has been divided into ten parts, which cover how and what to communicate. The standard also provided the interoperability between two or more devices from different vendors by defining logical nodes into SAS functions. The Ethernet based communication at substation process level with high speed allows the replacment of traditional copper electrical wiring used in the substation automation system, which could be time and cost saving. The IEC 61850-9-2 major features are discussed below:

2.1.1 Time Critical Messages Over Process-Bus

According to the operational of IEC 61850-9-2 based Process-Bus, there are two types of processes which are event-triggered and time-triggered. In case of event-triggered, the message will be sent due to the occurrence of a certain event (e.g. change of state). The node that will receive this event may perform some operations that is relevant to that event

triggered (e.g. algorithms computation). This event-triggered message will be casted many times at specific pre-defined time instances. The time-triggered SMV message period is typically small (e.g. 0.25 millisecond for 80 samples per cycle). The traffic over the Process-Bus due to SMVs casted from several merging units can occupy a large portion of the network bandwidth. It may need to be necessary that the time between the generation of an event and the algorithm computation should be bounded.

2.1.2 Mechanism of GOOSE Retransmission

GOOSE is mechanism for control in which any data format (value or status) can be grouped into a data set and then transmitted within a specific time period of 4 milliseconds. To ensure specified transmission speed and reliability, the following mechanisms should be used.

I. The GOOSE data should be embedded directly into an Ethernet data packet.

II. The GOOSE should work on mechanism of publisher-subscriber with multi-cast or broad-cast of MAC addresses.

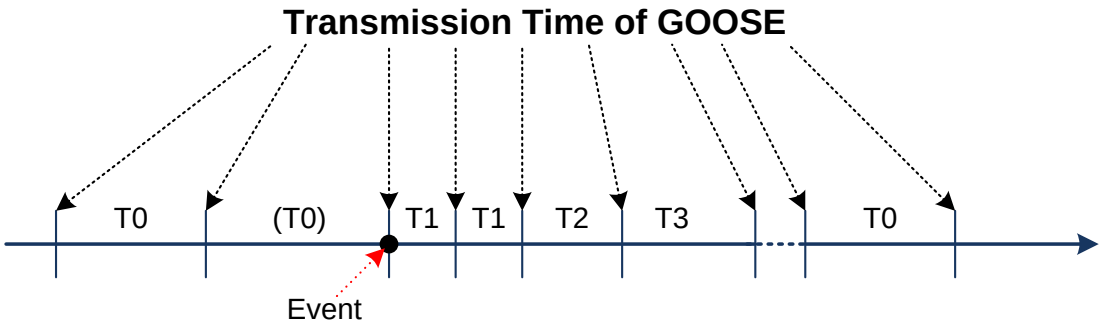
III. The GOOSE should use the VLAN with priority-tagging as per IEEE 802.1Q.

IV. The GOOSE should have a separate virtual network within the same physical network.

V. The same GOOSE message should be retransmitted with varying and increasing time intervals.

To realize the specified transmission speed and reliability of GOOSE message delivery, the part 8-1 of IEC 61850 standard specifies the retransmission mechanism for GOOSE message, as shown in Figure 2.1. Any change in status will initiate the GOOSE messages to be published repeatedly. In in each GOOSE message, there is a parameter called

maximum time which set the time of waiting between the publications of consecutive messages. When the maximum waiting time expired or any changes in data set value, the GOOSE message will be published over the Ethernet network. The change in any data set elements (e.g. some event takes place) will cause the GOOSE messages to be sent repeatedly with an incremental period to insure that all subscribed IEDs will receive them. To help the subscribed IEDs to monitor the correct data flow, the GOOSE message includes a total time to live, which predicts the time delay that should be before the next GOOSE message will be published.



T0: Retransmission in stable conditions (No event for long time)
(T0): Retransmission in stable conditions maybe shortened by an event
T1: Shortest transmission time after the event
T2, T3: Retransmission time until achieving the stable conditions time

Figure 2.1. Mechanism of Transmission Time for GOOSE

The occurrence of a new data set event (e.g. binary change of state) will create and publish a new GOOSE message. The data set of new event will be transmitted and repeated in the shortest T1 (see Figure 2.1) then, the retransmission time will be gradually increased from T2 to T3, and eventually will settle at the stable time of retransmission. This stable time of retransmission will be shortened if any next new event occurs. Subscribed IEDs calculate the time to be waiting constantly based on the time to live included in each GOOSE message. The subscribed IED considers “decay of data” if time to wait is expired and it has

not yet received any a new replacement message from the publisher. The subscribed IED will assume a loss of communication and then modify its logic if it detects an expiration of the wait time. The mechanism of GOOSE message retransmission is required to perform transmission from one publisher to one or many subscriber/s to allow the subscriber/s to identify that the communications over Process-Bus is healthy. However, the reliability of time critical tasks may not be guaranteed depending on the final retransmission time choice [79].

2.1.3 GOOSE and SMV Messages Multicasting

Any digital protection devices that work over IEC 61850-9-2 Process-Bus network should support the multicasting of time critical messages. The multicasting can allow the sending of same messages to multiple subscribers. In this mechanism, a publisher (IED or MU) will multicasts or in other words publishes a message to all subscribed IEDs. Since, the GOOSE and SMV messages are mapped directly to the OSI-7 data link layer 2; all devices (MUs and IEDs) should have an Ethernet data link layer-2 multicasting support.

Multicasting can be realized through the use of the Ethernet MAC (Media Access Control) source and destination addresses that can be defined in the frame of the Ethernet packet. This is 48 bits allocated in the Ethernet packet to identify both the destination and source addresses of the frame. The destination address can specify either the address for one subscribed IED or multicasting address for a group of IEDs. This can help in reducing the traffic over the Process-Bus network whereas the same time critical message need not to be repeated several times to subscribed IEDs.

2.1.4 Synchronization of IEC 61850-9-2 Based Process-Bus Devices Time

The IEC 61850 standard offers MUs to digitize analog signals at substation switchyard and cast them to protection IEDs at bay level through the Process-Bus communication networks. For a correct operation of all IEDs, these MUs which are located at different

locations of the substation switchyard have to be time synchronized. Currently, there are many techniques for synchronizing to a common reference time stamp. IEEE 1588 based PTP-v2 standard [40] and IRIG-B standard [80] are the major available solutions which offer powerful techniques for achieving a high time synchronization accuracy in the range of 1 μs. This high accuracy will made any digital protective relay to be able to use all SMVs published by any MU at substation switchyard. The following sub-sections of this chapter will provide more detailed information about these time synchronization techniques.

2.1.5 Ethernet Architectures

Since introducing local area networks (LANs) by the defense department in the mid of 1970's for interconnecting data processing equipments, the Ethernet technology based LANs become the most common and widely used in the world. Applications of the Ethernet technology based LANs in automation systems came to play in 1985 [81]. Later, many Ethernet technology based solutions such as Ethernet/IP, Profinet, Profibus, Modbus, EtherCat have been developed and used in the industrial automation systems [68]. Today, practice of using fast Ethernet based network using layer 2 in substation for protection and control is impassioning many utilities.

The major benefit that comes from the Ethernet switch technology is the needlessness of CSMA/CD protocol whereas the time critical messages can be read, buffered and forwarded directly to the corresponding MAC address/es in full-duplex mode which can give improved performance of the high speed real-time communications [82]. Some IEEE 802 protocol futures can be specified as follows:

i. **EEE 802.3x Full-Duplex:** A flow control mechanism that modulates frames transfer with a “PAUSE” to halt frames transmission until the slower server catches up to the faster server so no collisions can occur over the network.

ii. **IEEE 802.1p Priority Queuing:** A tagging mechanism that can allow frames to be tagged with different levels of priority to ensure that a real-time critical traffic is always available even during high periods of congestion.

iii. **IEEE 802.1Q VLAN:** A Mechanism to group and segregate specific ports into what is called virtual LANs for isolating different applications traffic.

iv. **IEEE 802.1w:** A Mechanism to allow the creation architectures of fault tolerant ring network that can be reconfigured in milliseconds which is less compared to the original Spanning Tree Protocol 802.1D in tens of seconds.

2.2 IEC 61850-9-2 Based Process-Bus Time Synchronization

As mentioned before, time synchronization with accuracy of 1µsec which is required in many protection applications can be achieved today by using two techniques, such as synchronization by using external time source based on IRIG-B protocol or synchronization using IEEE 1588 based PTP protocol over an Ethernet network.

2.2.1 Time Synchronization Using IRIG-B Protocol

Since IRIG-B protocol was developed in 2004, it has been widely used by many electric utilities to precisely synchronize relays, breakers and metering devices in their power substations. The time codes for IRIG protocol were developed originally by the Inter-Range Instrumentation Group (IRIG) [80] which is a part of the Range Commanders Council (RCC) of the US Army. The first publication of this protocol was in 1960 and followed by several revisions and updates. The latest version is IRIG standard 200-04 "IRIG Serial Time Code Formats," which is updated in 2004 [83].

The time code B (IRIG-B) protocol is widely used by many electric, industrials and other utilities for synchronizing the time between devices precisely. The IRIG-B protocol has a pulse rate of 100 pulses per second (PPS) with a time index count of 10 milliseconds

for one-second time frame. This frame contains time-of-year and the information of year in a Binary Coded Decimal (BCD) format and can contain (optionally) seconds-of-day in Straight Binary Seconds (SBS). Prior to the revision published in 2004, the information of year was not specified in the IRIG-B standard.

The IEEE Standard 1344 was used instead which was used as means for encoding the year in unused control bits that came to be known later as "IEEE-1344 Extensions." Typically, IRIG-B protocol is distributed based on a DC level shift with pulse-width coded signal (“unmodulated IRIG-B”) or alternative (“modulated IRIG-B”) that is an amplitude modulated signal based on a sine wave carrier with a 1kHz frequency.

The signal levels and the implementation details are not defined in the IRIG-B standard. The IRIG-B standard is usually distributed over shielded twisted-pair cable as an unmodulated TTL-level signal. The implementation of IRIG –B requires dedicated or separate cabling which can be considered a disadvantage. To synchronize all devices, a coaxial cable is used to connect them in daisy chain or in star topology with internal optical isolation also between them and master clock for preventing ground loops.

Figure 2.2 shows the Process-Bus (Ethernet switch network) of protection system for a transmission line-L with external Time Synchronization (TS) sources for synchronizing all devices (MUs and IEDs) in bay based on IRIG-B standard. Any IRIG-B signal source can be connected to any individual MUs and IEDs or can be also shared between any two closely located MUs. From this Figure, it can be noted that with this architecture, it would need to increase the TS sources in the Process-Bus. Also, it can be noted that there is a redundancy in the Process-Bus architecture, for example; some merging units (MU-17, MU-18, and MU-29) are connected to the Ethernet switch ESW-11 (redundant) over a redundant network. The P&C IED-2 for line-L is connected to redundant Ethernet switch in a cloud of 10 Ethernet switches connected in a ring.

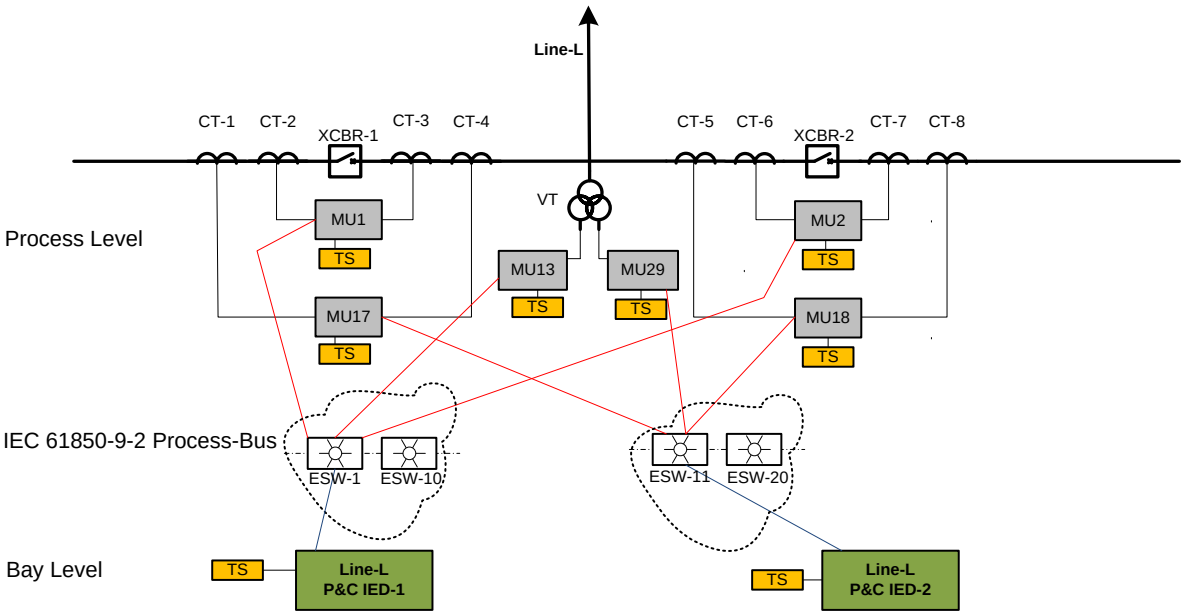


Figure 2.2. Time Synchronization Using IRIG-B

2.2.2 Time Synchronization Using IEEE 1588 Based PTP

The accuracy required for the measurements can be realized by installing a GPS receiver at each measuring device. Because of the cost considerations and the large number of measuring devices; this solution might be undesirable as the cost of installing new PMUs is high. The Precision Time Protocol (PTP) IEEE 1588 Standard [40] is a new solution which is intended for very precise time synchronization on the Ethernet networks. The IEEE 1588 supports the synchronization of distributed clocks with an accuracy of less than 1 microsecond by distributing the signal of a master clock to the slave clocks via an Ethernet network as shown in Figure 2.3. The extra dedicated cabling required for IRIG-B implementation can be eliminated and replaced by Ethernet network which is permitting the transmission of highly accurate reference timing signal of IEEE 1588.

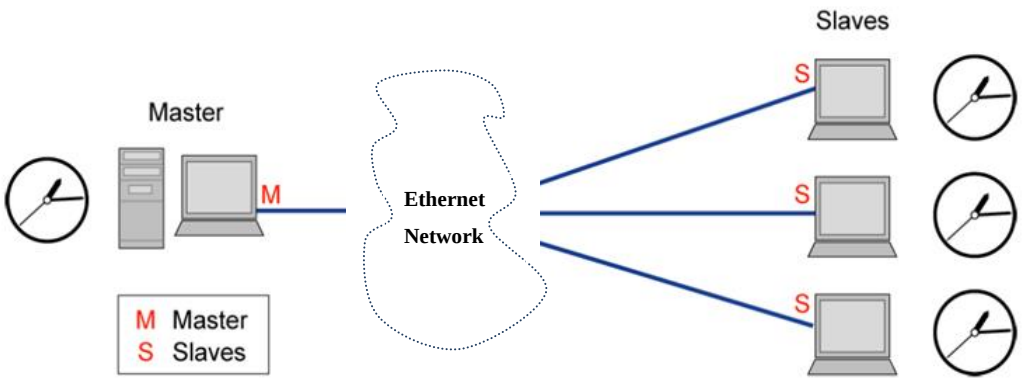


Figure 2.3. IEEE 1588 Based Precision Time Protocol

Distributing the reference time signal which can be obtained by a GPS receiver to all the measuring devices placed in the same area is a better solution. By utilizing the synchronization protocol IEEE 1588 based PTP described above; the synchronization of devices would be more favorable especially when the remote devices like IEDs are connected together by appropriate communication link. In reality, IED contains a clock that can be synchronized throughout Ethernet based LANs. This requires referencing to time that is "absolute" like UTC; the clock of the grand master is synchronized to an absolute UTC provided by the GPS receiver.

The synchronization is done through using a GPS receiver to determine the absolute time reference and then the PTP distributes the reference time signal obtained to all the measuring devices that are installed at all nodes in the electric subsystem. The redundant external Time Synchronization (TS) sources that shown in Figure 2.2 can be superseded by two master time synchronization sources (TS-1 and TS-2) which basically can provide the time synchronization signal in compliance with IEEE 1588 based PTP over the Ethernet network based Process-Bus as shown in Figure 2.4. The compatibility with IEEE 1588 based PTP can be fulfilled by employing hardware and/or software in all devices.

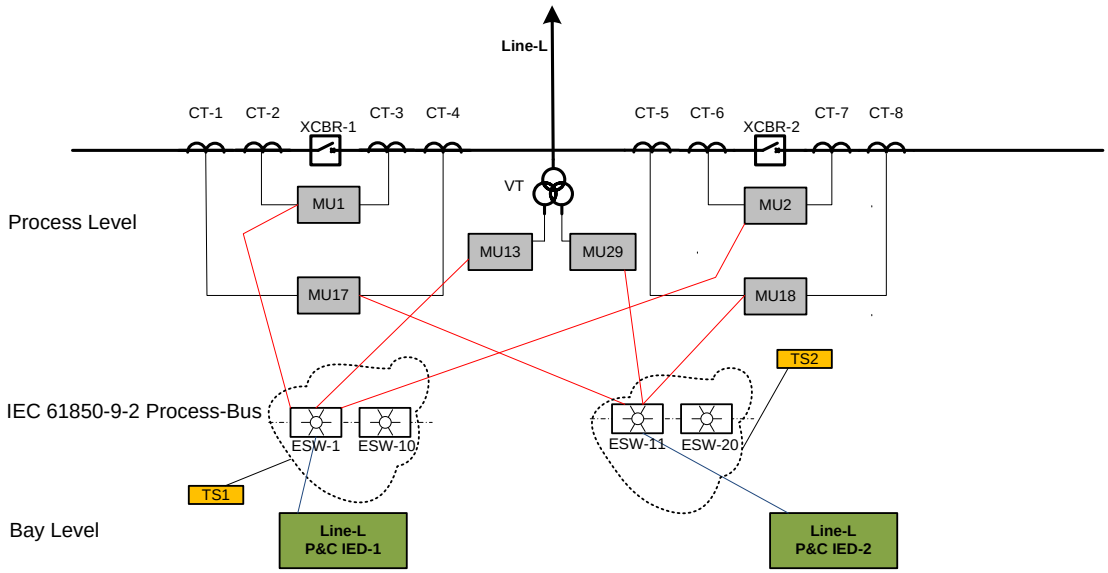


Figure 2.4. Time Synchronization Using IEEE 1588 Based PTP

2.3 SMVs Implementation

There are many existing standards that have being updated and many new others are being developed to ensure the smoothness of transition from current systems of electrical transmission to Smart Grids [84, 85]. One of these new developed standards is the IEC 61850 [26] which is had been originally developed to define the communications within any electrical substation as mentioned before. Recently, this standard being extended in order that all relevant aspects of Smart Grid, such as the communications between the electrical substations and the distributed energy resources (DERs) are covered [3].

The first published edition of the IEC 61850 standard in 2003 includes a total of 12 parts and each describes the different communication aspects that taking place in the substation automation system. IEC 61850 part-7 for example, is presenting the communication system logical view (i.e. device logical models, abstract communication service interface etc.), while parts like 8 and 9 are specifying how the logical concepts can be mapped into a specific communications infrastructure (i.e., the implementation of the communication protocol stack).

2.3.1 Traffic Generation of IEC 61980-9-2 SMVs

The IEC 61850 standard has defined 5 services of communication as shown in Figure 1.2. The first three services are defined as time-critical services and are used in substation protection and control. These services include the time-critical messages (i.e. SMVs and GOOSE) which should be mapped directly to the ISO-7 Data Link layer for increasing the performance. The other remaining two services are the time synchronized and the client-server (i.e. Manufacturing Message Specification (MMS)) which both are dealing with the time synchronization of the devices in substation, respectively. For better understanding of how the time-critical services are used in substation protection & control, let's consider again the simplified system that is shown in Figure 1.3.

The MU accepts multiple signals (analogue signals of voltage and current) which are coming from the voltage and current transformers and then, the MU acts as an analogue-to-digital-converter (ADC) to convert the signals into digital format. The digitized signals are encapsulated along with the relevant meta data to from the packets of SMV, and then casted over the Ethernet network based Process-Bus to the substation protection and control devices [86]. These protection and control devices extract and examines the data from the received packets. If any evidence of a fault in the protected zone is detected, it issues a time-critical trip message (which is encapsulated as GOOSE packet) to the circuit breaker for tripping the right switch-gear and isolating the fault. This exchange of time-critical messages is realized based on messaging model (publisher/subscriber model) in which subscriber/s can be subscribed to publisher/s, so they can be notified when a particular event is occurred.

2.3.2 SMVs Traffic Generation Function Description

The developed function for SMVs traffic generation in this research was based on the IEC 61850-9-2 LE implementation guide lines [38]. It had been created as scripts and its core was written in C programming language under Linux real-time operating system. The conceptual model of the SMVs traffic generation is described in

Figure 2.5 based on OSI-7 model. Node-A is responsible for generating SMV packets while Node-B is the sink node which can act as their addressee (receiver) and both of these nodes are implementing 3 layers (application, data link, and physical layer) out of OSI-7 layers as it is required by the IEC 61850 standard. The functionalities of the application layers in Node-A and Node-B are implemented as SMVs Publisher and SMVs Subscriber, respectively. The Publisher function is responsible to create every SMV packet consisting of a series of Application Service Data Units (ASDUs) encapsulated inside the Application Protocol Data Unit (APDU). The structure of the SMV will be presented in more details in section 2.3.5.1.

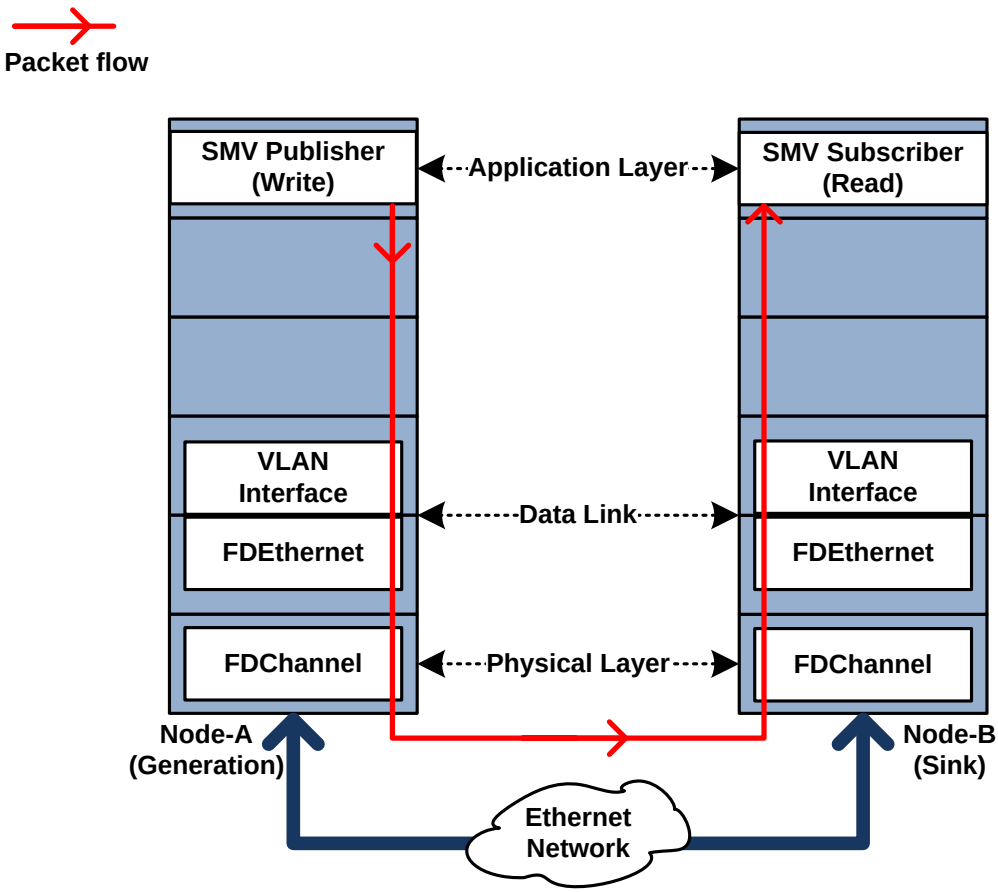


Figure 2.5. OSI-7 Layers Based SMV Traffic Generation

The APDU and ASDU byte structure is implemented based on: 1) the Abstract Syntax Notation One (ASN.1) [87] which is a standard and notations that describe rules and structures for representing, encoding, transmitting, and decoding data in telecommunications and computer networking; and 2) Basic Encoding Rules (BER) (Type Length Value (TLV)) as specified in IEC 61850-9-2. The TLV that is included in the frame of SMV APDU/ASDU is any of the optional fields (e.g. smpRate, datSet, smpMod and refrTm). These optional field values of triplets should be implemented to retain the SMV frame structure and they can be set to 0 or any other arbitrary value. Before SMV frame is sent down the lower layers, a specific header (Ether.type PDU) should be added to it. This header is 8 bytes in length and it is comprised of 4 fields (APPID, Length, reserved 1, and reserved 2).

The APPID (Application Identifier) is used to select the Ethernet frames that contain the SMV APDU, and also used to distinguish between frames of SMV and GOOSE. This APPID is equipped with 2 bytes and its two most significant bits have a value of 01 in case of SMV protocol, while the remaining 14 bits are used to represent the actual ID of a SMV publishing device in the substation communication network and by default, it should be set to 0x4000 and can range to 0x7FFF. In our function, default values of APPID and reserved 2 have been set to 0x4000 and to 0x0000, respectively.

Also, the value of reserved 1 field has been set to 0x8000 in order to indicate that a test device was used to generate the SMV packets. Length field is the actual total length of SMVAPDU (APDU+ specific header) and can be updated automatically. This length of the packet should be padded with zeroes if it is smaller than 46 bytes. This will ensure that the SMV frame is conforming to the Ethernet requirements of (i.e. the payload minimum length must be bigger than 46 bytes). On the other hand, the SMV subscriber function will act as sink for the received packets of SMV. Its functionality is to remove the header and do padding (if required) to the packet that is sent up to the application by the sub-layer of the interface.

The sub-layer of the VLAN interface has been implemented to emulate the behavior of an IEEE802.1Q virtual LAN (i.e. a node that is capable of tagging the frame before it is casted over network) [88]. It has been clearly stated in IEC 61850 that all SMV frames should be explicitly VLAN tagged. The functionality of this sub-layer has been designed to tag on or remove the tag control information (TCI) and field of the ether-type from the received packet that is delivered from next layer (upper/lower). This TCI is containing three fields: priority code point (PCP), canonical format indicator (CFI), and VLAN identifier (VID). The PCP field is 3 bits long which is used for treating the frame with the priority denoted in this field when it is travelling through EEE802.1Q switched Ethernet. The value of this PCP field can be ranged from 0 to 7 and there is no strict rule on how those values should be distributed but, by default, it should set to 4 in case of SMV-type payload according to IEC 61850. The field of CFI is one bit long which in general refers to the ordering of the bits format within the frame and precisely, it's meaning is depending on the underlying technology used (i.e. Ethernet, Token Ring, etc.).

For Ethernet based networks, these frames are typically sent with a field value set to 0. The VID field value is 12 bits long and it is used to identify the frame's VLAN explicitly. In case of SMV-type payload, this value should be set to 0x000 or otherwise, it should be set the value that is used by GOOSE protocol. The different filed values contained in the TCI are assigned at the application layer. In case of SMV Publisher, it is passed to the VLAN interface sub-layer together with the values of ether-type field and the pointer to SMV frame. The following values have been assigned in our function: PCP, CFI and VID are set to default values of 4, 0, and 0x000, respectively. Also, the field of ether-type which is used to identify the protocol within the Ethernet frame is set 0x88 BA; whereas in this case is SMV Protocol. When the SMV frame is received from the lower layer, the fields of TCI and ether-type are removed by the VLAN Interface sub-layer and then the removed ether-type field is inspected and compared with the ID of SMV Protocol for matching. If matching is found, the frame can be forwarded up to the SMV subscriber application; otherwise, it dropped at this sub-layer if there is no matching. By this mechanism, only tagged frames that carrying SMV-type payload are allowed to traffic up to the SMV

subscriber application. In next section, the implementations of the SMVs traffic generation function and how it should be mapped to the Ethernet (ISO/IEC8802-3) is described.

2.3.3 SMV Write/Read Functions Implementation

As it was mentioned before, the Process-Bus communication network works based on layer 2 of OSI-7 (the data link layer). For layer 2 based SMV traffic generations, programming of this data link layer should be accomplished. Based on the structure of OSI-7 layer communication model, the data link layer is defined as the layer that deals with the raw data traffic through the Ethernet based network. It is easier to program the transport layer protocol in C language rather than to program the traffic the data link layer. For instance, there is a command for programing the User Datagram Protocol (UDP) protocol [89] which is a transport layer protocol.

This command creates the functionality of encapsulating of structure automatically. The UDP frame structure will be generated and sent automatically by the command after assigning the data by the programmer. However, programing of the SMV frame structure needs to be from the raw frame structure. To be able to send out these raw frames, a high level permission is needed to encapsulate down to the link layer as it is shown in

Figure 2.5. Since the SMV communication traffic is sent and received through the Ethernet network, it is necessary to study Ethernet communication programming. A Simple UDP programing example presented in [90] is studied in order to understand how information can be sent and received through Ethernet work. However, the UDP is a communication service which is done based on the transport layer protocol (layer 4) while the SMV communication service is done based on the data link layer (layer 2).

To be able to program of SMV service, the data link layer needs to be to studied and understood first. Again, to implement the SMV service, an example of Address Resolution Protocol (ARP) [91], which is the protocol for data link layer communication is used in the

first step. With this implementation of ARP application, the sending of SMV can be done just by doing some modifications to the example code in [91].

2.3.3.1 Implementation of UDP

The UDP is a communication transport layer protocol for exchanging messages between applications in a network which is defined for use with the Internet Protocol (IP) [89]. These applications are typically named “Server” and “Client”. The “Client” exchanges the information with the “Server” by sending its massages through an Ethernet port. Since our aim is to send the SMV through the data link layer traffic, the concentration will put to program a "client" (sending node). For this purpose, two key functions ("socket()" and " sendto()") will be used for transmitting the UDP messages through the Ethernet port.

2.3.3.1.1 "socket()" function

This function is used by the application for sending or receiving packets and to do other socket operations. When the socket() function is called, it will open a socket communication and returns a descriptor. This function will return the file descriptor on success, -1 if any error occurred during the execution and errno if it is set appropriately. The calling format of this function is as following:

[1]int socket(int *domain*, int *type*, int *protocol*);

The *domain* parameter specifies the protocol family which will be used for communication. In this implementation, this parameter is set to **AF_INET**, which indicates that the address family is the Internet Protocol version 4 (IPv4). The function has the indicated *type*, which is use for specifying the communication semantics. It has been set to **SOCK_DGRAM** in this implementation which supports datagrams. The third

[1] http://www.tutorialspoint.com/unix_system_calls/socket.htm

parameter is *protocol* which specifies the particular protocol that to be used with the function. For UDP protocol, this parameter is **IPPROTO_UDP**.

2.3.3.1.2 "sendto()" function

This function is used to transmit a message to another socket. The format of calling this function as following:

[2]ssize_t sendto(int *s*, const void **buf*, size_t *len*, int *flags*, const struct sockaddr **to_addrs*,…
socklen_t *to_len*);

The parameter *s* is the file descriptor of the sending socket. This message is found in *buf* and has length *len.* The *flags* parameter is the bitwise OR of zero in case of UDP and it is used to specify the message transmission type. The parameter *to_addrs* points out to the sockaddr structure, where the destination address is stored. The parameter *to_len* is used to specify the length of the destination sockaddr. The function will return the number of characters sent when successed, otherwise; it will return -1.

With these two key functions ("socket()" and "sendto()"), the UDP messages can be transmitted through the Ethernet port. It is worth to mention that, the data like Ethernet header and the protocol tag are also included in the complete UDP packet with the message to be sent. This encapsulation of data can be done automatically for UDP since it is a transportation layer protocol, but for the data link layer; it cannot do so. As a result, the way that the data link layer is encapsulated needs to be studied with the help of the ARP example proposed in [91].

2.3.4 Implementation of ARP Traffic

It is necessary to encapsulate the data that is required to be sent out, before calling the two functions “socket()”and “sendto()” in order to be able to transmit the data link layer

[2] http://www.tutorialspoint.com/unix_system_calls/send.htm

traffic out. The ARP example programming code in [91] will help to understand the way of the data encapsulation. Based on this example, the ARP traffic will be implemented as shown in Figure 2.6. The major difference between the UDP example program and the ARP example program is that, the "sendto()" function in ARP example is not encapsulating the Ethernet packet automatically, which needs to be manually encapsulated according to the ARP protocol format.

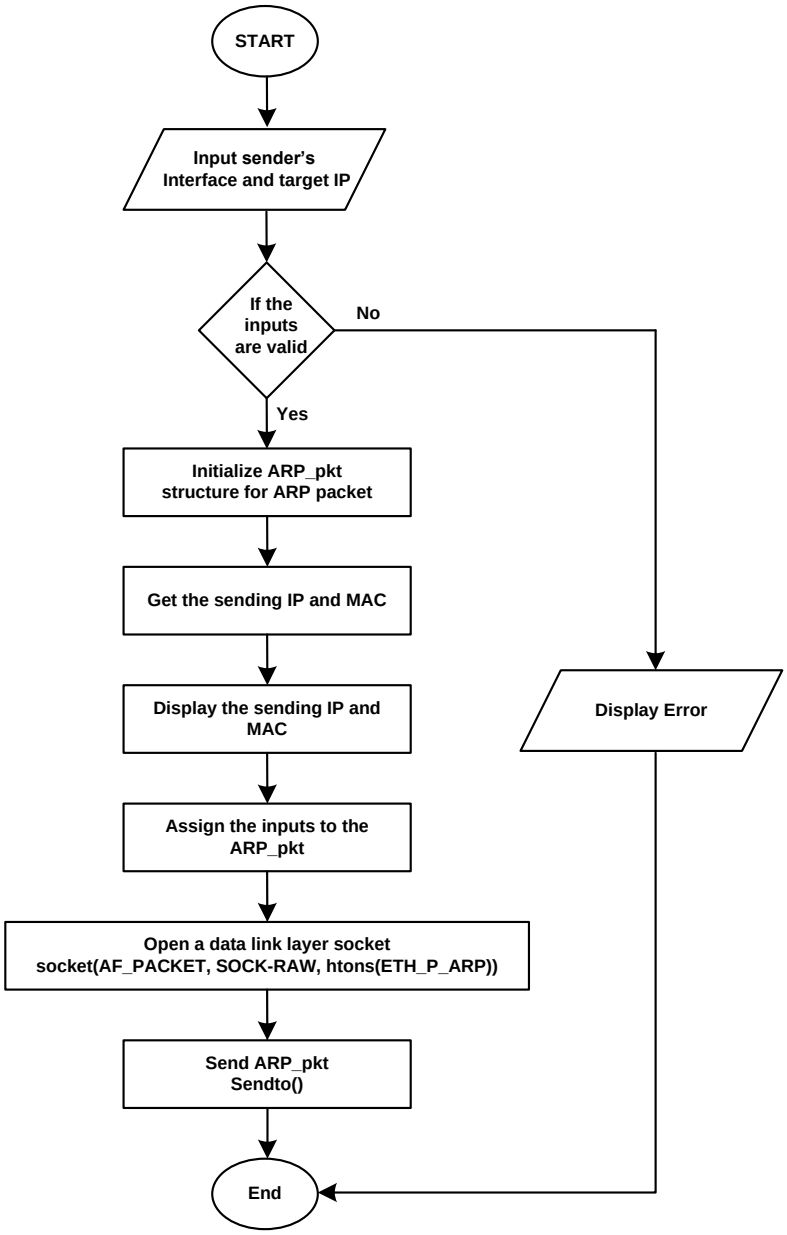


Figure 2.6. Flow Chart for Implementing ARP Traffic

The ARP packet format is defined [92] as follows:

Ethernet transmission layer (not necessarily accessible to the user):
48.bit: Ethernet address of destination
48.bit: Ethernet address of sender
16.bit: Protocol type = ether_type$ADDRESS_RESOLUTION
Ethernet packet data:
16.bit: (ar_hrd) Hardware address space (e.g., Ethernet, Packet Radio Net.)
16.bit: (ar_pro) Protocol address space. For Ethernet hardware, this is from the set of type fields ether_typ_<protocol>.
8.bit: (ar_hln) byte length of each hardware address
8.bit: (ar_pln) byte length of each protocol address
16.bit: (ar_op) opcode (ares_op_REQUEST | ares_op_REPLY)
nbytes: (ar_sha) Hardware address of sender of this packet, n from the ar_hln field. mbytes: (ar_spa) Protocol address of sender of this packet, m from the ar_pln field.
nbytes: (ar_tha) Hardware address of target of this packet (if known).
mbytes: (ar_tpa) Protocol address of target.

According to that, the structure of the ARP packet is defined exactly as it is defined in [92] as follows:

```
struct ARP_pkt

{
/*Header MAC*/
               int8_t dest_mac[6];/*Destination MAC Address*/
               int8_t src_mac[6]; /*Source MAC Address*/
               int16_t Ethertype; /*0x88BA for Sampled Value*/
/*ARP Structure*/
               uint_16_t ar_hrd   /*Hardware address space*/
               uint_16_t ar_pro   /*Protocol address space*/
               uint_8_t  ar_hln   /*byte length of each hardware*/
               uint_8_t  ar_pln   /*byte length of each protocol*/
               uint_16_t ar_op    /*opcode*/
               uint_8_t  ar_sha   /*Hardware address of sender of this packet*/
               uint_32_t ar_spa   /*Protocol address of sender of this packet*/
               uint_8_t  ar_tha   /*Hardware address of target*/
               uint_32_t ar_tpa   /*Protocol address of target*/
}
```

For sending out the ARP packets throughout the socket, the ARP_pkt pointer will be assigned to the “sendto()” function as a “*buf” argument and also all arguments need to be changed in order to enable the “sendto()” function of sending the ARP_pkt which

eventually will be sent out through the Ethernet port and recognized as a format of ARP packet.

2.3.5 SMV Write Function Implementation

Since the SMV traffic is accomplished based on the data link layer protocol, the “socket()” and “sendto()” function arguments can be reserved the same and only the structure of the ARP_pkt is need to be modified . This structure has to be modified to the stranded format of SMV packet as defined [39]. For implementing the function of SMV traffic, the implementation guideline [38] will be used as a reference for defining the SMV packet structure. In addition to the task of constructing the SMV packet format, the function will have another task which is controlling the SMV packet frequency rate.

As it was mentioned in section 1.2, two standardized sample rates defined in the IEC 61850 standard with one sample rate equals to 80 samples per cycle for basic substation protection and control, and the other has a high rate of 256 samples per cycle for power quality and measurements. Whereas the goal of this research is to use the SMV traffic generation for protection and control in substation, a frequency rate of 80 samples per cycle is considered in this function.

2.3.5.1 Formatting the SMV packet

The guideline in [38] is a practical guidance which is aiming to help the vendors for accelerating the introducing and realising of the IEC61850-9-2 standard to the markets. It was prepared by people from multi-vendors (including major players like Siemens and ABB). This document will definitely save us a lot of efforts and time for implementing the SMV traffic function. In this guideline, the frame format of SMV had been defined as shown in Figure 2.7.

As it can be noted from Figure 2.7, every field had been well defined and assigned a value. Also, the field APDU is separately defined in more details which is shown in Figure 2.8. According to the definition of the APDU structure, all variables have to be encoded

according to the basic encoding rules of the Abstract Syntax Notation One (ASN.1) [87] that is following the same pattern (i.e. tag_length_values). The first two parts (tag and length) of ASN.1 typically take two bytes (one byte each) while, the third part (values) can take value of multiple bytes which depends on how many values are included. All field values are expressed in hexadecimal. For example, the field "smpCnt" has the number 0x82, which means that the ASN.1 tag for "smpCnt" will be 82 in hexadecimal and the occupying memory is 1 byte.

After the tag 0x82, there is a letter "L" to indicate the length and a bracket with number "2" in decimal[3] under it which means that the "smpCnt" length is 2 bytes. It can also be seen in Figure 2.8 that, each one of APDU can consist of one or more of ASDUs (up to 8). The ASDU is the data unit (or the sink) where the SMVs and other related information are stored. The SMVs are stored in the "DataSet" field. The [38] also had defined the detailed format of the “DataSet" as shown in Figure 2.9.

[3] This decimal number should be converted to a hexadecimal number during the programming of the function.

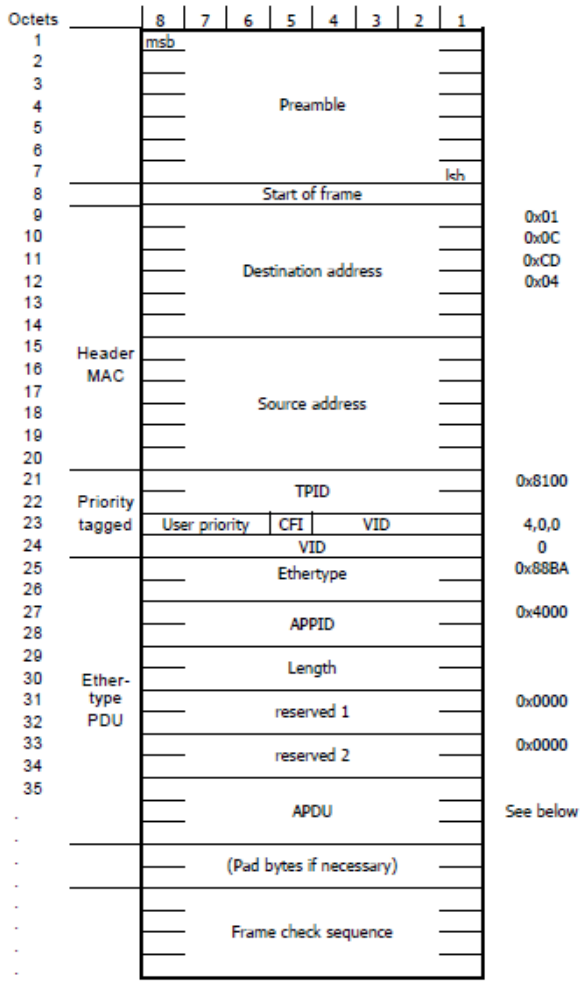


Figure 2.7. SMV Frame Format [38]

Each variable has a meaning which can be recognized easily by just observing its name. For instance, the "UnnATVTR1.Vol.instMag.i" means $Voltage_{nn}$ of Phase A (U_{nn} V) from the logic node TVTR1, the unit is in Voltage (V) and it is expressing the instant magnitude of the voltage (instMag.i). Also, the variable (.q) showed in "InnATVTR1.Vol.q" means the quality of logic node TVTR1 (quality of the measured value). As it can be seen in Figure 2.9, there are 8 instant values (each variable takes 4 bytes) and 8 quality identifiers and normally, these eight values are assigned to be the three-phase voltage (or current) with neutral.

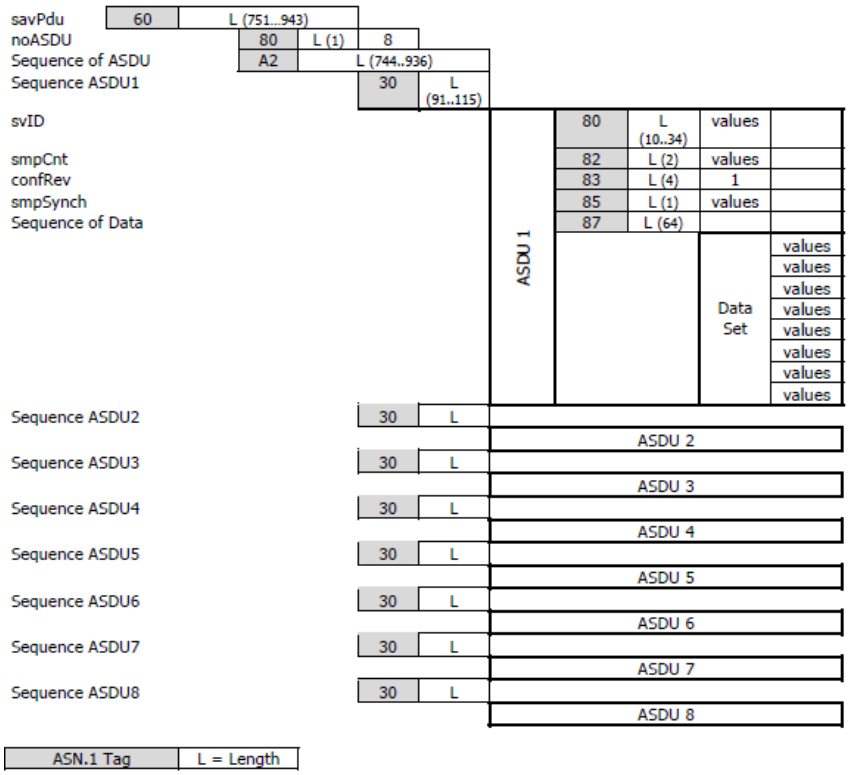


Figure 2.8. Structure of APDU in SMV Frame [38]

According to these definitions, the structure of the SMV frame has three layers which are depicted in Figure 2.10. These layers are:

I. 1st layer-SMV Ethernet frame.

II. 2nd layer-APDU (Application Protocol Data Unit).

III. 3rd layer-PhsMeas.

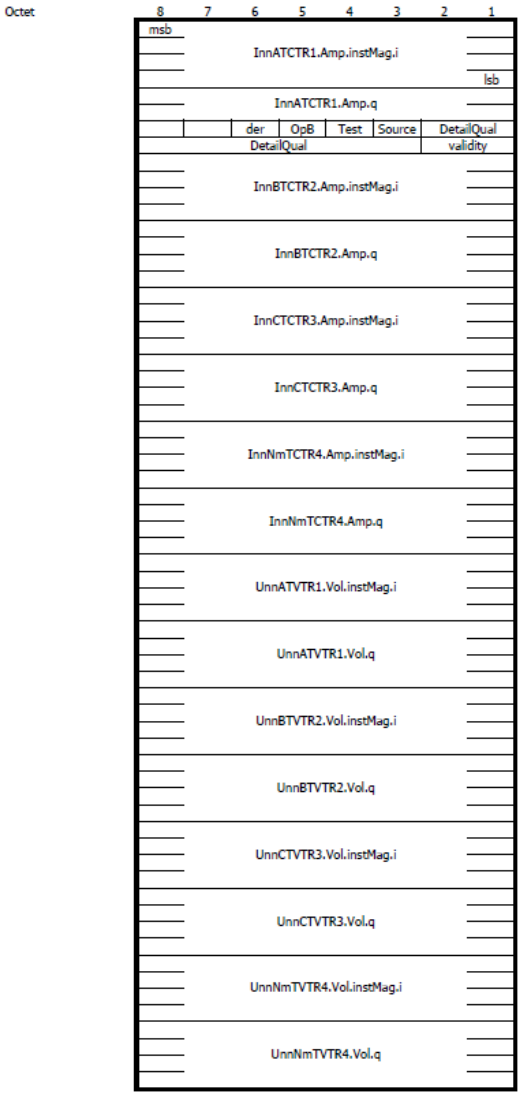


Figure 2.9. "DataSet" Definition [38]

From Figure 2.10, it is worth to note that, there is another layer (the Data Units layer) included in the APDU which makes eventually the structure of the SMV four layers as follows: The SMV Frame structure has one APDU. The APDU is consisting of one or more (Up to eight) ASDUs. The ASDU is the data unit where the measured values and other related information can be stored. The measured values of voltage or current signals are saved in the field "DataSet". The total size of the SMV is about 1000 bytes when one

ASDU is assigned. Every addition of one ASDU can add 124 Bytes to the total size of the SMV Frame.

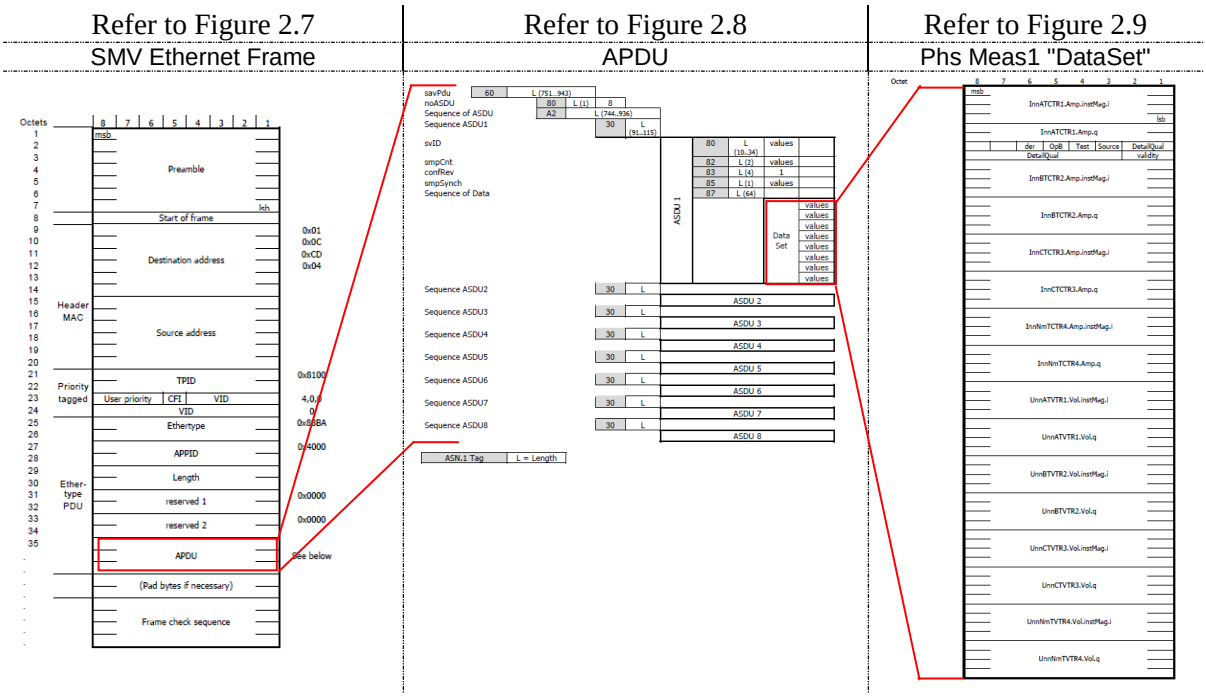


Figure 2.10. Layers Structure of IEC61850 SMV Frame

So, according to all of these definitions, the structure of the first layer of SMV frame can be implemented as follows:

```
struct SMV_FRM {
                /*Header MAC*/
                int8_t destination_mac[6];
                int8_t source_mac[6];
                /*Start Priority tagged*/
                int16_t TPID;// For 802.1Q VLAN, its 0x8100
                int16_t TCI;//0x4000
                /* Start Ethertype PDU*/
                int16_t Ethertype;/* For SMV, its 0x88BA*/
                int16_t APPID;/*0x4000 application identifier*/
                int16_t Length; /* After APPID*/
                int16_t reserved1;
                int16_t reserved2;
                /*Adding the SMV_APDU (the application protocol data unit)*/
                struct SMV_APDU SMV_apdu;
                /*pad bytes if necessary*/
                /*Frame check sequence*/
        }__attribute__((__packed__));
```

The value of the destination MAC address should be assigned in the range of "01-0C-CD-04-xx-xx". At the end of this structure, a new structure (2nd layer) for the APDU is added. This layer can be defined as follow:

```
struct SMV_APDU {
                int8_t savPDUtag;
                int8_t savPDULength;/*sizeof SMV_APDU*/
                int8_t noASDUtag;
                int8_t noSMVASDUngth;/*sizeof noASDU*/
                int8_t noASDU;
                int8_t SequenceofASDUtag;
                int8_t SequenceofSMVASDUngth;/*sizeof all ASDU*/
                struct SMV_ASDU SMV_ASDU1;
        }__attribute__((__packed__));
```

As it is mentioned before, the structure of the ASDU layer is taken out separately and the APDU layer could contain one or more ASDUs (up to eight). At the end of the APDU structure, the structure ASDU (3rd layer) is defined and of only one ASDU named as SMV_ASDU1 is added. The structure of SMV_ASDU is defined as follows:

```
struct SMV_ASDU {
                int8_t SequenceASDU_tag;/*0x30*/
                int8_t SequenceSMV_ASDUngth;/*0x5B*/
                int8_t svID_tag;/*0x80*/
                int8_t svID_Length;/*0x0A*/
                int8_t svID[11];/*According to the IEC 61869-9 naming rules */
                int8_t smpCnt_tag;/*0x82*/
                int8_t smpCnt_Length;/*0x02*/
                int16_t smpCnt;/* Samples Counter */
                int8_t confRev_tag;/*0x83*/
                int8_t confRev_Length;/*0x04*/
                int32_t confRev;/*Configuration revision number*/
                int8_t smpSynch_tag;/*0x85*/
                int8_t smpSynch_Length;/*0x01*/
                int8_t smpSynch; /* synchronization, 2-> global 1-> local 0-> not sych*/
                int8_t smpRate_tag;/*0x86*/
                int8_t smpRate_Length;/*0x02*/
                int16_t smpRate;/* 80 samples per cycle (0x0050 hex)*/
                int8_t SequenceofData_tag;/*0x87*/
                int8_t SequenceofData_Length;/*0x40*/
                struct SMV_dataset DataSet;
        }__attribute__((__packed__));
```

According to the IEC 61869-9 naming rules, the "svID" should not take more than 11 bytes. In this implementation, the name for "svID" is chosen to be "_AHMED_SMV_".

The American Standard Code for Information Interchange (ASCII) [93] had been used for encoding the "svID" letters whereas each character takes one byte to be represented.

The 4th layer is the dataset which is defined as " SMV_dataset ".

```
struct SMV_dataset {
        /* Instantaneous magnitude of Current (Phase A, B, C and neutral) in Ampere*/
                int32_t InnATCTR1_Amp_instMag_i;
                int32_t InnATCTR1_Amp_q; /*quality*/
                int32_t InnBTCTR2_Amp_instMag_i;
                int32_t InnBTCTR2_Amp_q;
                int32_t InnCTCTR3_Amp_instMag_i;
                int32_t InnCTCTR3_Amp_q;
                int32_t InnNTCTR4_Amp_instMag_i;
                int32_t InnNTCTR4_Amp_q;
       /* Instantaneous magnitude of Voltage (Phase A, B, C and neutral) in Volt*/
                int32_t UnnATVTR1_Vol_instMag_i;
                int32_t UnnATVTR1_Vol_q; /*quality*/
                int32_t UnnBTVTR2_Vol_instMag_i;
                int32_t UnnBTVTR2_Vol_q;
                int32_t UnnCTVTR3_Vol_instMag_i;
                int32_t UnnCTVTR3_Vol_q;
                int32_t UnnNTVTR4_Vol_instMag_i;
                int32_t UnnNTVTR4_Vol_q;
        }__attribute__((__packed__));
```

All the voltage and current values have to be scaled according to [38]. The scalar defined in [38] is 0.01 for voltage and 0.001 for current values. For instance, if the measured value of voltage is 1 Volt for phase A then, the value that should be assigned to "UnnATVTR1_Vol_instMag_i" is 1/0.01=100. The scaled value should be converted in 4 bytes hexadecimal (i.e. 0x00000064), so that the protection device (IED) can recognize that the measured value of voltage is 1 Volt. Also, to make the traffic of SMV frames that is encapsulated by the function recognized by the IED; the function should send them in standardized sample rate. As previously mentioned, there are two rates for SMV sending: 80 and 256 samples per nominal cycle. Whereas the goal in this research is the employing of SMV for protection purpose in power substation, 80 samples per cycle is considered.

For 80 samples per cycle, the SMV packet sending frequency "F_s" is 4 kHz (i.e. 80*50 = 4000 samples per second).To do so, the function "sendto()" should be used 4000 times a second. The variable "smpCnt" that is defined in the SMV frame structure is used as counter to count from 0 to 3999 (i.e. 4000 samples) and it should be increased by1 for each

sent of SMV. For nominal power frequency of 50 Hz and to realize F_s of 4000Hz, the counter takes a time period of one second to count from 0 to 3999 or in other words, it takes 250 µsecond (1/4000 second) to send a new SMV sample. To satisfy the requirement of sending rate for the SMV packet, the functions "gettimeoftheday()" and "usleep()" are used. The structures (syntaxes) of calling them are given in details below:

[4]`int gettimeofday(struct timeval *tv, struct timezone *tz);`

[5]`int usleep(useconds_t usec);`

The function "`gettimeofday()`" is used to get the time coupled with the timezone while the function "`usleep`" is used to suspend the execution of the calling thread for "usec" (at least µseconds). The structure of the tv argument is a struct "`timeval`" and is defined in the header file <sys/time.h> as follows:

```
struct timeval {
    time_t      tv_sec;     /* seconds */
    suseconds_t tv_usec;    /* microseconds */
};
```

When the "gettimeofday()" function is called, the seconds will be stored in the variable "`tv_sec`" and the µsecond is stored in "`tv_usec`".

With the help of the functions introduced above, the SMV publisher function is designed and programmed to send out the SMV with F_s equals to 4 kHz. The capture of SMV Frames in Wireshark [94] is shown in Figure 2.11. The complete code for writing this SMV frames is found in Appendix A.

[4] http://linux.die.net/man/2/gettimeofday

[5] http://linux.die.net/man/3/usleep

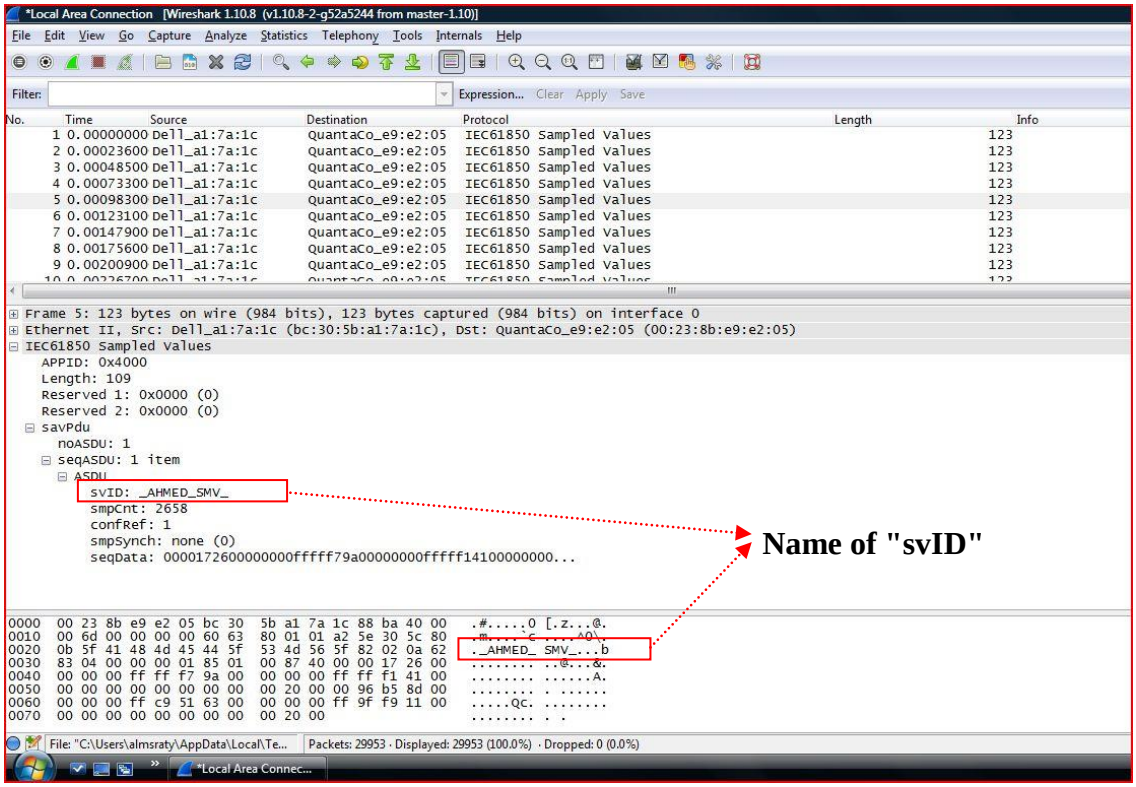


Figure 2.11. Capture of SMV Frames in Wireshark

2.3.6 SMV Read Function Implementation

This function implemented based on the ARP receive example presented in [91]. The main functionalities are to read the SMV frames and unwrapping the data included. In addition, the function contains subroutine which is responsible to estimate the phasors with a reporting rate of 100 frames per second according to IEEE C37.118 [95]. For the purpose of reading the SMV frames, the key function "`recv()`" is used. The format of calling this function as following:

```
[6]ssize_t recv(int socket, void *buffer, size_t length, int flags);
```

[6] http://linux.die.net/man/3/recv

This function is normally used with connected sockets to receive a message from a connection-mode or connectionless-mode socket. The function takes the following arguments:

"socket" specifies the socket file descriptor, "buffer" points to the buffer where the received message should be stored, "length" which specifies the length (in bytes) of the buffer pointed to by the buffer argument and "flags" specifies the type message reception. Next step is to unwrap the data included in the frame. To do so, the functions "htonl()", "htons()", "ntohl()" and "ntohs()" which are used to convert values between host and network byte order as follows:

The "htonl()" function is used to convert unsigned integer (*hostlong*) from host byte order to network byte order, the "htons()" function is also used to convert unsigned short integer (*hostshort*) from host byte order to network byte order, the "ntohl()" function is used to convert unsigned integer (*netLong*) from network byte order to host byte order and the "ntohs()" function is used to convert the unsigned short integer (*netshort*) from network byte order to host byte order. These functions are used to unwrap the data encapsulated in the SMV frame. For instance, the "Dataset" are unwrapped as follows:

```
// Print out contents of Dataset (Measurments).
   printf ("Current A= %d\n", ntohl(FRM->SMV_APDU.SMV_ASDU1.DataSet.InnATCTR1_Amp_instMag_i));
   printf ("Current B= %u\n", ntohl(FRM->SMV_APDU.SMV_ASDU1.DataSet.InnBTCTR2_Amp_instMag_i));
   printf ("Current C= %u\n", ntohl(FRM->SMV_APDU.SMV_ASDU1.DataSet.InnCTCTR3_Amp_instMag_i));
   printf ("Current N= %u\n", ntohl(FRM->SMV_APDU.SMV_ASDU1.DataSet.InnNTCTR4_Amp_instMag_i));
   printf ("Voltage A= %u\n", ntohl(FRM->SMV_APDU.SMV_ASDU1.DataSet.UnnATVTR1_Vol_instMag_i ));
   printf ("Voltage B= %u\n", ntohl(FRM->SMV_APDU.SMV_ASDU1.DataSet.UnnBTVTR2_Vol_instMag_i ));
   printf ("Voltage C= %u\n", ntohl(FRM->SMV_APDU.SMV_ASDU1.DataSet.UnnCTVTR3_Vol_instMag_i ));
   printf ("Voltage N= %u\n", ntohl(FRM->SMV_APDU.SMV_ASDU1.DataSet.UnnNTVTR4_Vol_instMag_i));
*/
```

Once the SMV data is unwrapped, the function can use it to estimate the phasors. The complete code for unwrapping and implementing of phasor estimation function can be found in Appendix B.

2.4 Integration with TrueTime Environment

TrueTime is a Matlab/Simulink-based simulator for real-time systems, developed at Lund University, Sweden. It facilitates co-simulation in real-time of task execution and network transmission dynamics by providing a Simulink block library and a collection of MATLAB MEX functions.

There are many ways to integrate C code with Matlab environment and interface with TrueTime simulation environment. In this thesis, two solutions are proposed to do that as follows:

2.4.1 Integration by "S-function"

"S-function" provides a help to the users in such way that they can run an external C code as a function block in Simulink simulation environment by calling the "S-function builder". The "S-function builder" block can help users to build an "S-function" very easily. The C code is embedded into "S-function builder" block to make SMV frame creator. Then, this block will provide the SMV to the TrueTime which act as IEC 61850 Process-Bus interface.

Disadvantage of using the "S-function" is the whole simulation can be slowed down greatly. The "S-function" is called 4000 times per second and in each call, the external C code (i.e. the SMV read/write function) will be run once by the "S-function". To avoid such slowdown in simulation speed, the whole Simulink model including the "S-function builder" should be converted to a Matlab's xPC target; so it can run in real-time on a normal PC.

2.4.2 Integration by "mex-command"

The C source external code (includes: SMV creator and Process-Bus interface) can be compiled into a one binary shared library by calling the syntax "mex C_file_name" in Matlab and called by TrueTime. The run of "mex" in Matlab will build an executable for

standalone Matlab engine and MAT-file applications. The application of "MAT- file" of "mex" command can be used to solve the problem of embedding C code with TrueTime and reading values from Simulink model. As with the case with the integration by "S-function", the whole model here should be also converted to a Matlab's xPC target; so it can run in real-time.

2.5 Summary

The significant features of the IEC 61850-9-2 based Process-Bus, such as the multicasting/retransmission of time critical messages (SMV/GOOSE), the time synchronization over the Process-Bus, and the fast Ethernet switched based networks are presented in this Chapter. Moreover, a brief overview of many architectures for practical Ethernet switched networks as suggested by the IEEE (PSRC report) [96], which include cascaded architecture, ring architecture, star-ring architecture and redundant-ring architecture. The key time synchronization techniques, such as the IRIG-B and the IEEE 1588 based PTP protocol time synchronization over Ethernet based network are also presented in this Chapter. The implementation procedures that have been used to create the SMVs publisher/subscriber functions are explained in detail in this chapter. The programming for the encapsulation of different layers of SMV frame in C programming under Linux real time system and how the Process-Bus interface was integrated with Matlab environment are also documented in this Chapter.

Chapter 3

3. Implementation of Phasor Estimator Using Unscented Kalman Filter and Studying Frequency Response Based On SMVs

To benefit from the raw IEC 61850-9-2 SMVs streamed with high sampling rate from various Merging Units at substation bay level, Unscented Kalman Filter (UKF) Model-Based for phasor estimation is presented in this Chapter. The model implementation and studying the frequency response of proposed one (UKF) and a comparison between it and the classical estimator (DFT) are also presented in this Chapter.

3.1 Introduction

The purpose of the state estimator is to determine state variables that represent the best match of a power system model and measurements obtained at various points in the system. This computation process is affected by the accuracy and availability of the measurements. Traditionally, Supervisory Control and Data Acquisition (SCADA) system provides all measured quantities as well as statuses of all switches in the system. However, these measurements are not synchronized which presented a limitation when used in the system state estimation. Typically, the snapshots of a measurement are done every three seconds and all measurements are sequentially collected.

Due to the measurements limitations these samples will have time skew errors which will affect the state estimation accuracy. The conventional static state estimation methods [97], [98] are not designed to capture the dynamic state changes in power system state variables (e.g. bus voltage magnitude and phase angle). Nowadays, dynamic state changes in a system can be acquired by direct means via dedicated Phasor Measurement Units (PMUs). This technology is capable to track any dynamic changes in the system with eliminated time skew errors through using PMUs that are synchronized using Global Positioning System (GPS) time information.

The concept of utilizing phasors in computing power quantities back dates to 1916 when Proteus Steinmetz published a paper on a mathematical technique for analyzing AC networks [99]. In 1992, Jay Murphy of Macrodyne developed Proteus's technique into the estimation of synchronized phasors to an absolute time reference and introduces his Phasor Measurements Unit (PMU). Later, tests have been conducted on this unit [100, 101], which determined the accuracy as 0.01 degree for phase angle at 60 Hz and 0.1% for the magnitude. The time-synchronized phasor has been firstly defined and codified in the IEEE 1344 and later to be voted as IEEE C37.118 in 2005 [102]. In this Standard, the parameters deviation of the input power signals has been restricted by allowable error over a range of operating conditions. This allowable error describes the phasor difference between an estimated and a theoretical value. This difference is calculated as a vector difference in percent and named Total Vector Error (TVE).

It's worth mentioning that the accuracy defined in IEEE C37.118 standard is only for stationary signals with exception of step functions for transient. The vector difference is only valid in steady-state conditions since the standard in its current form is only defined for steady-state systems. In these systems, the steady-state conditions assume no faults, line trippings or power swings. As it was mentioned in section 1.1.1, the sequence of events of the 2003 blackout that occurred throughout parts of the USA and Canada started with incorrect telemetry provided to an inoperative state estimator. It is reported in the examinations of the blackout that, some PMU types are not well protected against abnormal inputs and parasitic oscillations.

The operation of PMUs relies on steady-state conditions and only slow oscillation changes in the power flow can be considered. The PMU experiments [103] illustrate that frequency burst was recorded, this burst was due to using a wrong filter algorithm for estimating the system's phasors. The classical phasor estimators rely on the Discrete Fourier Transform (DFT) algorithm [15, 16], which can produce considerable magnitude and phase errors that increases the TVE under certain conditions. To overcome the DFT limitations, a high resolution spectral method like frequency analysis has been proposed [104]. An overview of the PMU applications in state estimation has been presented in

[105]. Conversely to DFT, KF has been recognized as one of the most powerful algorithms for phasors estimation at burst changes with less error [17-23]. KF is a mathematical technique, which is extensively used to compute optimal estimates of dynamic system states. The estimates are optimal in the sense that estimation errors are minimized in the least-squared sense. To avoid the drawbacks of traditional PMU and supersede it with robust one, this Chapter proposes the use of Kalman filter i.e. a Linear Quadratic Estimator (LQE) to estimate the phasors at bay.

The technique relies on employing the IEC 61850 Sampled Measured Values streamed from the substation Process- Bus. These time-critical streamed values have sampling rates of 4 kHz or 12.8 kHz, which can be used to facilitate tracking of rapid changes in input signals within TVE values defined by IEEE C37.118.

Today practice of using stand-alone PMUs for dynamic measurements can be replaced by computation of Phasor using IEC 61850-9-2 Sampled Measured Values [39] streamed from various Merging Units (MUs) at Process-Bus as suggested in [106]. This can be done at a central substation process computer for all measurement points in a substation. The chapter aims at evaluating the performance of this alternative: Phasor Estimator based on IEC 61850-9-2 communication network. For this purpose, we utilize Unsented Kalman Filter (UKF) to perform the estimation since it is recognized for years as one of the most powerful dynamic estimation technique [25], [107], [108] .

3.2 Power System Transients

Since the IEEE C37.118 was introduced in 2005, many test procedures and routines have been developed for PMUs for steady state signals where all parameters remain constant [109-111]. In addition, numerous tests were conducted according to the standard [112-114] for signals in steady state and step tests. In accomplishing these, step change was applied to considered dynamic parameter related to magnitude, angle or frequency. In evaluating the performance of PMUs, test signals may be divided into two groups. Group#1 represents electro-mechanical transient signals. The signals of this test group can be

handled by PMUs without any problems and the measurements provided should have small errors. On the other hand, Group#2 represents electro-magnetic transient signals. The signals can be used to understand how the electro-magnetic transients are filtered by the PMUs and also can be used to check if any errors in the measurements are produced.

For evaluating of any PMU, the key issue is to find out the accuracy depending on test signals. Test signals can be generated by: 1) Functions defined mathematically and programmed in any programming language, 2) Using dedicated software, 3) Recording real-transients by oscillograms. It's easier to work with signals defined mathematically as they are clean and their representation is easy to understand. In either case, the testing signals have to be chosen adequately to present the transient of the power systems. The produced signals are signals with step-wise change in amplitude or phase, frequency-ramped signals, and amplitude-phase modulated signals.

Although records from oscillograms represent real case transient, they are not easy to be attained, not easy to interpret, and can be noisy. Test signals produced by simulating software seem most promising as they can represent the transient's phenomena well enough and they can be produced in any quantities. Also, electro-mechanical processes can be simulated entirely by using simulation software (e.g. MATLAB), which produces results as phasors and can control the step changes in magnitude and phase perfectly. Thus, phasor estimator can be evaluated comparing the phasors produced by the estimator and the simulated phasors.

3.3 PMU Based On IEC 61850-9-2

For purposes of estimating phasors in substation, the practice of using stand-alone PMUs for producing phasors at process level can be replaced by KF function at bay level. The KF can accomplish the function of estimation for all SMVs received from measuring points (i.e. MUs) in the substation. Estimating the phasors at bay level can help in reducing the cost of installing new dedicated physical PMUs at each measurement point or may supersede the PMUs installed at process level. Computation of phasors using KF can be

accomplished by updating its states by the time-critical IEC 61850-9-2 Sampled Measured Values streamed from the various MUs at substation Process-Bus. In this way, the changes in amplitude and phase angle during transient phenomena in substation can be well tracked. The KF model should satisfy the phasor measurement allowable errors according to IEEE C37.118 standard. The IEEE C37.118 standard proposes several reporting rates (frames per second) with maximum reporting rates of 60/sec at 60 Hz nominal power frequency. Higher rates of 100/sec and 120/sec have been encouraged.

It may be difficult to report at higher rates with the classical phasor estimator. Classical phasor estimators employ DFT to estimate phasors. When DFT is employed for estimating signals in real-time, time windowing should be considered. Results of DFT can be considered accurate only if the following assumptions are satisfied: 1) Periodic and stationary waveforms, 2) The sampling frequency should be greater than twice the highest frequency of the signal, 3) Number of periods in each time window should be integer, 4) Each frequency in the signal should be integer multiple of the frequency resolution read out by the time windowing.

DFT is not suitable for time-frequency studies because of the time and frequency resolutions trade-off. The functionality and the accuracy of the estimated phasor depend on how many samples (N) per cycle (window size) are used. Also to increase the reporting rate, the time window size should be decreased. Decreasing time window size increases the time resolution, but frequency resolution is decreased and vice-versa. Instead of using DFT, KF estimates recursively with every new sample received. With N samples, there are N phasor estimations.

3.4 Kalman Faltering For Phasor Estimation

The Kalman Filter is a mathematical technique which is extensively used to compute the optimal estimates of a dynamic system's states. The estimates by this filter are optimal as the estimation errors are minimized in the least-squared sense [115]. The dynamic system can be described and its states estimate change with time by using a set of linear

differential equations that express the process model (in vector and matrix format) as follows:

$$\dot{\mathrm{x}} = \mathbf{F}\mathrm{x} + \mathrm{q}\,, \tag{3.1}$$

where, x is the state vector; F is the system matrix and q is a random driving noise vector. Suppose further that we have set of measurements and these measurements are related to system state by the following liner equation:

$$\mathrm{z} = \mathbf{H}\mathrm{x} + \mathrm{r}\,, \tag{3.2}$$

where, z represents the measurement vector, H represents the measurement matrix and r a random vector represents noise in measurements. All of x ,**F**,q ,z ,**H** and r are expressed in continuous-time at present. For computing the states of the system at particular instant in time k, the equation (3.1) and equation (3.2) can be transformed to discrete-time model as follows:

$$\mathrm{x}(k+1) = \mathbf{\Phi}(k)\mathrm{x}(k) + \mathrm{q}(k)\,, \tag{3.3}$$

$$\mathrm{z}(k) = \mathbf{H}(k)\mathrm{x}(k) + \mathrm{r}(k)\,, \tag{3.4}$$

where, $\mathrm{x}(k)$ is the system state vector at time instant k, $\mathbf{\Phi}(k)$ is the state transition matrix. This matrix describes the change of system states from time instant k to $k+1$. $\mathrm{z}(k)$ and $\mathbf{H}(k)$ are the measurement vector and the measurement matrix at time instant k, respectively. One way of finding $\mathbf{\Phi}(k)$ is to evaluate the Taylor-series expansion [116] as follows:

$$\mathbf{\Phi}(k) = \mathbf{I} + \mathbf{F}\Delta\mathrm{t} + \frac{(\mathbf{F}\Delta\mathrm{t})^2}{2!} + \frac{(\mathbf{F}\Delta\mathrm{t})^3}{3!} + \cdots + \frac{(\mathbf{F}\Delta\mathrm{t})^n}{n!}\,. \tag{3.5}$$

where $\Delta\mathrm{t} = (k+1) - (k)$ is the time step. The covariance matrix of q and r are $\mathbf{Q}(k)$ and $\mathbf{R}(k)$, respectively. The standard Kalman Filter (KF) is shown in Figure 3.1.

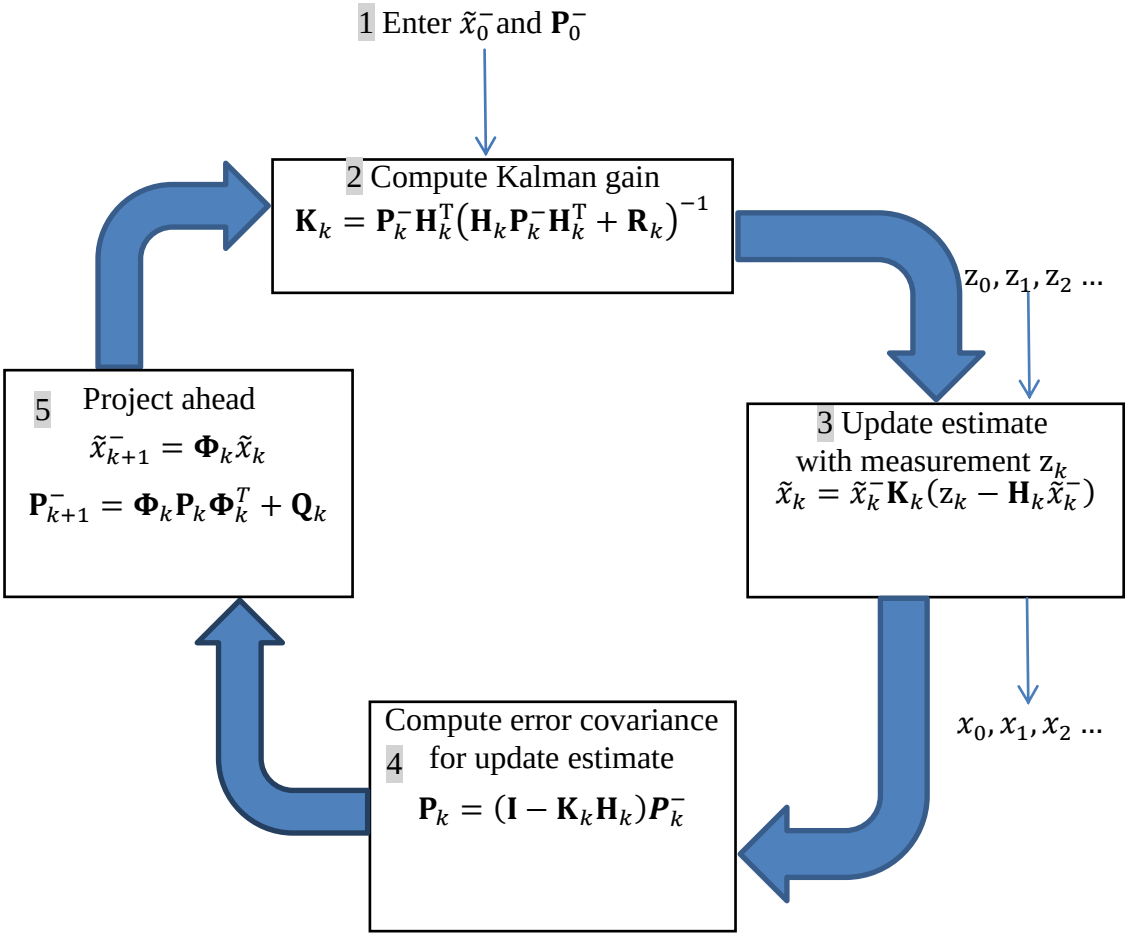


Figure 3.1. Kalman Filtering Algorithm Process

One basic and important requirement of the standard Kalman Filter algorithm is that both the process and the measurement models are linear. However, most signals in power systems are nonlinear, thus systems need to be linearized before KF can be applied. The traditional Extended Kalman Filter (EKF) [20] has been used for many years to estimate signals using linearized equations to approximate the nonlinear model, but the performance of EKF deteriorates drastically under a highly nonlinear system.

3.4.1 Unscented Kalman Filter Model-based Design

The conventional estimators incorporated in the Energy Management System (EMS) assume balanced three-phase operating conditions; consequently, a single-phase equivalent model is used for estimation with an adopted criterion to accommodate three-phase measurements. There is another method [117] which assumes the three independent measurements have the same magnitude and then incorporate them into a single-phase model. As a result the redundancy in this method is increased and the bad data at each phase can be determined. However, this method may result in perceptible distortions since the voltage and the phase angle are sensitive to estimation variables errors, especially if the imbalance is not small. An alternative method can be obtained using the positive-sequence component value from the three-phase measurements, and then add this component value to the Phasor estimator as an actual measurement.

A more recent development known as the Unscented Kalman Filter can minimize the error and overcome the weakness of traditional Kalman Filters [118]. UKF generates a finite set of points called as sigma points; then, these sigma points are transformed to a new set of points using the nonlinear model. System states and associated error covariance matrices are determined numerically based on the mean and covariance values of the transformed sigma points with minimized error. If the system under consideration is unbalanced and has different harmonic frequencies, then, UKF is a powerful technique to filter out all these harmonics [119] and only the amplitude and phase of the fundamental component are passed.

Let us consider that one of the nodes acquiring instantaneous measurements (here is voltage) for phase a, b and c respectively with time-varying harmonics is corrupted by additive white noise and can be represented as follows:

$$z_a(k) = Z_m(k)\sin(\omega k\Delta t + \varphi) + r_a(k), \tag{3.6}$$

$$z_b(k) = Z_m(k)\sin\left(\omega k\Delta t + \varphi - \frac{2\pi}{3}\right) + r_b(k), \tag{3.7}$$

$$z_c(k) = Z_m(k)\sin\left(\omega k\Delta t + \varphi + \frac{2\pi}{3}\right) + r_c(k), \tag{3.8}$$

where, $Z_m(k)$ is the amplitude and $r_a(k)$, $r_b(k)$ and $r_c(k)$ are noises that can be a white noise and n is the order of the superimposed harmonic. By applying Clarke transformation [120] to transform these signals to αβ stationary reference frame as:

$$\begin{bmatrix} z_\alpha(k) \\ z_\beta(k) \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} z_a(k) \\ z_b(k) \\ z_c(k) \end{bmatrix} + r_{\alpha\beta}(k), \tag{3.9}$$

In complex form, the αβ components can be expressed as:

$$z_{\alpha\beta}(k) = z_\alpha(k) + j\, z_\beta(k)\,, \tag{3.10}$$

where, $z_\alpha(k) = A(k)\cdot\cos(\omega k\Delta t + \varphi)$, $z_\beta(k) = A(k)\cdot\sin(\omega k\Delta t + \varphi)$ and

$A(k) = |Z_{\alpha\beta}(k)|$ is the amplitude of the signal.

And in polar form is:

$$z_{\alpha\beta}(k) = A(k)\cdot e^{j(\omega k\Delta t+\varphi)}. \tag{3.11}$$

The problem of interest is to find an estimation of the amplitude A and the phase angle φ of the harmonic n. For signals including harmonics, the model suggested in [121] is chosen. For each frequency component, there are two state variables. Thus, there are a total of $2n$ state variables. To represent the state variables of a signal with time-varying magnitude using a stationary reference, let the state variables for the harmonic n at time at time instant k expressed as follows:

$$x_1(k) = z_\alpha(k) = A(k)\cdot\cos(\omega k\Delta t + \varphi(k)) \text{ and}$$

$$x_2(k) = z_\beta(k) = A(k)\cdot\sin(\omega k\Delta t + \varphi(k))\,.$$

At time k+1, which is $(k+1)\Delta t$, these signals can be expressed in other form [122] as follows:

$$x_1(k+1) = A(k+1) \cdot \cos(\omega k\Delta t + \omega\Delta t + \varphi(k+1))$$

$$x_2(k+1) = A(k+1) \cdot \sin(\omega k\Delta t + \omega\Delta t + \varphi(k+1))$$

If we assume $A(k) = A(k+1) = A$ and $\varphi(k) = \varphi(k+1) = \varphi$, then the state variable equations can be rewritten as:

$$x_1(k+1) = x_1(k)\cos(\omega\Delta t) - x_2(k)\sin(\omega\Delta t)$$

$$x_2(k+1) = x_1(k)\sin(\omega\Delta t) + x_2(k)\cos(\omega\Delta t)$$

Thus, state variable equations can be written in matrix form as follows:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} \cos(\omega\Delta t) & -\sin(\omega\Delta t) \\ \sin(\omega\Delta t) & \cos(\omega\Delta t) \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}, \tag{3.12}$$

And the measurement equation can be written as:

$$z_{\alpha\beta}(k) = [1 \quad j] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}, \tag{3.13}$$

When the measured signal includes n frequencies; the fundamental plus $n-1$ harmonics, then the general form of the state variable equation can be represented as follows:

For n harmonics, the state variables equation can be represented in matrix form as follows:

$$\begin{bmatrix} x_1^{(1)}(k+1) \\ x_2^{(1)}(k+1) \\ \vdots \\ x_1^{(n)}(k+1) \\ x_2^{(n)}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{M}^{(1)} & 0 & 0 & 0 & 0 \\ 0 & \ddots & \vdots & \iddots & 0 \\ 0 & \cdots & \ldots & \cdots & 0 \\ 0 & \iddots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & \mathbf{M}^{(n)} \end{bmatrix} \begin{bmatrix} x_1^{(1)}(k) \\ x_2^{(1)}(k) \\ \vdots \\ x_1^{(n)}(k) \\ x_2^{(n)}(k) \end{bmatrix}, \tag{3.14}$$

where the sub-matrices $\mathbf{M}^{(n)}$ is $\begin{bmatrix} \cos(n\omega\Delta t) & -\sin(n\omega\Delta t) \\ \sin(n\omega\Delta t) & \cos(n\omega\Delta t) \end{bmatrix}$. The measurements equation is given by:

$$z_{\alpha\beta}(k) = [1 \quad j \quad \cdots \quad 1 \quad j] \begin{bmatrix} x_1^{(1)}(k) \\ x_2^{(1)}(k) \\ \vdots \\ x_1^{(n)}(k) \\ x_2^{(n)}(k) \end{bmatrix}. \tag{3.15}$$

Finally, the estimated state variables can be used to estimate the amplitude $\hat{A}^{(n)}(k)$ and the phase angle $\hat{\varphi}^{(n)}(k)$ of the harmonic n at instant k as:

$$\hat{A}^{(n)}(k) = \sqrt{x_1^{(n)}(k) + x_2^{(n)}(k)} \tag{3.16}$$

$$\hat{\varphi}^{(n)}(k) = \tan^{-1}\left(\frac{x_2^{(n)}(k)}{x_1^{(n)}(k)}\right). \tag{3.17}$$

3.4.1.1 The unscented transformations

For the model in section 3.4.1, let's assume that we have n harmonics. That means we have $2n \times 1$ state vector that is transformed by a nonlinear function $z = f(x)$. For N state variables (where $N = 2n$ in this case), we can choose $2N$ sigma points $x^{(i)}$ with mean $\bar{x}(k)$ and covariance **P** and they can be calculated as:

$$x^{(i)}(k) = \bar{x}(k) + \tilde{x}^{(i)}(k) \quad , \quad i = 1, \ldots, 2N \tag{3.18}$$

$$\tilde{x}^{(i)}(k) = (\sqrt{N\mathbf{P}})_i^T \quad , \quad i = 1, \ldots, N \tag{3.19}$$

$$\tilde{x}^{(N+i)}(k) = -(\sqrt{N\mathbf{P}})_i^T \quad , \quad i = 1, \ldots, N \tag{3.20}$$

where $\sqrt{N\mathbf{P}}$ is the matrix square root of $N\mathbf{P}$ in such way $\left(\sqrt{N\mathbf{P}}\right)^T \sqrt{N\mathbf{P}} = N\mathbf{P}$.

The sigma points can be related to these state variables as:

$$\mathbf{X}(k) = \left[\bar{x}(k), \bar{x}(k) \mp \sqrt{N\mathbf{P}_{xx}}\right] \tag{3.21}$$

The sigma points are propagated through the measurement equation as $z(k) = h(\mathbf{X}(k))$, then; the steps of UKF algorithm process are followed to get an estimation of the state variables.

3.4.1.2 UKF filtering algorithm process

According to [123], there are nine steps in the UKF processing for solving nonlinear systems which are presented below:

1. Use Equation (3.18) to propagate the state estimate and covariance from instant time $k-1$ to next instant time k by forming N sigma points $x^{(i)}(k-1)$ with appropriate changes, since the best guess of mean and covariance of $x(k)$ are $\hat{x}^{+}(k-1)$ and $\mathbf{P}^{+}(k-1)$.
2. The known nonlinear equation (i.e. $h(x)$) is used to transform the sigma points into $\hat{x}^{(i)}(k)$. With the appropriate changes, we use $f(\cdot)$ instead of $h(\cdot)$ since it's transformation is nonlinear. The transformation becomes $\hat{x}^{(i)}(k) = f(\hat{x}^{(i)}(k-1))$.
3. The $\hat{x}^{(i)}(k)$ vectors are combined for obtaining a priori state estimate at the time instant k based on Equation (3.22) .

$$\hat{x}^{-}(k) = \frac{1}{2N}\sum_{i=1}^{2N}\hat{x}^{(i)}(k) \tag{3.22}$$

4. The a priori error covariance is estimated by using this Equation (3.23)

$$\mathbf{P}^{-}(k) = \frac{1}{2N}\sum_{i=1}^{2N}\left(\hat{x}^{(i)}(k) - \hat{x}^{-}(k)\right)\left(\hat{x}^{(i)}(k) - \hat{x}^{-}(k)\right)^{T} \tag{3.23}$$

5. Use the known nonlinear measurement equation $h(.)$ for transforming the sigma points into predicted vector measurements $\hat{z}^{(i)}(k) = h(\hat{x}^{(i)}(k))$.

6. The vector measurements $\hat{\mathrm{z}}^{(i)}(k)$ are combined to obtain the predicted measurements at instant time k as in Equation (3.24).

$$\hat{\mathrm{z}}(k) = \frac{1}{2N}\sum_{i=1}^{2N}\hat{\mathrm{z}}^{(i)}(k) \tag{3.24}$$

7. Estimate the covariance of predicted measurements as in Equation (3.25)

$$\mathbf{P}_{\mathrm{z}}(k) = \frac{1}{2N}\sum_{i=1}^{2N}\left(\hat{\mathrm{z}}^{(i)}(k) - \hat{\mathrm{z}}(k)\right)\left(\hat{\mathrm{z}}^{(i)}(k) - \hat{\mathrm{z}}(k)\right)^{T} \tag{3.25}$$

8. Estimate the cross covariance between $\hat{\mathrm{x}}^{-}(k)$ and $\hat{\mathrm{z}}(k)$ as in Equation (3.26)

$$\mathbf{P}_{\mathrm{xz}}(k) = \frac{1}{2N}\sum_{i=1}^{2N}\left(\hat{\mathrm{x}}^{(i)}(k) - \hat{\mathrm{x}}^{-}(k)\right)\left(\hat{\mathrm{z}}^{(i)}(k) - \hat{\mathrm{z}}(k)\right)^{T} \tag{3.26}$$

9. Finally, the update of measurement state estimate is done by using the standard Kalman Filter as below:

$$\mathbf{K}(k) = \mathbf{P}_{\mathrm{xz}}(k)\mathbf{P}_{\mathrm{z}}^{-1}(k)$$

$$\mathrm{x}^{+}(k) = \mathrm{x}^{-}(k)\,\mathbf{K}(k)(\mathrm{z}(x) - \hat{\mathrm{z}}(k))$$

$$\mathbf{P}^{+}(k) = \mathbf{P}^{-}(k) - \mathbf{K}(k)\mathbf{P}_{\mathrm{z}}\mathbf{K}^{T}(k)$$

3.4.2 Frequency Response of UKF

It's mentioned in section 3.3 that, to increase the reporting rate when DFT is employed, the time window size should be decreased or by overlapping the windows (windows sliding).

This adaptive window width makes the DFT to suffer from aliasing, leakage and picket fence effects and hence need error compensation; consequently, the estimated amplitude and phase angle will be affected. Hence, studying the effects of window size on frequency response of a filter (estimator) is important to keep the error within allowable limits. For

studying the frequency response, the generic model shown in Figure 3.2 will be considered. This figure includes two filters $G_1(z)$ and $G_2(z)$ for eliminating undesirable harmonics and providing necessary phase shift of the two state variables in equation 3.27 (i.e. x_1 and x_2). Studies of frequency response are defined for Linear Time Invariant (LTI) systems. Most systems in reality are nonlinear. So, if systems are nonlinear, they should be linearized before UKF model can be applied.

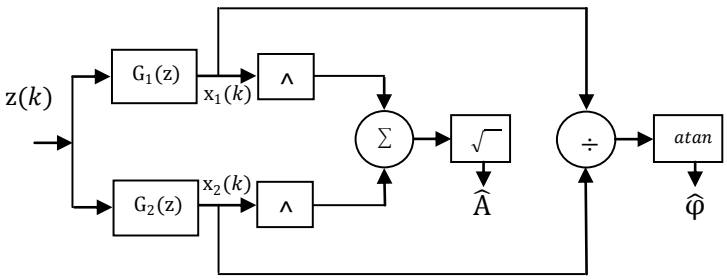


Figure 3.2. Generic Model for Phasor Estimation

In Figure 3.2, symbol "^" represents taking the square, $G_1(z)$ and $G_2(z)$ represent the transfer functions of two nonlinear filters, $\hat{A}$ and $\hat{\varphi}$ represent estimated magnitude and phase angle. Consider that the input signal is:

$$z(k)=A\cos(\omega k\Delta t+\varphi). \tag{3.28}$$

Here A and φ represent the real values of the magnitude and phase angle, respectively. Based on the model shown in Figure 3.2, the steady state outputs of $G_1(z)$ and $G_2(z)$ to x(k) can be described as:

$$x_1(k)=A\,G_1\cdot\cos(\omega k\Delta t+\varphi_1), \tag{3.29}$$

$$x_2(k)=A\,G_2\cdot\cos(\omega k\Delta t+\varphi_2), \tag{3.30}$$

where

$$G_1 = \left| G_1\left(e^{j\omega\Delta t}\right)\right|, \qquad \varphi_1 = \angle G_1\left(e^{j\omega\Delta t}\right).$$
$$G_2 = \left| G_2\left(e^{j\omega\Delta t}\right)\right|, \qquad \varphi_2 = \angle G_2\left(e^{j\omega\Delta t}\right).$$

The magnitude and phase angle estimation according to the generic model in Figure 3.2 will be as follows:

$$\widehat{A}(k) = \left({G_1}^2 A^2 \cos^2(\omega k\Delta t + \varphi_1) + {G_2}^2 A^2 \sin^2(\omega k\Delta t + \varphi_2)\right)^{\frac{1}{2}}. \tag{3.31}$$

The equation (3.31) can be rearranged to become:

$$\widehat{A}(k) = \frac{1}{\sqrt{2}}\left({G_1}^2 + {G_2}^2 + G\cos(2\omega k\Delta t + \theta)\right)^{\frac{1}{2}} \cdot A, \tag{3.32}$$

where

$$G = \sqrt{{G_1}^4 + {G_1}^4 + 2{G_1}^2{G_2}^2\cos 2(\varphi_1 - \varphi_2)} \text{ and } \theta = \tan^{-1}\frac{{G_1}^2\sin(2\varphi_1) + {G_2}^2\sin(2\varphi_2)}{{G_1}^2\cos(2\varphi_1) + {G_2}^2\cos(2\varphi_2)}$$

The sinusoidal part Equation (3.32) in controls the variation range of the estimated magnitude. If this part is set to its maximum and minimum values (+1 and –1), the maximum and the minimum values of variation in estimated amplitude can be found as follows:

$$\max(\widehat{A}(k)) = \frac{1}{\sqrt{2}}\left({G_1}^2 + {G_2}^2 + \sqrt{{G_1}^4 + {G_1}^4 + 2{G_1}^2{G_2}^2\cos 2(\varphi_1 - \varphi_2)}\right)^{\frac{1}{2}}.A \tag{3.33}$$

$$\min(\widehat{A}(k)) = \frac{1}{\sqrt{2}}\left({G_1}^2 + {G_2}^2 - \sqrt{{G_1}^4 + {G_1}^4 + 2{G_1}^2{G_2}^2\cos 2(\varphi_1 - \varphi_2)}\right)^{\frac{1}{2}}.A \tag{3.34}$$

Thus, the estimated magnitude represented by $\widehat{A}$ has a band whose upper boundary and lower boundary can be defined by Equation (3.33) and Equation (3.34), respectively. These boundaries can be also represented in normalized form as:

$$\frac{\max(\widehat{\mathrm{A}}(k))}{\mathrm{A}}=\frac{1}{\sqrt{2}}\left({G_1}^2+{G_2}^2+2{G_1}^2{G_2}^2\cos 2(\varphi_1-\varphi_2)\right)^{\frac{1}{2}}, \tag{3.35}$$

$$\frac{\min(\bar{\widehat{\mathrm{A}}}(k))}{\mathrm{A}}=\frac{1}{\sqrt{2}}\left({G_1}^2+{G_2}^2-2{G_1}^2{G_2}^2\cos 2(\varphi_1-\varphi_2)\right)^{\frac{1}{2}}. \tag{3.36}$$

Once the normalized variation band is known, the estimated magnitude presented can be assigned for any signal frequencies. If the upper boundary is very close to zero at harmonic frequencies, this indicates a high capability of the filter in rejecting harmonics. It's obvious that, the ideal filter will be the one whose upper boundary and lower boundary are both equal to one at the fundamental frequency and zero otherwise. The general model of KF for the two state variables shown in Equation (3.12) can be represented as:

$$\begin{bmatrix}x_1(k)\\x_2(k)\end{bmatrix}=\Phi\cdot\begin{bmatrix}x_1(k-1)\\x_2(k-1)\end{bmatrix}+\mathrm{K}\cdot\left(z(k)-\mathrm{H}\cdot\Phi\begin{bmatrix}x_1(k-1)\\x_2(k-1)\end{bmatrix}\right). \tag{3.37}$$

where $\mathrm{K}=[\mathrm{K}_1 \quad \mathrm{K}_2]^t$. This model represents LTI filter inherently; because the steady state gain of Kalman Filter (K) is constant vector and its elements are time-invariant and dependent on the state variables $x_1(k)$ and $x_2(k)$. For studying the frequency response, the model suggested in Equation (3.12) with the two state variables is chosen. In this model, the state transition matrix and the measurements matrix have the following form:

$$\Phi=\begin{bmatrix}\cos(\omega\Delta t) & -\sin(\omega\Delta t)\\ \sin(\omega\Delta t) & \cos(\omega\Delta t)\end{bmatrix}, \tag{3.38}$$

If $\mathrm{T}=\mathrm{N_s}\times\Delta t$, where T is the time period of the signal and $\mathrm{N_s}$ is the sample rate; then, this state transition matrix can be expressed in terms of the sample rate "$\mathrm{N_s}$" as follows:

$$\Phi=\begin{bmatrix}\cos(\frac{2\pi}{\mathrm{N_s}}) & -\sin(\frac{2\pi}{\mathrm{N_s}})\\ \sin(\frac{2\pi}{\mathrm{N_s}}) & \cos(\frac{2\pi}{\mathrm{N_s}})\end{bmatrix}, \tag{3.39}$$

$$\mathrm{H}=[1 \quad 0]. \tag{3.40}$$

By substituting Equation (3.39) in Equation (3.37) and rearranging, we get:

$$\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \end{bmatrix} + \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} z(k), \tag{3.41}$$

where $a_{11}=(1-K_1)\cos\frac{2\pi}{N_s}$, $a_{12}=-(1-K_1)\sin\frac{2\pi}{N_s}$, $a_{21}=\sin\frac{2\pi}{N_s}-K_2\cos\frac{2\pi}{N_s}$ and $a_{22}=\cos\frac{2\pi}{N_s}+K_2\sin\frac{2\pi}{N_s}$.

The transfer functions $G_1(z)$ and $G_1(z)$ are obtained after taking z transform of Equation (3.41) and manipulating the result as follows:

$$\begin{bmatrix} X_1(z) \\ X_2(z) \end{bmatrix} = \begin{bmatrix} G_1(z) \\ G_1(z) \end{bmatrix} \cdot z(z), \tag{3.42}$$

where

$$G_1(z) = \frac{K_1 - (K_1\cos\frac{2\pi}{N_s} + K_2\sin\frac{2\pi}{N_s})z^{-1}}{1 - ((2-K_1)\cos\frac{2\pi}{N_s} + K_2\sin\frac{2\pi}{N_s})z^{-1} + (2-K_1)z^{-2}} \tag{3.43}$$

$$G_2(z) = \frac{K_1 - (K_1\cos\frac{2\pi}{N_s} - K_2\sin\frac{2\pi}{N_s})z^{-1}}{1 - ((2-K_1)\cos\frac{2\pi}{N_s} + K_2\sin\frac{2\pi}{N_s})z^{-1} + (2-K_1)z^{-2}} \tag{3.44}$$

With the standardized sample rates according to IEC 61850-9-2, processes categorized in the group#2 that persist for a cycle or less and introduce very high frequency components (harmonics) can be tracked very well. According to Equation (3.43) and Equation (3.44), evolution of Kalman Filter frequency response depends on the sampling frequency (N_s) and Kalman gain (K). So, frequency response should be evaluated under the two standardized sampling rates according to the IEC61850-9-2 (80 and 256 Samples/Cycle). Also, Kalman vector gain K only depend on the ratio between the covariance Q_k and the variance R_k of noise. In the simulation of UKF, a value of 10-3 white Gaussian noise is added. At that value of noise, Kalman gain is computed as:

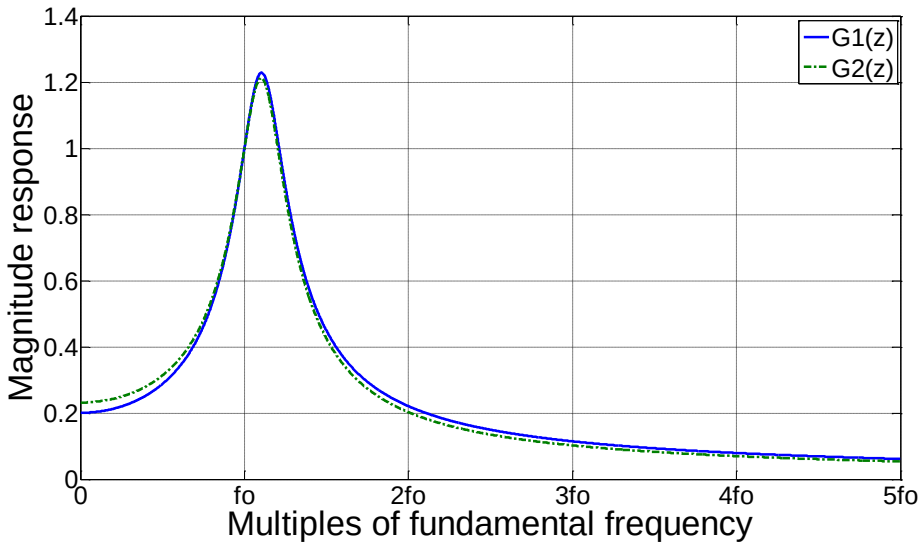
$$K = [0.0230 \quad -0.0184]^t. \tag{3.45}$$

By substituting Equation (3.45) into Equation (3.43) and Equation (3.44) and using the two standardized sampling rates according to the IEC61850-9-2 to substitute for N_s, the transfer functions $G_1(z)$ and $G_2(z)$ are obtained and the frequency response of the selected model is shown below in Figure 3.3. From this Figure, it can be seen that, at the fundamental frequency, the selected model provides an accurate estimation of the magnitude. With a sample rate of 80 Samples/Cycle, the frequency response has sharper curve. This is an indication that, it needs a longer period of transient; but with a higher attenuation on harmonics. For evaluation of the frequency response of the classical phasor estimator, the full cycle DFT has been selected which has the following transfer functions:

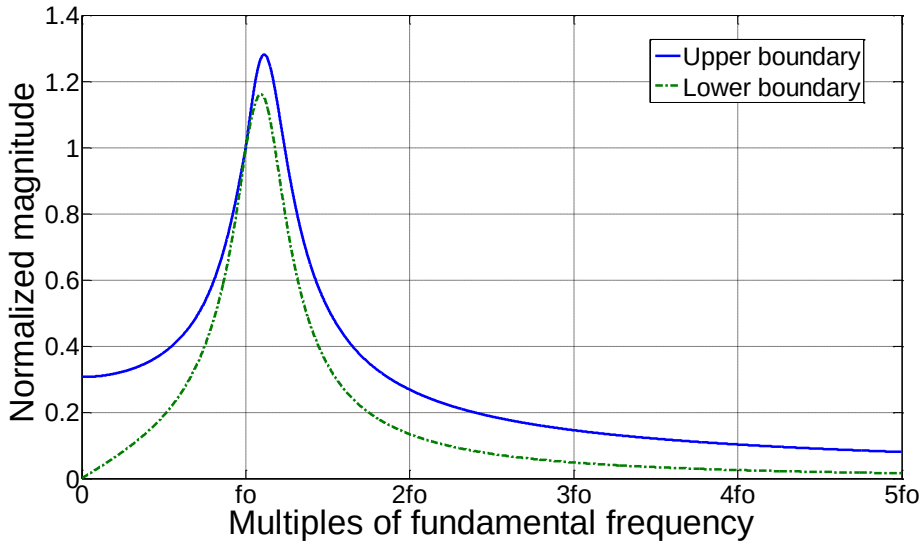
$$G_1(z) = \frac{2}{N_s} \sum_{i=0}^{N_s-1} \cos\left(\frac{2\pi(N_s-1)}{N_s}\right) z^{-i}, \tag{3.46}$$

$$G_2(z) = \frac{2}{N_s} \sum_{i=0}^{N_s-1} \sin\left(\frac{2\pi(N_s-1)}{N_s}\right) z^{-i}. \tag{3.47}$$

These two transfer functions often named as cosine and sine filters. To satisfy a reporting rate of 100 frames per second with the lowest standardized sample rate according to IEC61850-9-2 (4 kHz), the window size of those two transfer functions should be 40 samples (4000/100). If the higher standardized sample rate (12.8 kHz) is considered, then, the window size is 128 samples. Whereas the sampling frequency rate (4 kHz) is greater than twice of the highest frequency of the signal, it will be enough for studying the frequency responses. The amplitude responses and frequency response at window size of 40 samples are shown in Figure 3.4 .



a)

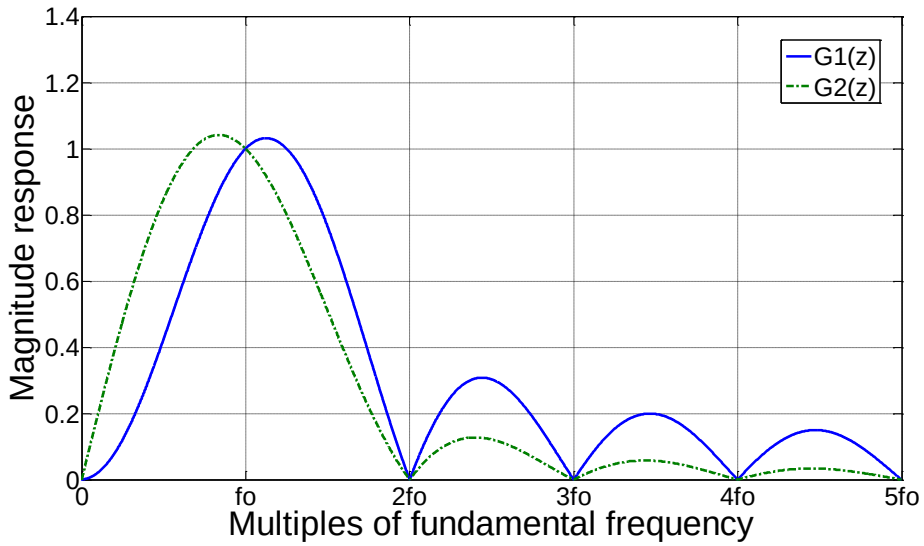


b)

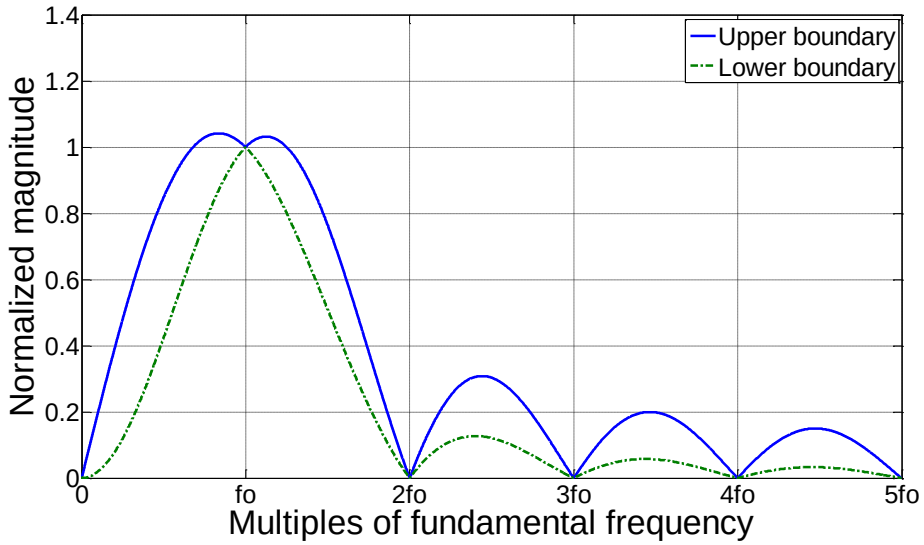
Figure 3.3. a) Magnitude Responses of UKF at N_s= 80 samples/cycles, b) Frequency Responses of UKF at N_s= 80 samples/cycles

By comparing Figure 3.3, which represents most excellent case of UKF frequency responses with Figure 3.4, capability of UKF can be decided. The sharpness of curves is

higher for UKF and there is no side lobes which indicates its capability of attenuating and rejecting harmonics.



a)



b)

Figure 3.4. a) Magnitude Responses of DFT at N_s= 40 samples, b) Frequency Responses of DFT at N_s= 40 samples

3.5 Summary

The implementation of UKF based model for phasor estimation and studying its frequency response performance are presented in this Chapter. The frequency response of the proposed estimator indicated the capability to attenuate and reject harmonics at frequencies other than the fundamental frequency. From simulation results, it can be seen that, at the fundamental frequency, the proposed model provides an accurate estimation of the magnitude. With a sample rate of 80 samples/cycle, the frequency response has a sharper curve. This is an indication that, it has a higher attenuation on harmonics.

Chapter 4

4. Performance Evaluation of Phasor Estimator within IEC 61850-9-2 Communication Network

This Chapter presents a performance evaluation of a new architecture for phasor estimation within IEC 61850 substation Process-Bus. Effects of IEC 61850-9-2 communication network and the time errors due to the Asynchrony between Merging Units on the performance of the estimated Phasor are evaluated and presented in the Chapter. The phasor estimation is also evaluated for the Total Vector Error in this Chapter. The criterion for evaluating the estimated Phasor is the Total Vector Error as define in IEEE C37.118 standard. The Total Vector Error is analyzed for the effects of the IEC 61850-9-2 samples loss, operating conditions at different Nodes Number and IEC 61850-9-2 frame size. The evaluation tests in this Chapter include test of the phasor estimator for both stationary and transient phenomena in power systems, to determine the accuracy of the measurements obtained at various conditions.

4.1 Introduction

As mentioned before, the IEC 61850 is the current worldwide accepted standard for the communication within the Substation Automation System (SAS). In general, the IEC 61850 standard consists of two kinds of communication Buses, Process-Bus and the Station-Bus which are Ethernet technology supported. The conceptual architecture of IEC 61850-based SAS and Process-Bus was shown in Figure 1.1.

One of the typical applications for the process bus communication is to transfer the information between instrument transformers, protection devices and circuit breakers. This information transfer is time critical and has a strong impact on the response time and accuracy of the protection function. The Process-Bus applications of the IEC 61850 require synchronized sampling processes for the current and voltage values acquired from the high voltage equipment (electronic instrument transformers and the conventional MUs). MUs perform sampling and should be synchronized to a global clock with the required accuracy

to guarantee that all samples are taken at the same time. These MUs convert the analogue voltage and current waveforms into the Sampled Measured Values and then cast them on the substation Local Area Network (LAN).

4.2 Proposed Architecture Based on IEC 61850 Process-Bus

The proposed architecture has two stages. Stage one (Acquisition): this stage simulates the Level-0 where the principle of the merging unit will be applied. Here, a number of nodes (MUs) that produce the raw data (SMV) from the substation switch yard are assigned with a maximum of 16 MUs and with standardized sampling rates of 4 kHz. Stage two (Estimation): this stage simulates the substation SAS (Level-2) function. Here, the UKF will be incorporated as a new function for estimating Phasor relying on the SMV is provided. The UKF algorithm is implemented for providing the Phasor computation in the central control of substation (Substation Computer). In the estimation stage all SMV published by number of MUs in switch yard can be processed at the substation computer by the dynamic estimator software. The performance of the proposed Phasor estimator is validated under architectures of different number of nodes (2, 4, 8 or 16 nodes) assigned for the measuring and one node for the control. Control node represents the substation computer and its clock is considered as the master clock in this configuration.

4.3 Tests and Performance Evaluation

To evaluate the performance of the proposed architecture, several factors have been taken into account like the number of nodes, clock offset, drift, sample losses and the number of samples per frame. Testing these factors will show how the performance of the estimated Phasor at substation control center can be affected by IEC61850 communications. IEC61850-9-2 implementation guide specifies the maximum delay value to be less than 4 msec. In the following modeling we assume the maximum number of nodes is 16 that represent the maximum number of MUs. Also, the number of samples is set to 8 samples

per frame, which present the maximum payload allowed according to IEC61850-9-2 frame structure. Furthermore, 10 simulations were conducted to test the effect of number of nodes with fixed SMV per frame as stated earlier.

The value of the maximum delay is depending on a number of factors, (i) the traffic load or more precisely how many nodes are effective and generating SMV, and (ii) Band-Width (BW) occupied. The size of the SMV frame when it carrys 8 samples (Maximum Samples/frame according to the standardized frame structure in IEC61850-9-2) is about 1000 bytes which means one MU will consume at 4 kHz sampling frequency approximately 30 Mbit/sec. To see only the effect of nodes number on delay, 100% of BW has been assigned. The effect of increasing the number of nodes on network delay is shown in Figure 4.1. Delay value is varied with the number of nodes and increased from 1.4 msec to a maximum value of 3.2 msec. Here, the frame size is assigned to be fixed at 8 SMV. The second factor in this context which has effect on the delay is the number of SMV samples per frame (frame size). To examine the impact of this factor, the simulation is repeated 10 times again and in each time the standard deviation (STDEV) of delay was recorded. Figure 4.2 shows how the delay of network is affected by increasing the minimum and maximum numbers of SMV per frame.

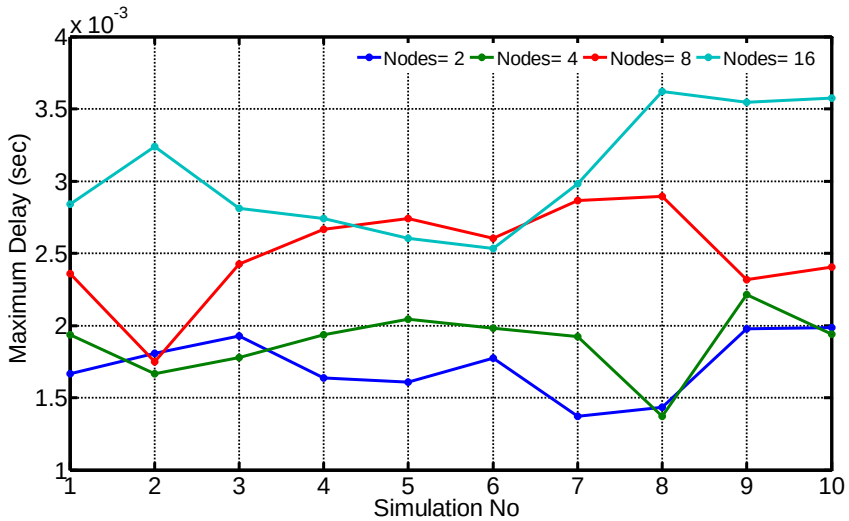


Figure 4.1. Variation of Maximum Delay with Nodes Number

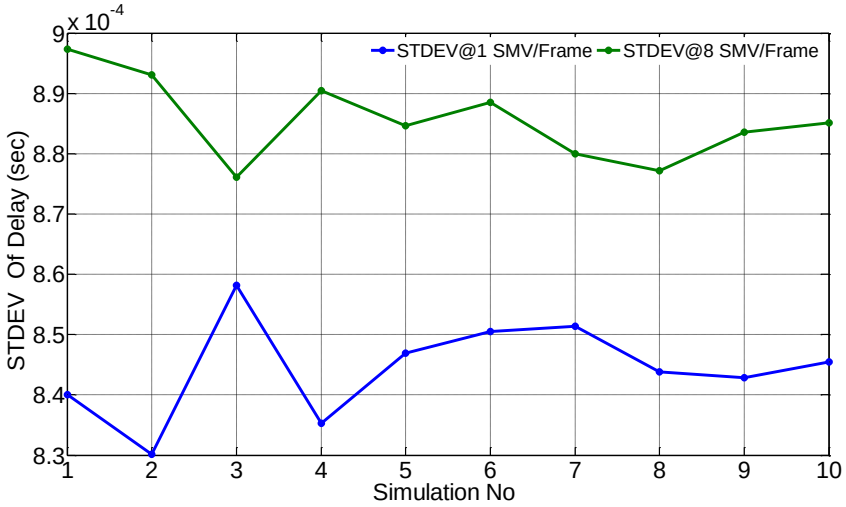


Figure 4.2. Standard Deviation of Delay at Different Frame Sizes

In order to see the effects of sample loss and time errors on the estimated Phasor by UKF, a test signal with a positive-sequence of 1.0 pu, 25 degrees was produced. To simulate the voltage , a positive sequence harmonics of 2nd and 5th degree with 1.0 pu, 20 degrees are superimposed at t=0 sec. A step change of 0.5 pu is applied at t=0.1 sec on positive-sequence voltage magnitude, then, at t = 0.3 sec, 1 pu decrease was applied again. The entire simulation time is set to a period of 0.5 sec and the sampling frequency was set to SMV#1 (4 kHz). Many cases are assumed and simulated depending on losing the samples and/or the time error effects. In each case, the percentage error results are computed and provided.

***Case 1**: No sample loss or time errors*

This case represents the ideal case in which there are no time errors or sample losses. To simulate this case, the time clock of the considered node was set to the true time of the main control node (Master Clock). Also, to avoid the sample loss, the BW of this node was assigned 100%. The estimated Phasor is shown in Figure 4.3. It's clear the estimator is

able to track the step change in magnitude and the percentage errors are always less than 1% (Figure 4.4).

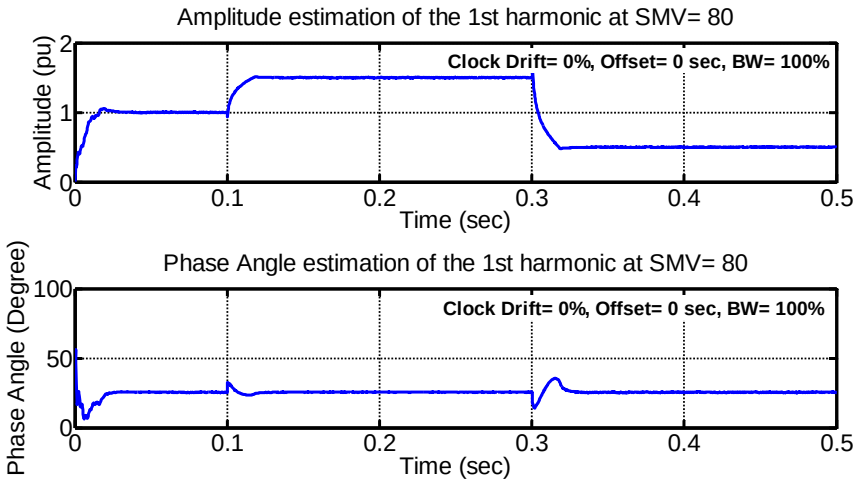


Figure 4.3. Change in Phasor Estimation at Ideal Case

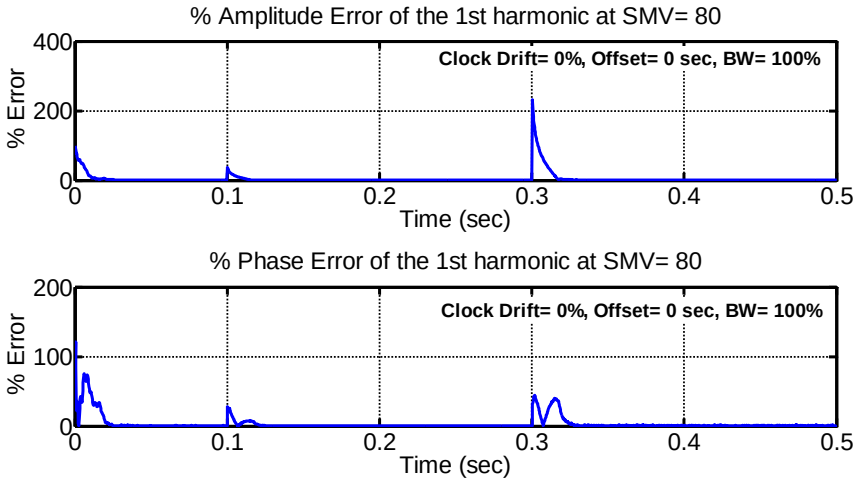


Figure 4.4. % Errors of Estimated Phasor at Ideal Case

***Case 2**: In the presence of time errors*

Functionality of IEC61850 depends on time-stamping the samples then publish them to the network. All nodes that perform sampling and time-stamping should be synchronized to one central clock (Master). In the proposed configuration, the clock of the central control node where the estimation function is performed has been chosen as the master clock. To show how the error of the clocks can affects the performance of Phasor calculation, a value of 0.01 clock drift is assigned to emulate the value of a time error in the considered node. This case expresses a situation where the clock of a sensing node is not perfectly synchronized to the master clock (substation computer node). A node with time drift equals 0.01 means that the time of node clock runs 1% faster than the nominal time assumed. The performances of the estimated Phasor are shown in Figure 4.5. The Figure shows how the drift can affect the % error in Phasor estimation in the node under consideration. Comparing the upper with the lower graphs one can see how large is the change in the phase angles of the fundamental harmonic. Also, Figure 4.6 and Figure Figure 4.8 show the % errors in fundamental harmonic estimation and 2^{nd} harmonis, respectively. The error is increasing rapidly with time and can reach a value of 100% at 0.1 second (5 cycles). It's clear that clock error has a deep impact on the stability and performance of the phasors estimated generated by UKF.

***Case 3**: In the presence of sample losses*

According to IEC61850 standard, the Process-Bus should realize a BW of 10 Gbit/sec. In our proposed architecture, the maximum number of nodes (MUs) is assumed 16 and with consumption of 30 Mbit/sec per MU. This means a total of 480 Mbit/sec for all nodes. Here, the aim is to show how the performance is drastically degraded due to the loss of some of the samples and assigned BW limit. In reality, IEC61850 communication should handle the traffic burden of the substation without any limitation. If such limitation exist, the traffic limitation requirement can be decreased by filtering the SMV and only those where the IEDs are connected are allowed to traffic. Figure 4.8 and Figure 4.9 show how

the performance of the estimated Phasor is getting worst after decreasing the BW from 100% (480 Mbit/sec) to 99% and 90%, respectively. The effect of deceasing BW appears in terms of sample a loss which in turn appears as spikes in both amplitude and phase angle errors.

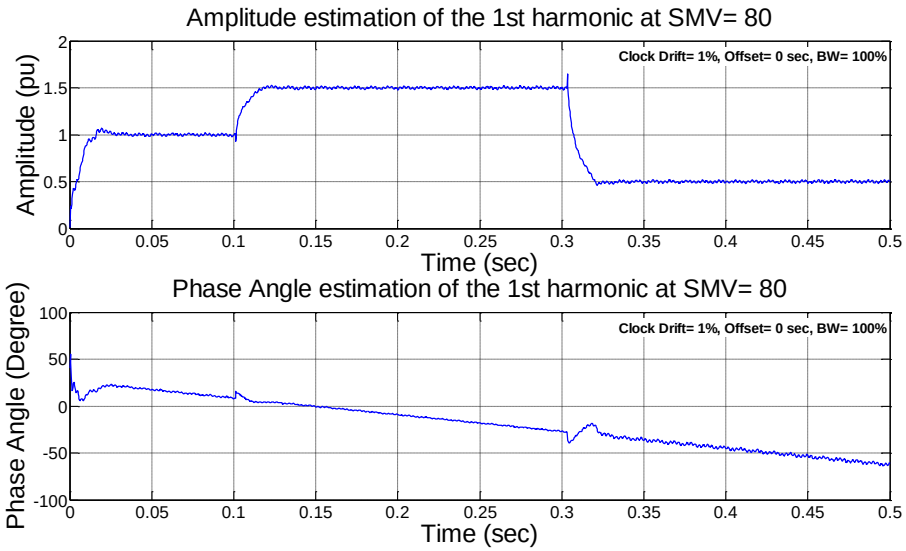


Figure 4.5. Effect of 1% Drift on Phasor Performance

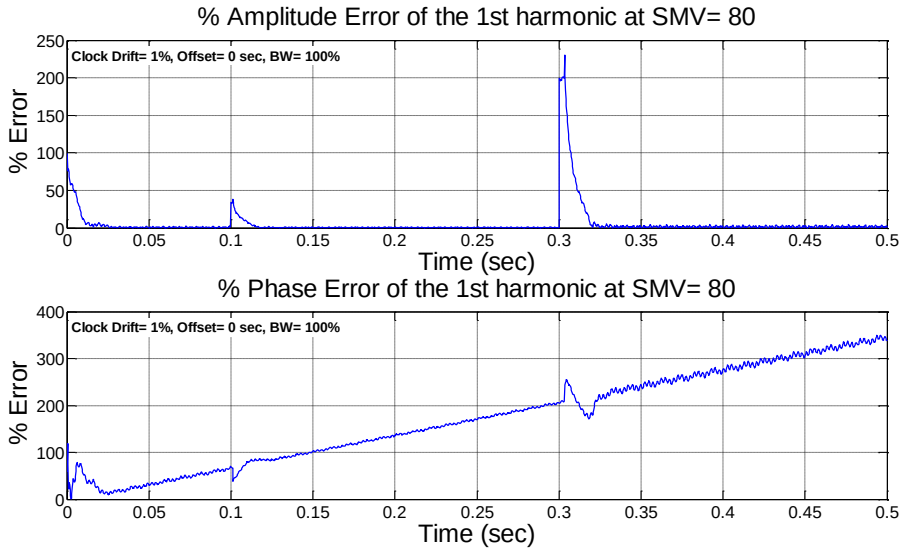


Figure 4.6. % Error of Estimated Phasor at 1 % Drift Error

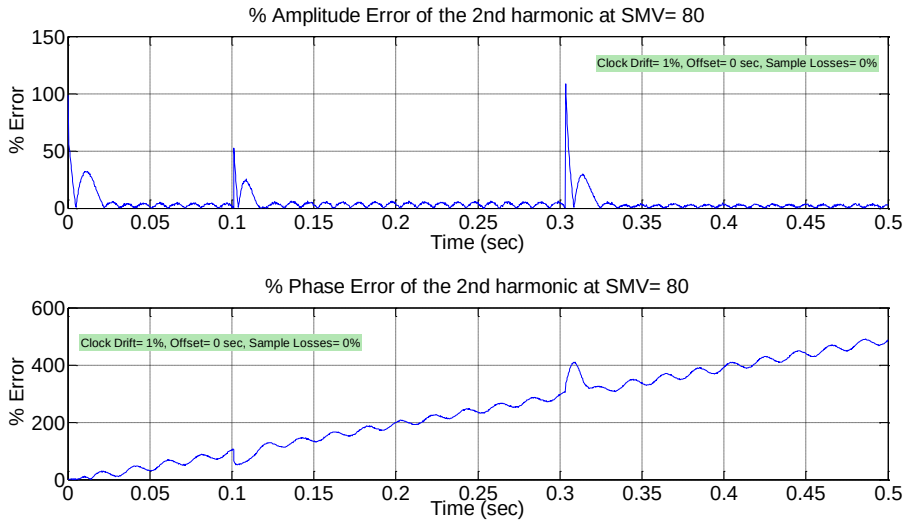


Figure 4.7. Effect of 1% reduction in BW (i.e. 99% BW) on Estimated Phasor (2nd harmonic)

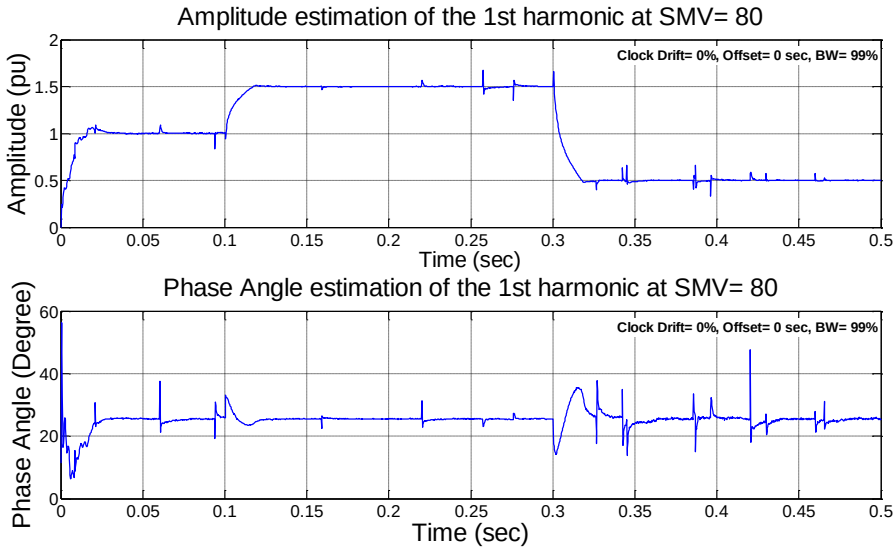


Figure 4.8. Effect of 1% reduction in BW (i.e. 99% BW) on Estimated Phasor

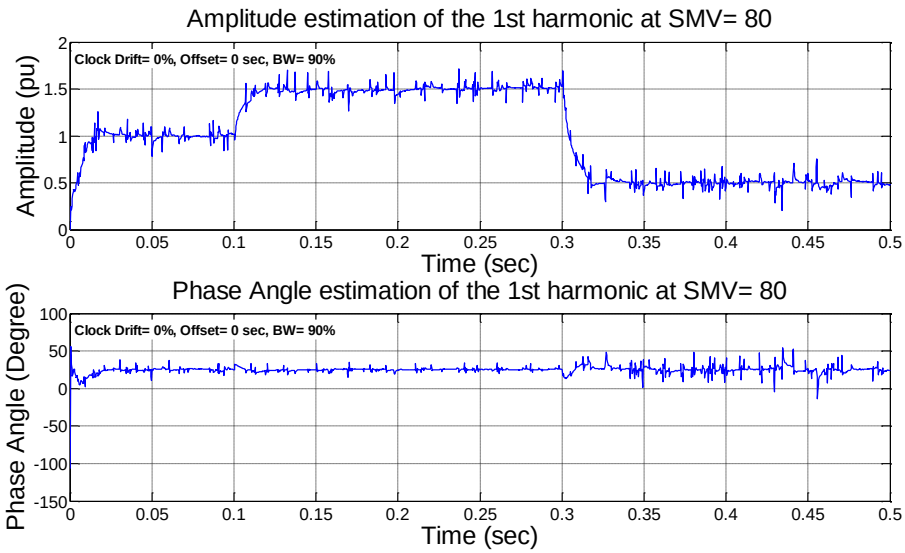


Figure 4.9. Effect of 10% reduction in BW (i.e. 90% BW) on Estimated Phasor

4.3.1 Impacts of IEC 61850-9-2 Samples Loss on Total Vector Error of an Estimated Phasor

The functionality of UKF depends on the updating of his states with SMV received from MUs. If there is no samples loss, the functionality of UKF will be outstanding and hence the TVE error will be assumed less than the maximum value allowable in IEEE C37.118 ($\leq$ 1%). Samples loss is depended on a number of factors, (i) the traffic load or more precisely how many nodes are effective and generating SMV, and (ii) Band-Width (BW) occupied.

The size of the SMV Frame when it carries eight samples (Maximum Samples/Frame according to the standardized Frame structure in IEC61850-9-2) is about 1000 bytes which means one MU will consume at 4 kHz sampling frequency approximately 30 Mbit/sec. To evaluate the effect of nodes number on the TVE, the number of nodes has been increased gradually until the TVE reach its allowable value. Also, the number of samples is varied from 1 to 8 samples (ASDUs) per frame, which present the maximum payload allowed according to IEC61850-9-2 Frame structure.

4.3.1.1 TVE Definition and SMV frame structure

Problems of classical Phasor estimators can be avoided by using Kalman Filter which has been recognized as one of the most powerful algorithms for Phasors estimation at burst changes with less error [17-23]. It's a mathematical technique, which is extensively used to compute the optimal estimates of a dynamic system states. The estimates are optimal in the sense that estimation errors are minimized in the least-squared sense. Updating states of KF may be done by using the IEC 61850-9-2 Sampled Measured Values (SMVs) streamed in the substation Process- Bus [39, 106]. The streamed values that have sampling rates of 4 kHz or 12.8 kHz at the 50 Hz nominal frequency of power systems can be quite suitable for application of tracking the robust changes in input signals.

As the IEEE C37.118 standard published in 2005 and its updated version in 2012, one of the innovations in both versions is that the measurement must be accurate to within one percent TVE. The standard defines the TVE as follows:

$$\mathrm{TVE} = \frac{\left|\vec{\mathrm{U}}_{\mathrm{Measured}} - \vec{\mathrm{U}}_{\mathrm{Ideal}}\right|}{\left|\vec{\mathrm{U}}_{\mathrm{Ideal}}\right|} \tag{4.1}$$

where $\vec{\mathrm{U}}_{\mathrm{Measured}}$ is the measured Phasor and $\vec{\mathrm{U}}_{\mathrm{Ideal}}$ is the ideal Phasor. The measured Phasor can be a voltage Phasor or a current Phasor. The Phasor error can be written as follows:

$$\vec{\mathrm{U}}_{\mathrm{Error}} = \vec{\mathrm{U}}_{\mathrm{Measured}} - \vec{\mathrm{U}}_{\mathrm{Ideal}} \tag{4.2}$$

As was shown before in Figure 2.8, the APDU is consisting of one or more (Up to eight) ASDUs. The ASDU is the data unit in which the measured values and other related information can be saved. The measured values of voltage or current signals are saved in the field "DataSet". Each "DataSet" field can carry eight measured values which represent three-phase voltage and current measured signals with neutral. The total size of the SMV is about 1000 bytes when one ASDU is assigned. Every addition of one ASDU can add 124 Bytes to the total size of the SMV Frame.

4.3.1.2 UKF and Preliminary Test for Covariance Matrix Effect

The noise covariance matrix (R) of the measurement noise r in Equation (3.2) is playing a big rule in the accuracy of the Kalman Filter and the practical implementation of the Kalman Filter is depending on getting a good estimate of it. To start testing R, a testing a signal should be generated first to simulate the real ones. The simulation has been done for signal that has a voltage unbalance, a positive sequence harmonics of 2nd and 5th degree with 1.0 pu, 20 degrees are superimposed at t=0 sec. The entire simulation time is set to a period of 0.2 sec and the sampling frequency was set to 4 kHz. Monte Carlo simulations are conducted for computing the R and the number of ensemble runs was 100.

The effect of change in R on the TVE is seen in Figure 4.10. The minimum value of R which make the TVE exceeds its maximum allowable error is 0.004. With R=0.004, the maximum TVE is calculated and shown in Figure 4.11. TVE is not considered in the initializing period (t=0 to 0.05 sec) where it can reach meaningless values of hundred of percents. After the period that the Kalman Filter needs to be initialized, the TVE is always less than one percent.

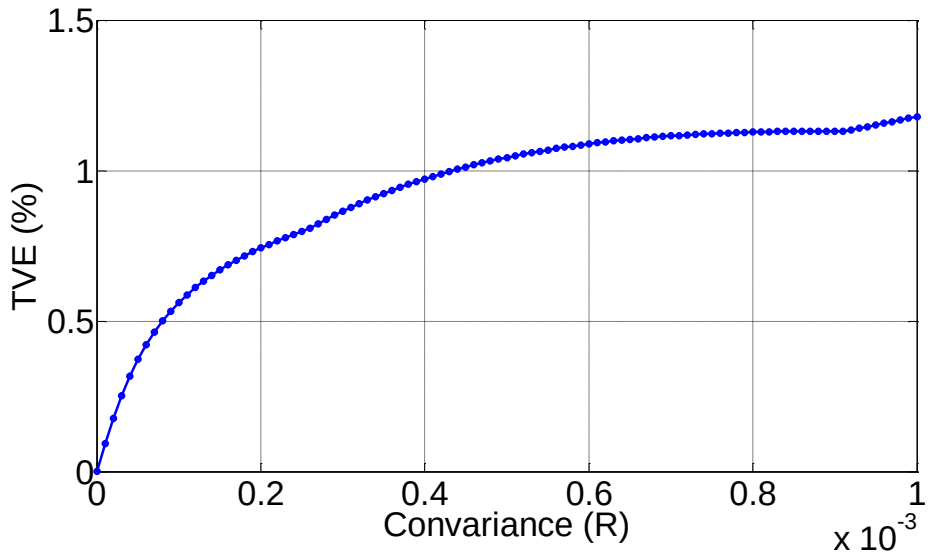


Figure 4.10. TVE vs. Covariance R

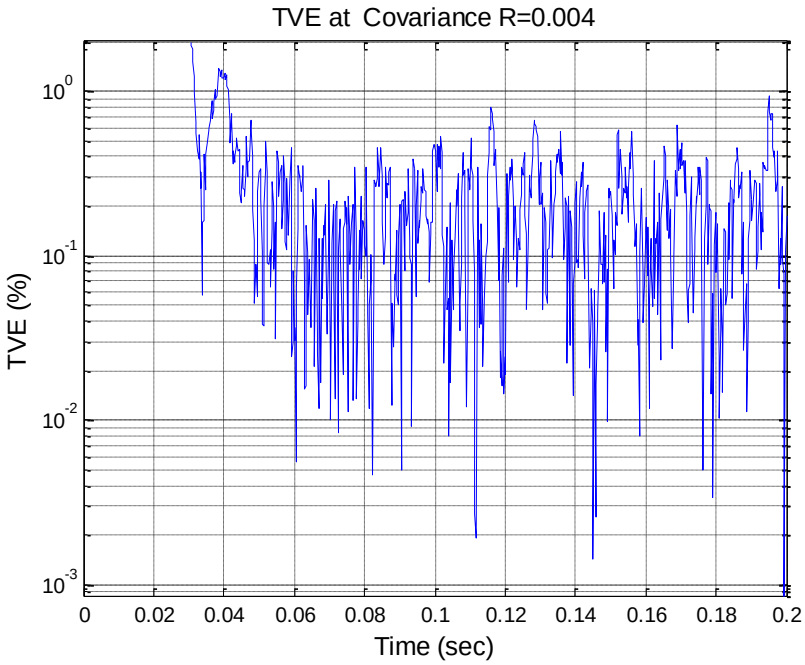
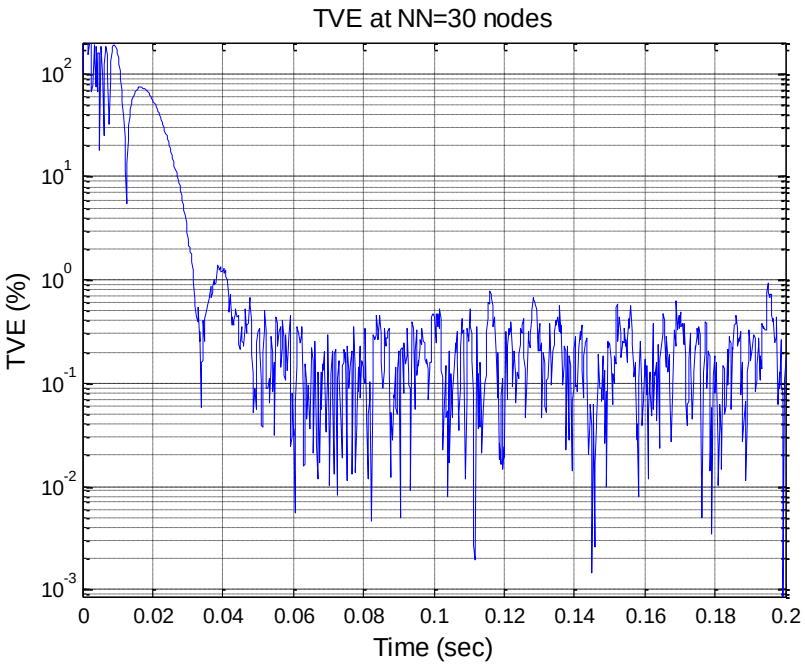


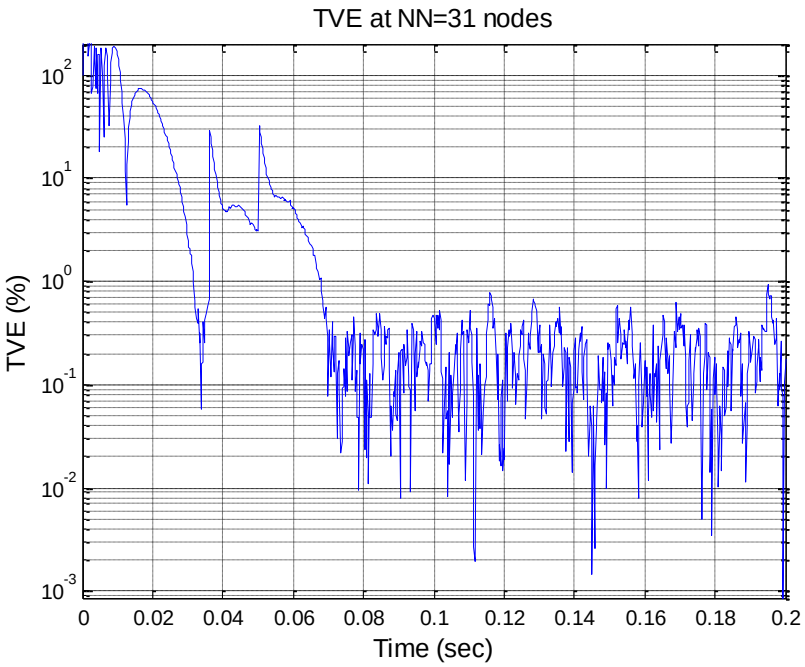
Figure 4.11. Maximum TVE at R=0.004

4.3.1.3 Test for Nodes Number Effect

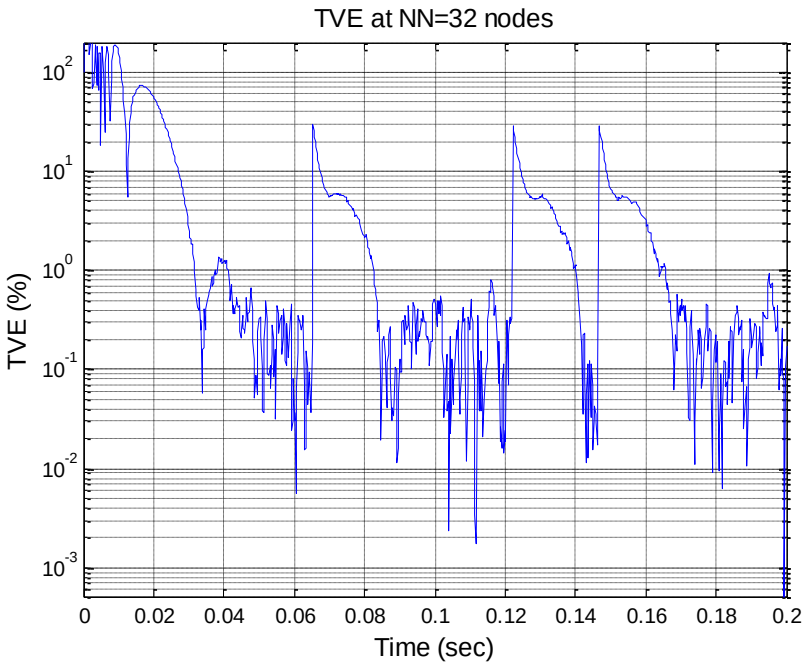
To evaluate the effect of Nodes Number on the maximum TVE, the size of Frame has been set to its maximum size and the number of Nodes has been changed from one to a number of Nodes where the maximum value of TVE is becoming higher than the allowable value of TVE. It has been found that, this number is thirty nodes. This number is considered the maximum load that IEC61850 Process-Bus can handle. When the Nodes Number is larger than thirty, the TVE increases suddenly to values higher than the allowable error and the increase can reach a value of about 30 of percent due to the samples loss at different moments of operation. The effect of Nodes Number on TVE is shown in Figure 4.12.


TVE at NN=30 nodes
TVE (%)
Time (sec)
10^2
10^1
10^0
10^-1
10^-2
10^-3
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2

a)


TVE at NN=31 nodes
TVE (%)
Time (sec)
10^2
10^1
10^0
10^-1
10^-2
10^-3
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2

b)

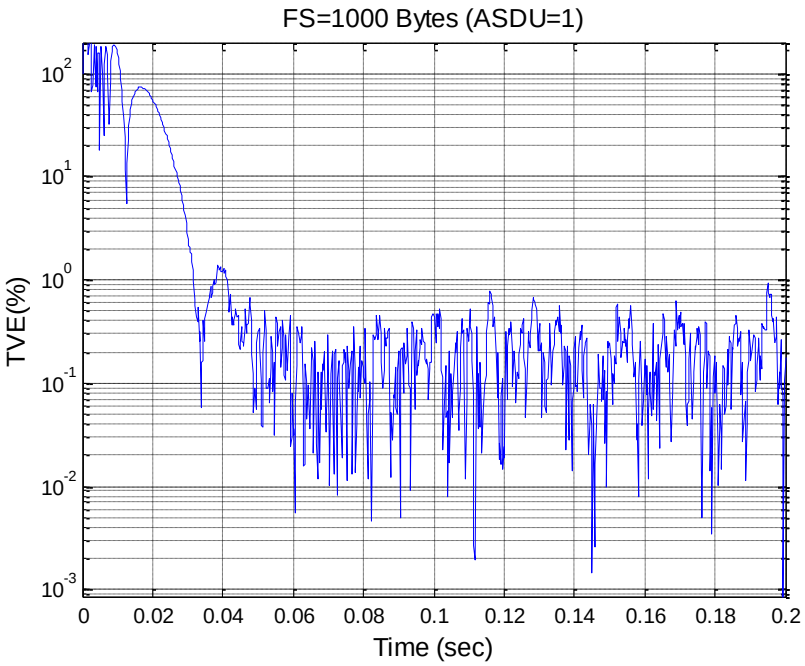


c)

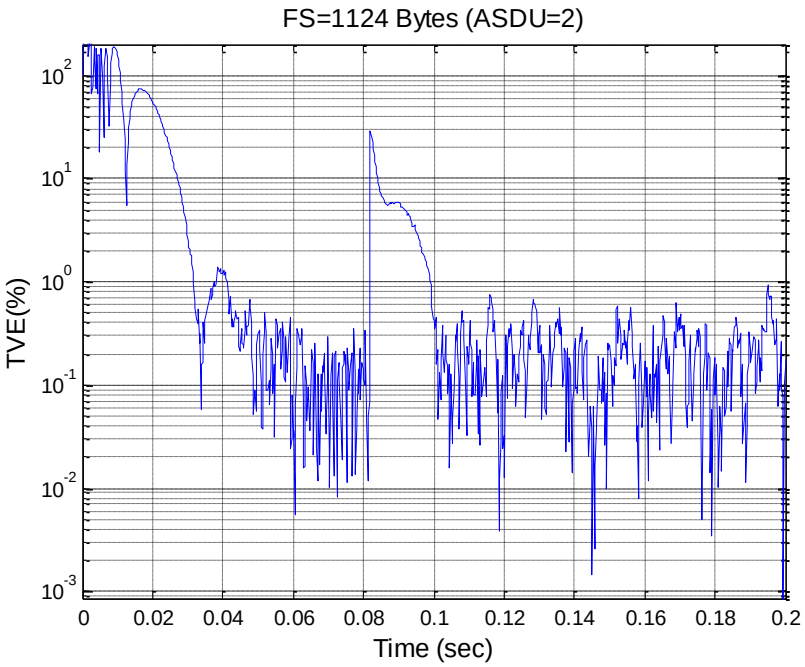
Figure 4.12. TVE at Different Nodes Number (NN=30, 31 and 32)

4.3.1.4 Test for Frame Size Effect

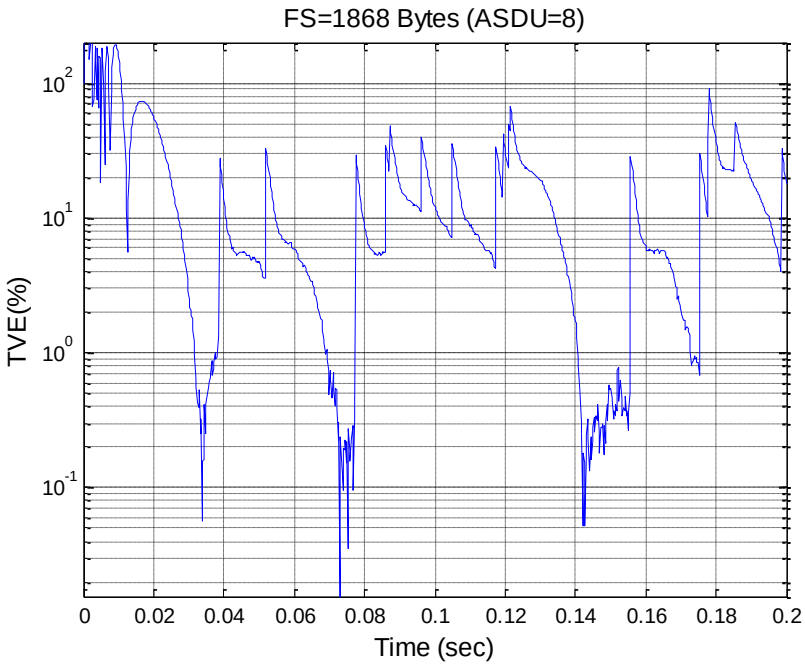
According to IEC61850 standard, the Process-Bus should realize a Band-Width (BW) of 10 Gbit/sec. To evaluate the effect of SMV Frame Size on TVE, the number of ASDUs is varied from one to eight ASDUs (Samples) per frame, which present the maximum payload allowed according to IEC61850-9-2 Frame structure. Increasing SMV Frame Size make it occupied more BW and as a result traffic increase and hence more samples loss occurs. Also, the Nodes number is set to thirty Nodes which is the worst case of Nodes Number that can be handled. The effect of increasing the number of ASDU per frame makes the number of sample lost increases which in turn appears as shoots in TVE as shown in Figure 4.13.


FS=1000 Bytes (ASDU=1)
TVE(%)
Time (sec)
10^2
10^1
10^0
10^-1
10^-2
10^-3
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2

a)


FS=1124 Bytes (ASDU=2)
TVE(%)
Time (sec)
10^2
10^1
10^0
10^-1
10^-2
10^-3
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2

b)



c)

Figure 4.13. TVE at Different Frame Size (ASDU= 1, 2, 3 and 8)

4.3.1.5 Effect of BW on TVE at different LAN Networks

The LAN transmission can produce a significant error which can affect the performance of protective system. In most circumstances, this error can be minimized by choosing the appropriate LAN. The Media Access Controller (MAC) is crucial for minimizing the impact of errors and should be selected carefully [124, 125]. The transmission of packets in LAN can be accomplished today in one of three different ways [126, 127]:

1. Using Carrier Sense Multiple Access with Collision Detection CSMA/CD (e.g. Ethernet).
2. Carrier Sense Multiple Access with Arbitration on Message Priority CSMA/ AMP (e.g. CAN).

3. Using a Switched Ethernet, where each sender node in the network connects to the network central switch using a full-duplex connection.

CSMA/CD technique depends on avoiding any traffic collision when the network is busy. Any node that is sending should wait until the network becomes free. When any collision is detected, the sender node should back-off for a period of time. At the end of waiting period, the node attempts to re-transmit its last packet. In CSMA/ AMP technique, if the network is busy, the sender node should wait until the network becomes available. In case of any collision detection, the message with a highest priority is transmitted first. When two or more messages having the same priority and are requesting the transmission in the same time, then, the priority can be decided in an arbitrary manner by the network. In Switched Ethernet technique, each sender is connected to the network central switch by using a full-duplex connection so that no collisions occur on the network segments compared to an ordinary Ethernet. The network central switch can stores any received message in its buffer and then can forward it later to the right receiving node.

The BW assigned at different LAN networks can affect the accuracy of estimated Phasor (i.e. TVE). This BW depends on a number of operational factors like the Number of Nodes and the size of the Packet (Frame). According to the IEC 61850 SMV structure, the frame size occupies 1000 Bytes when there is one ASDU. Only one transmission Node with one ASDU has been used in the evaluation. The effects of BW on TVE at different LAN networks can be seen in Figure 4.14.

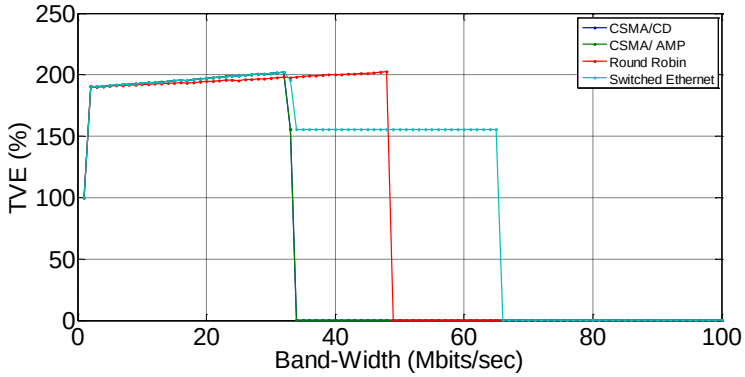


Figure 4.14. TVE Performance at Different LAN Networks

It's noted from the figure that to satisfy TVE value less than the maximum value allowable in IEEE C37.118 (≤ 1%), each LAN network needs different BW and the Switched Ethernet needs a higher BW (65 Mbits/sec) . It's also noted that, both CSMA/CD and CSMA/AMP networks have the lesser and the same value of BW (35 Mbits/sec) to satisfy the allowable TVE value.

4.3.2 Effect of BW on Latency at Different LAN Networks

It's assumed in real-time networks (i.e. LANs) that the Latency is very small and can be neglected. Actually, the value of the latency is varying from one network to another and this value should be considered (even, if it's value is very small) in many applications. According to the IEC 61850 implementation guideline, the maximum recorded Latency should be less than 3 msec. This value can be satisfied with any real-time network, but at different assigned BW. Figure 4.15 shows the BW needed for each network. It's clear that the Switched Ethernet needs less BW then others to satisfy the allowable value of Latency. Also, BW is changing linearly until it satisfies allowable value of Latency in the case of Round Robin network.

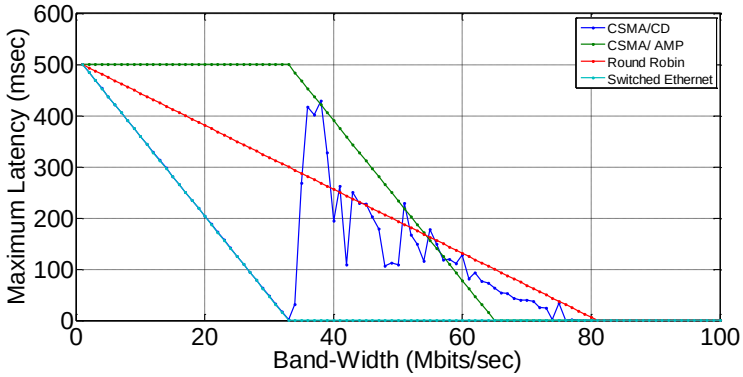


Figure 4.15. Maximum Latency Introduced at Different LAN Networks

4.4 Phasor Estimation in Real-Time

4.4.1 Real-Time Environment and Testing Procedure

The testing environment that has been developed is depicted in Figure 4.16. The environment consisting of two models representing client (MU) and server (PMU) and it works in real-time. The main functionality of the client is to produce and write (cast) SMV at standardized sample rate of 4 kHz according to IEC 61850 standard, while the main functionality of the server is to produce phasors at reporting rate of 100/sec according to IEEE C37.118 standard. Both client and server models are "light weight" soft programmed in C language and run in two Linux machines. This soft testing environment is flexible and not requiring expensive hardware. The implantations of client and server models are explained in details in the next subsection.

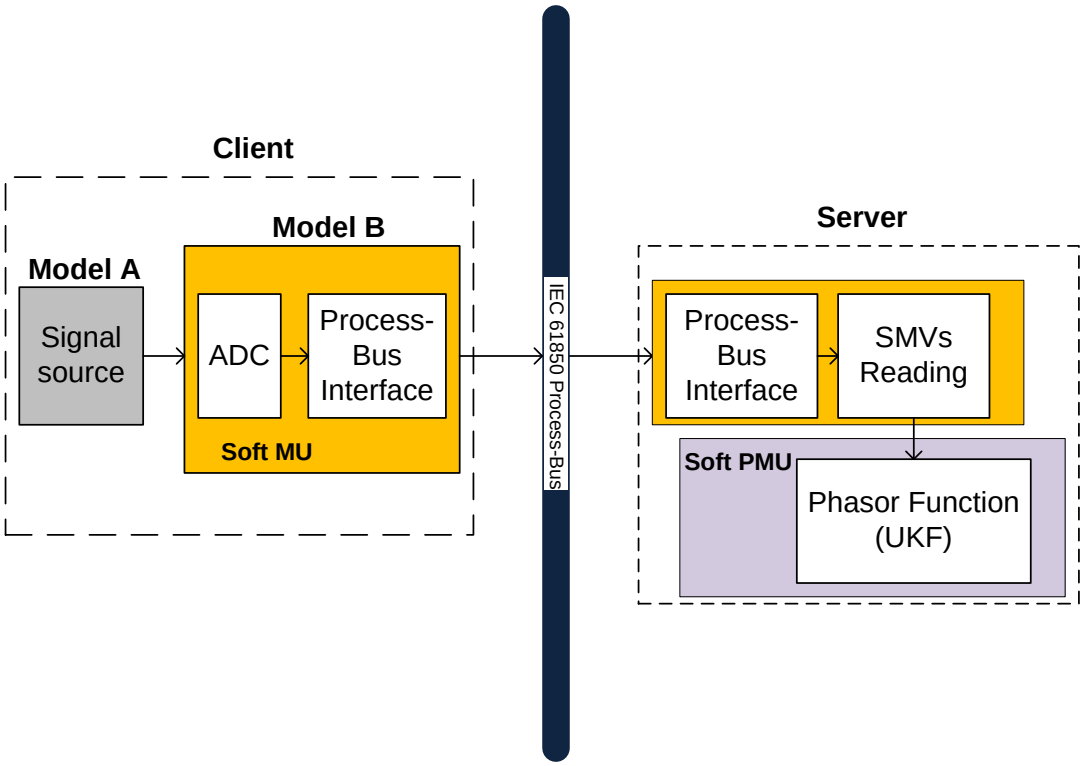


Figure 4.16. Structure of Testing Environment

4.4.2 Client and Server Models Implementation

The client model is divided into sub-models A and B. Model A represents the test signals generation. These signals are generated using a signal source in Simulink to facilitate synthesizing the required test signals such as step changes or harmonics, to the soft MU in real-time. On the other hand, Model B represents the soft MU, which is modelled in software using a C code function. The "S-function" in Simulink is used to run the C code write function of SMVs as a function block in Simulink simulation environment which is called by the "S-function builder" calls 4000 times per second. The "S-function builder" represents a MU in this simulation environment and used to digitalize the signals with 80 samples/cycle sample rate, then, encapsulated them as SMV frame by the Process-Bus interface and cast them into the Process-Bus. These casted SMV can then be read by the

server. The server contains two C code functions: SMVs read function and phasor estimation function. The two functions integrated as one C code function and the implementation of the server follows the same way that the client was implemented before in Simulink. To work in real-time, the whole Simulink models should be converted to a Matlab's xPC target; so it can run in real-time on a normal PC.

As mentioned earlier, the main functionality of server is to produce phasors. These phasors should be within the allowable error according to IEEE C37.118 standard. When using UKF, it should run fast enough to work as a robust estimation tool for tracking phasors. This requires that the computational time needed to extract the phasors should be less than the sampling period. To do so, UKF should update its estimation with execution time interval equals to the sampling time interval (1/4000 second) or less in case of using SMV#1 configuration. Also, the performance quality of UKF depends on no loss of received samples and that is its ability to update its state within sampling period before the next samples. The soft PMU has been configured to produce phasors at reporting rate of 100 Hz. To reduce the sampling rate produced by KF (4kHz) to a sampling rate of 100Hz so the phasors can be produced at the standardized reporting rate of 100 Hz, a decimation function by fifth-order Finite Impulse Response (FIR) filter [128] has been implemented with a sampling factor equals to 40.

4.4.3 Testing Signals Development

There are many kinds of events that can occur in a power system under normal operating conditions. The most common kinds of these events are faults which can be represented as electro-magnetic transients, and the lines tripping that induced oscillations which can be represented as electro- mechanic transients. If any event occurs, the measurements should be consistent or not interrupted heavily by this event. The performance during step changes shall be determined be applying balanced 3-phase step change to balanced 3-phase input signal. These developed input signals are steady state signals until the occurring of an

event. At the moment of event occur, step changes or harmonics are superimposed to the input signals.

In order to start, a positive-sequence of 1.0 pu amplitude and 30° phase angle is specified. Although, IEEE C37.118 standard specifies the step changes value to be $\pm 10\%$ and $\pm 10°$ for amplitude and phase angle respectively, more step changes of ± 20 and $\pm 50\%$ in amplitude and $\pm 60°$ and $\pm 90°$ in phase angle have been chosen arbitrary to represent severe cases of the power systems. The presentation of higher step changes can help in evaluating the proposed estimator in worse cases. The step changes have been applied at t=0.25 sec. To represent voltage unbalance, a positive sequence harmonics of 2nd and 5th degree with 1.1 pu, 30 degrees are superimposed also at t=0.25. The entire period time of test signal is set to a period of one second and the sampling frequency of the soft MU is configured to SMV#1 (4 kHz) as previously mentioned . The frequency is set to the 50 Hz power systems nominal fundamental frequency.

4.4.4 Test Results of Measurement Reporting Latency

According to IEEE C37.118 standard, measurement reporting latency can be defined as the maximum time difference from which the message is sent to the time that it is received over at least 1000 consecutive messages and the maximum allowable value should be below $1/F_s$. Many factors are included in determining the latency, for example time to make measurements, communication link, and estimation method. Unlike the PMU which estimate phasors at process level, the proposed technique estimates phasors at bay level by using the time- critical SMV received from the process level.

The main advantage of using SMV for updating the proposed estimator is the high speed of communication due to using date link layer traffic. Increasing the number of MUs that are producing SMV has an impact on the latency of the SMV and hence the latency of estimated phasors at bay level. Here, the latency includes the time that the soft merging unit needs to process SMV, the time the SMV travel via communication link and the time

that KF function needs to estimate phasor. The aim of the proposed estimator is to satisfy reporting rate equals to 100 frames per second.

To see how an increase in the number of MUs can affect the latency of phasors estimate, test setups have been done for 1, 5 and 10 merging units and a number of 4000 consecutive messages. It's clear from Figure 4.17 that the latency is increase with the increase in number of MUs and the standard deviation is almost double with every increase of 5 MUs, which means that the traffic can handle about 10 MUs with latency value under the allowable maximum value by the standard.

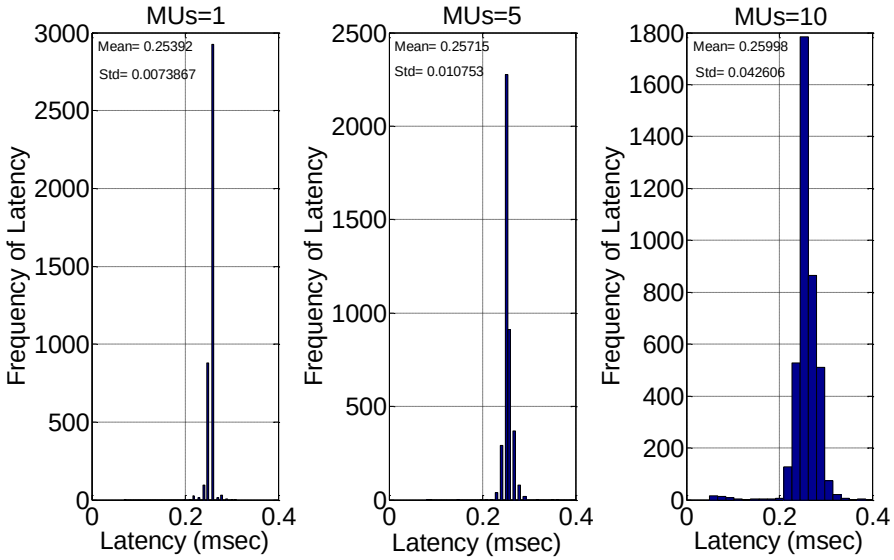


Figure 4.17. Effect of Number of Merging Units on Latency of Phasor

4.4.5 Test Results of Step Changes in Amplitude and Phase Angle

Step changes of signal can cause oscillations in the phasors measured by estimators. Assessing the performance of estimators by calculating TVE during step changes is not useful. The difference of the calculated TVE before and after the step change can reach hundreds of percents. A more simpler measure of estimator's performance during step change is suggested in [129]. It's suggested the performance as elapsed number of

reporting (Frames) between the moment of fault occurrence and when the TVE value backs again to its limit value (1%). Testing results of the suggested estimator are listed in tables Table 4.1 and

Table 4.2. The highest settling time recorded at angle deviation of $\pm 90°$ which takes eleven reporting frames or 110 msec. These tests were performed at reporting rate of 100 frames per second.

Table 4.1. Step Response of Amplitude

Magnitude step, %	10	20	50	-10	-20	-50
Elapsed Number of Frames	9	9	9	9	9	10

Table 4.2. Step Response of Amplitude

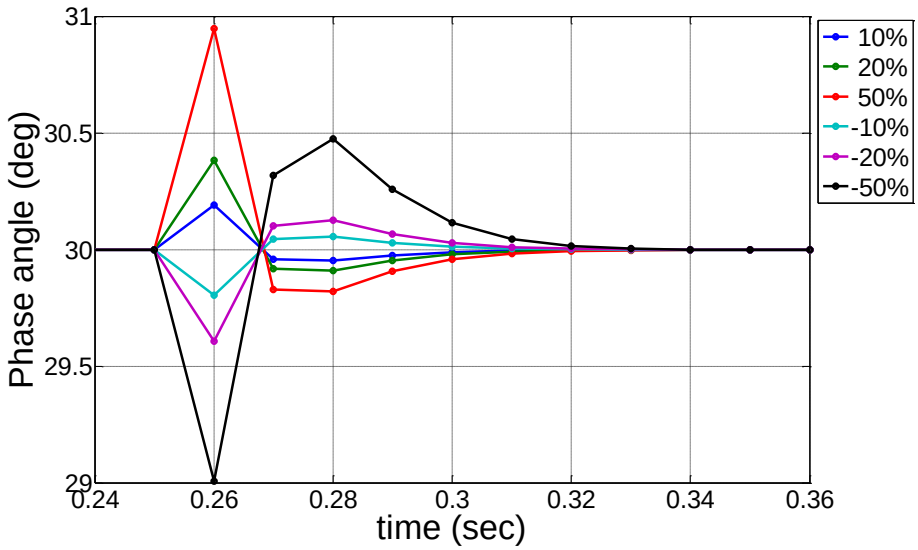
Phase angle step, degrees	10	60	90	-10	-60	-90
Elapsed Number of Frames	10	10	11	9	10	11

In Figure 4.18-a, the results of the phase angle are displayed when the step changes in amplitude are applied. During the period of steady state ($0 \leq t < 0.25$ sec), the phase angle measured remains constant at $30°$ and the TVE recorded was 0.2%. However as the step changes in amplitude applied at $t = 0.25$ sec, the estimator measures different angle deviations. The maximum deviation was $1°$ at $\pm 50\%$ amplitude step change. At $\pm 10\%$, the testing value specified by IEEE C37.118 standard, the deviation was $0.33°$.

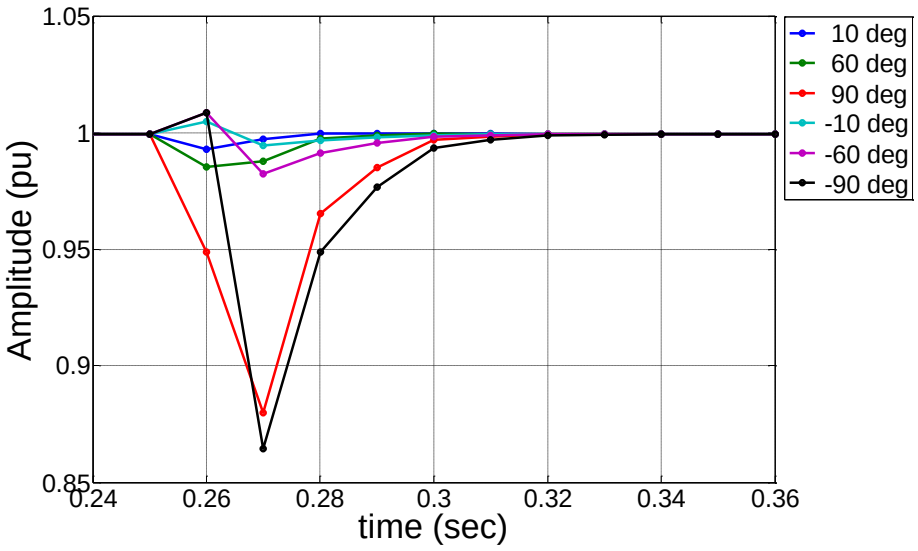
In Figure 4.18-b, the results of the amplitude are displayed when the step changes in phase angle are applied. The maximum recorded amplitude deviation was -12.5 % at step change of phase angle $-90°$. It's worth noting the effects of the phase angle sign on the amplitude step response (Response of $90°$ is less than $-90°$). The deviation values when

the transient below the steady state amplitude value (1 pu) are higher than the values above. The maximum absolute deviation value below is 0.125 while above is 0.015.

In Figure 4.19 (a and b), the calculated TVE for step and amplitude changes are shown respectively. The TVE is displayed with level with a reporting rate (F_s) of 100 frames per second and for the steady state only after the signal settled ($0.35 \leq t$ sec). However, it can be also used during transient's evaluation, so the deviation of the estimated phasors can be derived for all the period of entire test procedure clearly. For the entire step changes test, the TVE calculated has value less than 1%. When step changes of $+10\%$ and -10% are applied, TVE has values close to 0.075% and 0.175%. This reveals that TVE is more affected by a step change decrease than a step change increase. Also, some values of recorded TVE when settling to steady state after fault are getting higher than the TVE value recorded at steady state before the fault occurrence (0.2%). The maximum TVE was 0.3% when 50% step change in amplitude was applied and about 0.35% when $\pm 60°$ step changes in phase angle were applied.

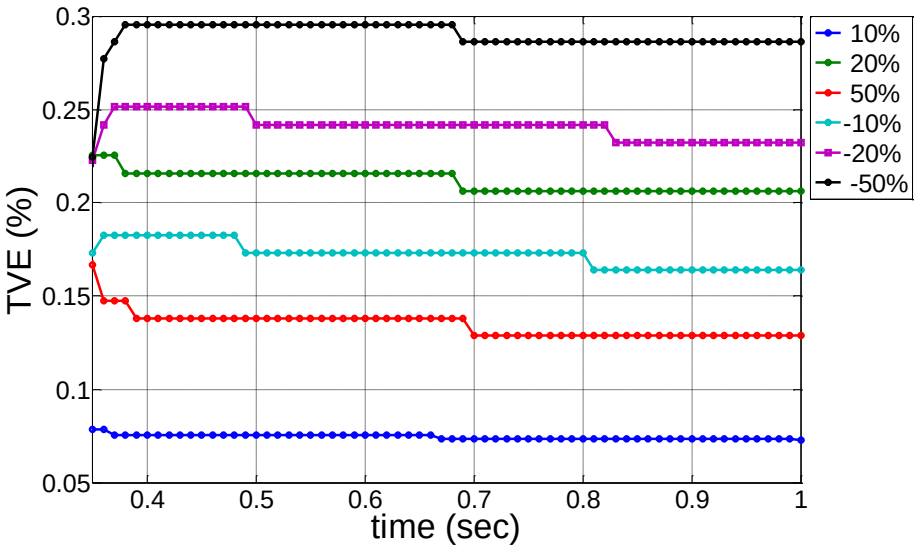


a)

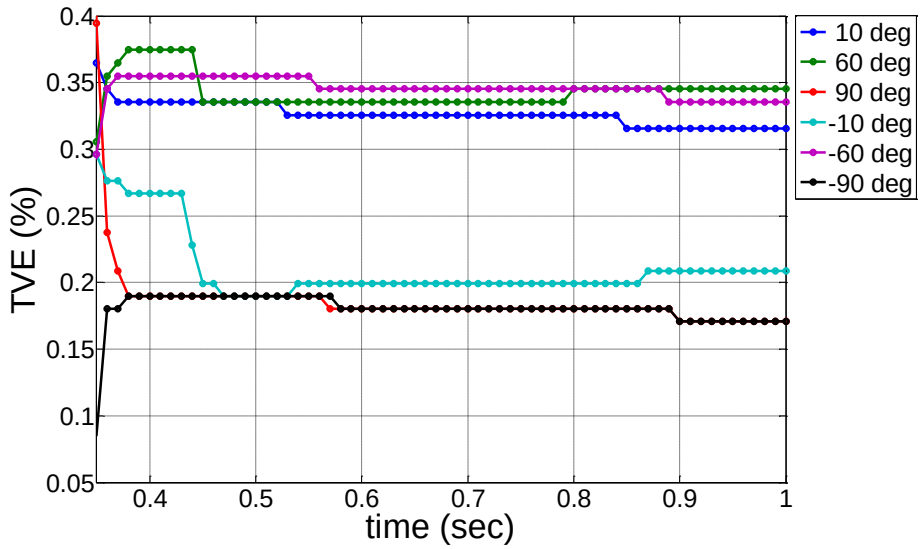


b)

Figure 4.18. a) Effect of Step Changes in Amplitude on Estimated Phase Angle, b) Effect of Step Changes in Phase Angle on Estimated Amplitude



a)



b)

Figure 4.19. a) Effect of Step Changes in Amplitude on TVE, b) Effect of Step Changes in Phase Angle on TVE

4.5 Summary

Test results in this Chapter has shown that, the phasor estimator can be implanted by UKF and for a number of nodes equal to thirty nodes at a Band-Width of 10 Gbit/sec and a sampling frequency of 4 kHz without any problem related to the Total Vector Error. Also, it has been shown that the Frame Size has a deep impact on the TVE if the frame has more than one ASDU at the maximum Nodes Number.

Moreover, results showed that allowable values of TVE and Latency can be satisfied with all LAN networks but at different values of BW. It's found that both CSMA/CD and CSMA/AMP networks have the same value of BW (but lesser than others) at which the TVE has a value less than allowable in IEEE C37.118 standard. Also, for Latency, it has been found that the Switched Ethernet needs less BW than other to satisfy the value of latency less than that specified in IEC 61850.

Also, the simulations results showed, if the BW assigned to the Process-Bus is equal or higher than the nominal value of the BW needed to handle the total BW required by all the MUs, then, UKF will estimate the Phasor at bay level without any limitation. If time errors exist between the measured and control nodes, this will have more effect on the estimated phase angle rather the estimated magnitude.

The proposed phasor estimator can produce phasors at bay level with reporting rate of 100 frames per second, which is encouraged in IEEE Std C37.118 and can satisfy higher accuracies than provided by the traditional Phasor Measurement Units, which are 0.01 degree and 0.01% for phase and magnitude, respectively.

Chapter 5

5. Simulation and Testing of the Over-current Protection System Based on IEC 61850 Process-Buses and Dynamic Estimator

This Chapter presents a simulation and testing of the over-current protective system based on IEC 61850-9-2 SMVs communication and a dynamic estimator using a simulator of the real-time environment. Performance of the IEC 61850 Process-Bus communication can be affected by many factors like, bandwidth and cable length. These factors will affect the measurements provided to protective relays through Process-Bus due to delay or loss of some Sampled Measured Values. The purpose of the tests in this Chapter is to show how an overcurrent protective relay would react to operation conditions of the IEC 61850 Process-Bus. Understanding how a physical protection system can respond to IEC 61850 Process-Bus communication will give engineers the confidence that is needed to assure that the system under consideration will behave as acceptable.

5.1 Introduction

In real-time power system studies, real-time simulators are needed for testing real-time performance of protection and control devices. In this Chapter, we propose use of a cost-effective alternative which is the real-time environment simulator presented in Chapter 2. In case of testing a protective relay performance, measurement signals (i.e. voltage or current) provided to the relay must be in real-time as well as sending trip and reclose signals. These signals depend on the communication system operating conditions and we are in particular interested in the impact of the communication latency and loss of samples. Since the real-time environment simulators have the capability to simulate various communication conditions, the performance of any protection and control device in real-time can be evaluated.

In the proposed testing, IEC 61850-9-2 SMVs sent by MU can be applied as voltage and current signals to a digital relay instead of using the traditional analogue values. According to the standard, for protection applications, the transmission time of the IEC 61850-9-2 SMV message should be less than 3 msec. The MU is an IEC 61850 device that is used to generate SMVs resulting from the transducers at bay-level and is sending them as the IEC 61850-9-2 messages over the Process-Bus which is Ethernet-based. Messages sent by MUs contain up to 8 signals of current and voltage (i.e. signals per phase and neutral).

Ethernet-based technology can offer a data rate up to 10 gbps with the physical mediums such as coaxial cable, twisted pair or optical fiber. The communication over the wired Ethernet-based technology has been standardized as IEEE 802.3. The first Ethernet-based technology used a copper co-axial or a twisted pair cable to provide a data rate up to 10 Mbps, then later; the optical fiber (100BASE-BX10) was able to achieve a data rate up to100 Mbps speed. Currently, fiber optic based on IEEE 802.3 can support a data rate up to 10 gbps (10GBASE-ER) with distance up to 40 km. Fiber-optic Ethernet-based technology provides a very high data rate and Electromagnetic interference/ radio-frequency interference (EMI/RFI) noise immunity characteristics, which make it suitable in the applications of the high-voltage operating environments [130]. Fiber-optic Ethernet-based technology is quite suitable for the large amounts of critical data, but however is not suitable when the location is remote due to economical reason.

The IEC 61850-9-2 communication standard defines three values for network BW. These values are standardized at 0.1, 1 and 10 gbps transmission within Local Area Networks (LAN). Even though Latency is considered very small in LAN networks, the network BW can play a big rule where some samples can be lost due to the insufficient BW assigned. The LAN transmission can produce a significant error which can affect the performance of protective system. In most circumstances, this error can be minimized by choosing the appropriate LAN.

The IEC 61850 protocol does not confine the type of LAN used for communications within the Process-Bus. The implementation of the IEC 61850-9-2 communications

requires that all measurements or control signals that are sent to and received from the protective relay should be transmitted via a separate Process-Bus which is Ethernet-based. The Process-Bus requires this Ethernet-based communication to pass up to 8 SMVs to each relay in real-time. The distribution of SMVs through Process-Bus allows simple isolation of individual equipment without disturbing the transmission of digital values of voltage and current to different parts of substation.

The contribution in this thesis is the implementation of an over-current relay that is based on powerful dynamic phasor estimation technique like UKF. The UKF is fed by the SMVs as suggested in chapter 3. That's means, the analog inputs to the conventional relays are replaced by binary I/Os. The analog signals received from current or voltage transformers which were digitized by using the MUs into digitized signals (SMV) and casted to the protection relays in the bay level over the Ethernet based Process-Bus will be used to feed the Kalman filter function built in the digital protection relay.

This chapter describes the real-time environment for testing such configuration. The application example of such system is a simple inverse-time over-current protection of a radial feeder. The PC-based real-time environment simulation tool has been used to simulate the system composed of MU, Ethernet based SMV transport, UKF phasor estimator and over-current protection function.

5.2 Implementation of Digital Over-current Relay Based on IEC 61850-9-2 Process-Bus

5.2.1 Overcurrent Protection Function

In power system protection, inverse definite minimum time (IDMT) over-current relays are employed to protect systems from the excessive currents that occur either due to short-circuits or over-load conditions. In this Chapter, we use an example of distribution feeder protection based on over-current relays. Any over-current relay that is employed should be

coordinated with other relays in system so if the primary protection relay fails to respond, then back-up one can accomplish the task [131, 132]. In microprocessor-based protection technology, it is important to understand impact of filtering nuisance signal components such as decaying DC component and harmonics. These nuisance components can cause the relay to malfunction [133]. The nuisance signal components should be filtered out efficiently to prevent any reduction in the operating time of relay which can eventually cause increase of severity of the primary side fault [134, 135]. The characteristics of IDMT over-current relays can be varied according to the required time needed for tripping. For this purpose, IEC 60255 standard for electric relays [136] defined and standardized a number of characteristics as follows:

1. Standard Inverse (SI)
2. Very Inverse (VI)
3. Extremely Inverse (EI)
4. Definite Time (DT)

Even though, all of these overcurrent relays can be fitted with a high-set instantaneous element, use of the standard SI can prove satisfactory in most cases. The mathematical description of standard SI inverse time current characteristic of an over-current relay can be written as:

$$T = TMS \times \frac{K}{I_a{}^n - 1} \tag{5.1}$$

where, I_a is the normalized fault current (i.e. $I_a = \frac{I_c}{I_p}$); I_c and I_p are the actual and pickup current, respectively. 'K' is a constant which represents the relay operating time, 'n' represents the relay inverse characteristics and TMS is the time multiplier setting. The actual fault can be obtained by modifying Equation (5.1) and assuming TMS is equals to 1.

To achieve the desired relay characteristics, the value of I_c should be raised to a power of n and then integrated as follows:

$$\text{Intg} = \int I_c^n \, dt \tag{5.2}$$

As long as the value of the excess current is below of the pickup current, the value of the integrator will keep rising until it equals to the pre-set value of constant K to cause the relay to send its trip signal. When the value of the excess current is temporary, the rising integral will be reset to zero if the current become less than the pickup current before reaching the pre-set value of the constant K. Figure 5.1 shows the logic diagram of microprocessor-based over-current relay.

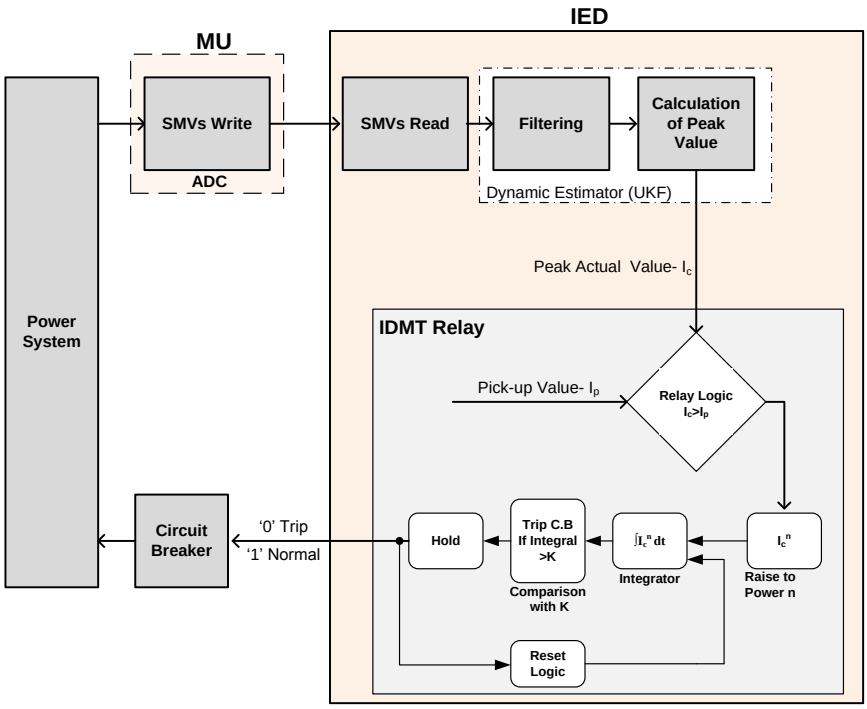


Figure 5.1. Block Diagram of Digital-Based Over-Current Relay

5.2.2 Measurement of Current Peak Value

For the purpose of filtering nuisance signal components and determining the peak value of the actual current, the UKF proposed in section 3.4 is employed. Among the several methods for amplitude and phase estimation of non-stationary signals like Frequency-based algorithms such as Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT), artificial neural networks, linear prediction technique adaptive filter, supervised Gauss-Newton algorithm and least-error square, UKF recognized as one of the most powerful algorithms for estimation [17, 19-21, 25, 107, 118, 119, 137, 138]. The DFT and FFT methods suffer from aliasing, leakage and picket fence effects and hence need error compensation and adaptive window width. Some of the known signal processing techniques like artificial neural networks, linear prediction technique adaptive filter, supervised Gauss-Newton algorithm, least-error square and its variants, extended Kalman filters, have been used for time-varying signal parameter estimation. Most of these algorithms require heavy computational outlay and suffer from inaccuracies in the presence of noise with low signal to noise ratio (SNR).

Traditional Relays employ the Fourier algorithm for estimating the signals amplitude. Fourier algorithm uses one period of measured samples to estimate the frequency content. It use the transformation, shown in Equation (5.3)

$$\mathrm{I}_h = \frac{2}{\mathrm{N}} \cdot \sum_{\mathrm{n}=0}^{\mathrm{N}-1} i(\mathrm{n} \cdot \Delta t) \left[\cos\left(\frac{2\pi \mathrm{n} h}{\mathrm{N}}\right) - j \sin(\frac{2\pi \mathrm{n} h}{\mathrm{N}})\right] \tag{5.3}$$

Where I_h is the estimated harmonic current signal, N is the number of samples in one period, n is the sample number and h is the number of the harmonic frequency being calculated. This is a rather simple method to implement, but it has it's drawbacks, such as that it is dependent on samples from a whole period, which makes the prediction relatively slow. In contrast, Kalman filter uses recursive calculations for estimating the signals. It suffices with only one sample to estimate the signal and which reduces the memory storage when it implanted in digital relays. When estimating the current signal, using the Kalman

filter method, $i(n+1)$ should be measured and compared to the estimated current $\hat{i}(n+1)$. One of the benefits of the Kalman estimation is that it is not an average of present and past samples within a sample window, but a statistical weighting between the old estimated value and the new sample.

For calculation of the peak value of the actual current, we consider the amplitude provided by the phasor estimation from the UKF presented in section 3.4; so it can be compared to the pick-up value of the overcurrent protection function. It's worth emphasizing that the covariance of the additive white-noise plays a big rule in satisfying the required accuracy of UKF [139]. In section 5.3.45.3.4, the effect of covariance on the estimated peak value and hence the response time of the digital relay will be tested and evaluated.

5.2.3 Simulator Implementation of the 61850-9-2 based Over-Current Protection

For simulating the IEC 61850-9-2 Process-Bus, the Write/Read functions of SMVs created in Chapter 2 have been used again. For the over-current protection of distribution feeder a MU should be installed to provide digital current measurements in the SMV format to a relay. The MU is set to work at standardized sampling rate of 4 kHz (80 samples/cycle). The BW of the Process-Bus is made adjustable to the different IEC 61850 standardized values (i.e. 0.1, 1 and 10 gbps). If the connection between the MU and the digital relay is not optical cable, then the cable length is considered as another factor which could also affect the digital relay performance. In this simulation, the cable length has been also made adjustable and we assigned a maximum value of 100 meters to emulate the real LAN.

The testing environment consists of two models representing IEC61850-9-2 and Intelligent Electronic Device (IED). The main functionality of the IEC61850-9-2 model is to produce (write) SMV at standardized sample rate of 4 kHz. The IED model includes the read function of SMV (to read and unwrap the current measurements casted by IEC61850-9-2 SMVs), and the phasor estimator to produce the amplitude and the digital over-current

relay. The digital over-current relay has been developed in SIMULINK. The SIMULINK model (i.e. digital over-current relay) and TrueTime models (i.e. Write/Read function) and all other models in testing environment (Radial feeder system model or substation model) converted to a Matlab's xPC target model; so it can be possible to work in real-time with the simulated signals.

This simulation-based testing platform provides flexible way of testing the whole IEC61850 measurement system. For the case of one node connected to protection relay, a simple radial feeder system is used as shown in Figure 5.2. The setting of the pick-up current allows the load to carry the continuous full-load current (i.e. 179A rms or 253A peak). The fault current (I_s) is more than the starting current (1000A peak). The constant K=100 has been selected in such that it does not cause false tripping during start-up and any other transient conditions. The value of n is set to be 0.9. For the relay, if we assume the fault current is 770 A and it's wanted to have a relay operating time of 0.25 sec; then K constant can be found from Equation (5.1).

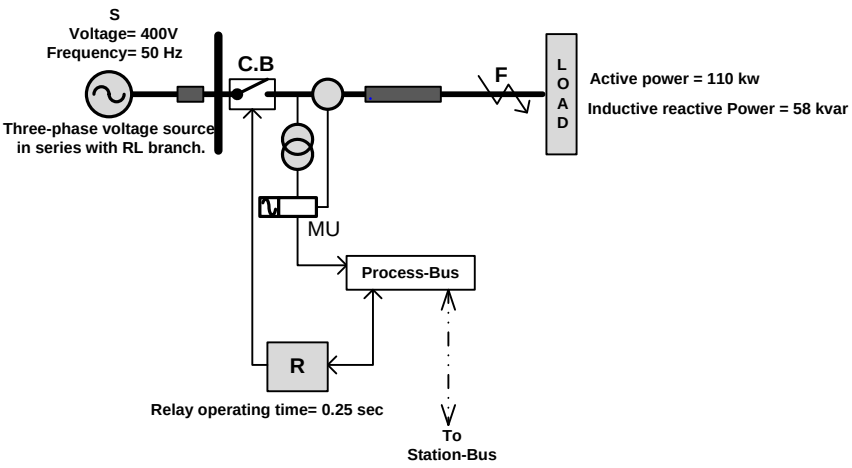


Figure 5.2. Simple Radial Feeder Network with one Relay Connected to Process-Bus

5.3 Test Results and Evaluations

To understand impact of the IEC 61850-9-2 Process- Bus communication on the relay performance, a varied testing of the system operation for different operation factors are

necessary. In order to examine the over-current protection system based on the IEC 61850-9-2 Process-Bus, various scenarios such as different BW, losses in SMV, cable lengths and covariance of the measurement noise needs to be considered. The inverse time over-current relay characteristic is set to protect distribution feeder by using the integration of the fault current. The operation time of the relay has been observed after fault inception (ground fault) simulated as a step change in the current signal produced by the MU. The fault current was increased suddenly from 253 A to 560 A (120 % step change). This step change can be tracked by the dynamic estimator (UKF) for the peak value of the actual current.

5.3.1 The BW Impact

For testing the impact of BW assigned to IEC 61850-9-2 Process-Bus, the BW has been made adjustable to one of the standardized values (0.1, 1 and 10 gbps) according to IEC 61850 to handle the communications of installed MU and the cable length has been made fixed to 100 meters. Figure 5.3 shows the channel utilization at each value for BW. At 0.1 gbps, the channel utilization becomes very high and reaches a value higher than 80%. This higher value of channel utilization produces a traffic collision that made some of SMVs to be delayed or lost. The delayed or loss of SMVs cause a delay in timing of the relay trip signal due to more integration time is needed to reach the pre-determined value of K as shown in Figure 5.4 and Figure 5.5.

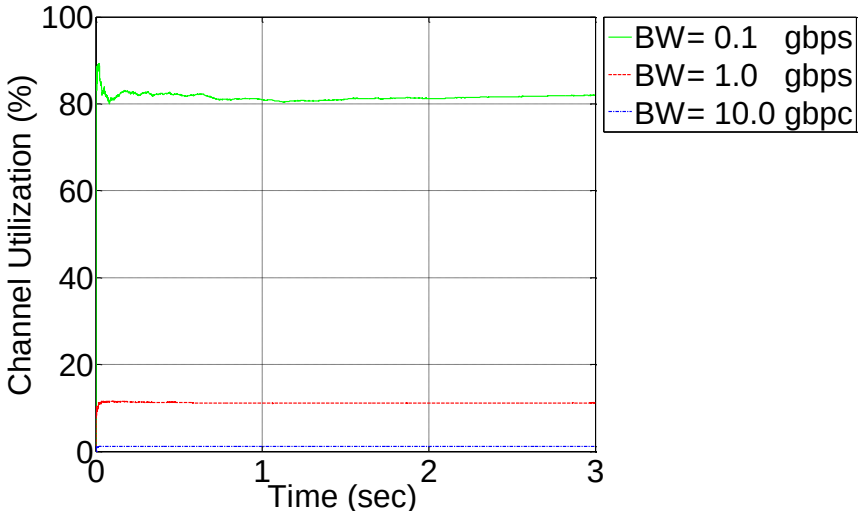


Figure 5.3. Channel Utilization

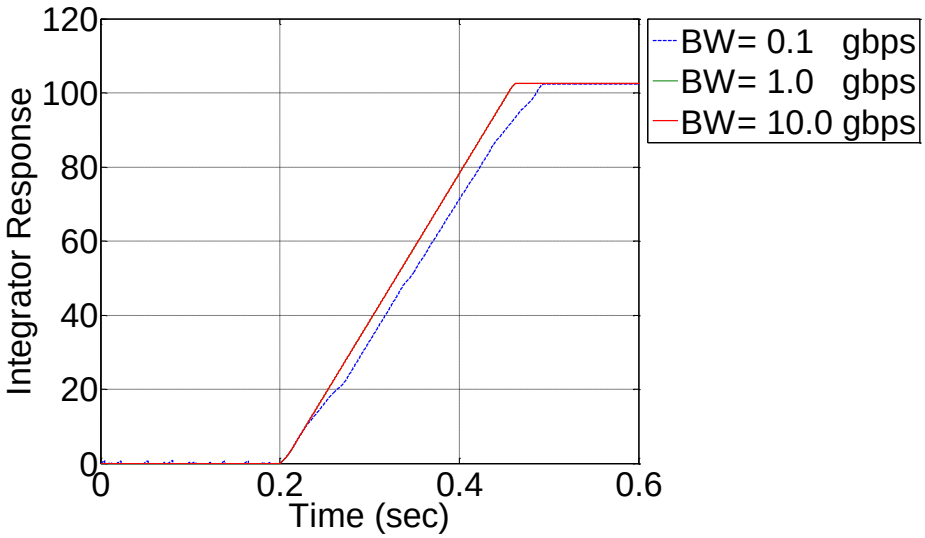


Figure 5.4. Integrator Response for Three Band-Widths (BWs)

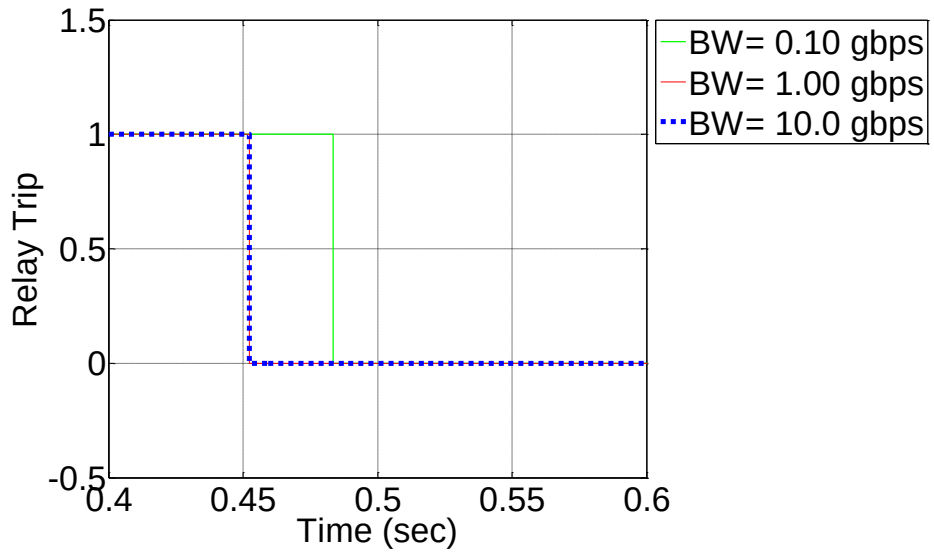


Figure 5.5. Relay Trip for Three Band-Widths (BWs)

5.3.2 Impact of LAN Type

As mentioned earlier in section 4.3.1.5, the transmission of packets in LANs can be done by one of three different techniques:

1. CSMA/CD (e.g. Ethernet),
2. CSMA/ AMP (e.g. CAN),
3. Switched Ethernet.

The delay of trip signal can be affected by the different types of LAN network. Each type of LAN network can handles specific number of nodes and beyond that number, a digital relay becomes non-responsive within the pre-timed settings for each type of communication due to the high traffic of communication at these handled nodes which make some SMV to be lost. To test the effect of BW on different types of LAN networks, the BW was been set to the minimum standardized value of IEC 61850 (i.e. 0.01 gbps).

The Nodes Number was made changeable from 1 to 8 Nodes despite that one Node (MU) only is needed in our application.

Figure 5.76 shows the delay in trip signal and how Switched Ethernet type is able to handle number of Nodes that others types of LAN can't handle for the same number of Nodes. The digital relay is not able to respond (send a trip signal) after four Nodes when CSMA/ AMP type is used and its delay increases dramatically more than that of other LAN types. The change in trip delay for the Switched Ethernet is very small and can be eliminated while the change in trip delay for CSMA/ AMP reaches its maximum at four nodes.

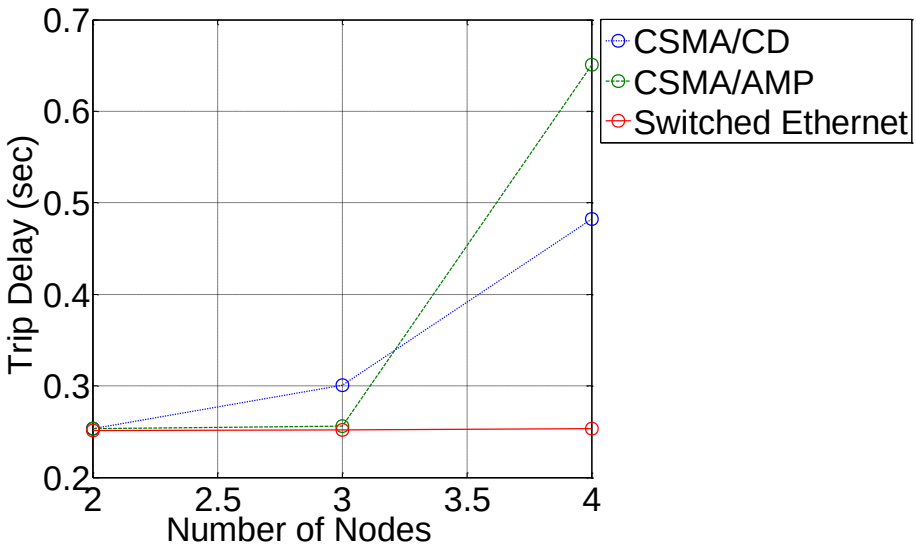


Figure 5.6. Delay in Trip Signal for Different Types of LAN

5.3.3 The Nodes Number Impact

It's more interesting to see what can happen if the number of nodes that LAN can handle is increased to utilize BW to full capacity. This case can happen in a substation with Process-Bus having many nodes. To simulate this case, a substation layout that comprises a breaker is considered with the MUs assigning as shown in Figure 5.7. In this substation layout, there are a total of 15 CTs. Each circuit breaker has one CT installed at the frontend and

backend. Additionally, it has been considered that every MU can be configured to handle 8 analog signals from 2 different three phase instrument transformers (CTs/CCVTs) and one circuit breaker. Therefore, a total of 12 MUs are needed for this substation for the protection and control.

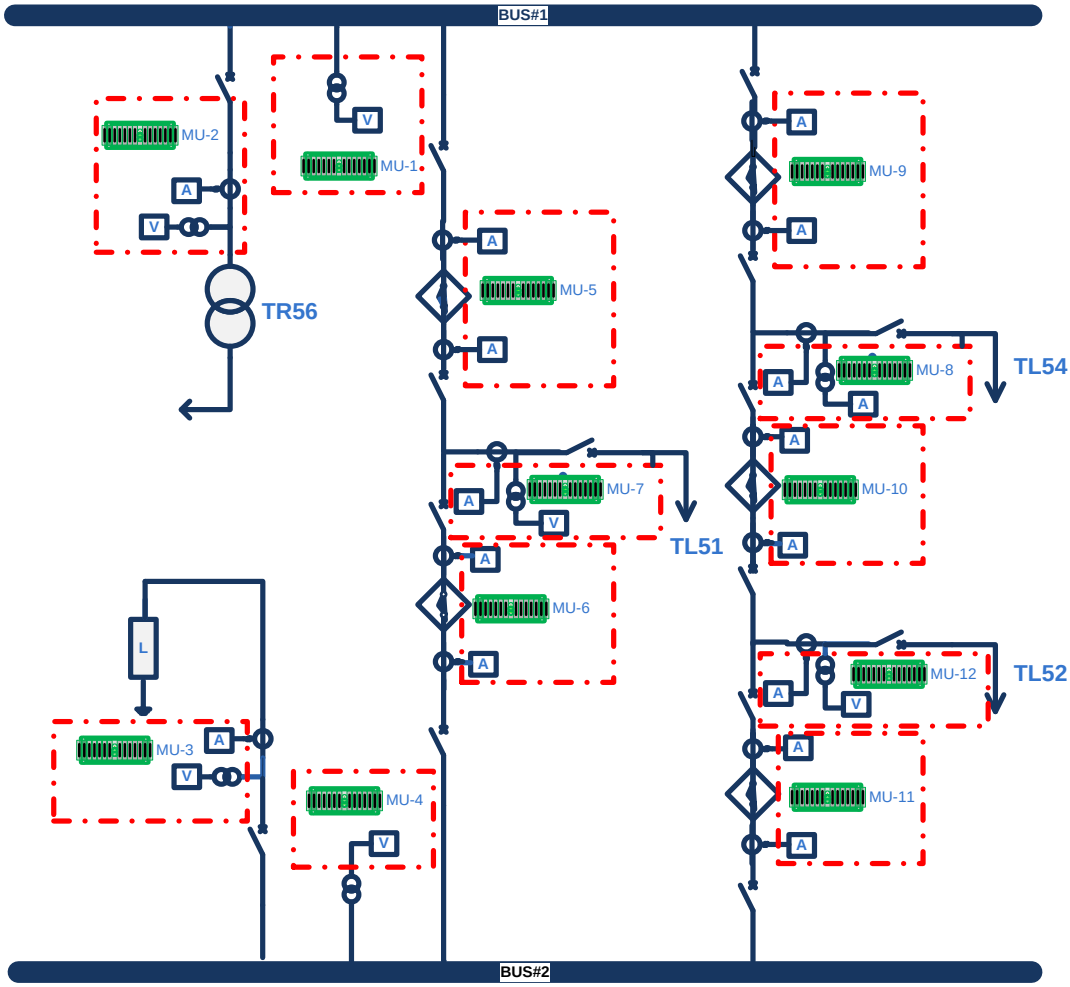


Figure 5.7. A substation Layout with the MUs Assigning

From Table 5.1, it's clear that at BW of 0.1 gbps the time delay of the relay response increases to almost double at 5 nodes and the relay becomes an inoperative beyond this number of nodes. For 1 gbps, the relay becomes an inoperative beyond 21 nodes and the

time delay is almost the same for all nodes before that. Table 5.2 shows the case when the LAN communication depends on CSMA/AMP.

It's obvious that CSMA/AMP communication can handle more nodes than CSMA/CD at 1 gbps. For BW of 0.1 gbps, both types of LAN communications failed to handle the nodes number required for this substation where the relays become inoperative, but behave almost the same and satisfy the requirements for both LAN at 1 and 10 gbps.

Table 5.1. Trip signal vs. Nodes number for different BW when LAN is CSMA/CD

Nodes Number	CSMA/CD		
	0.1 gbps	1 gbps	10 gbps
1	0.452	0.452	0.450
2	0.453	--	--
3	0.684	--	--
4	0.817	--	--
5	0.888	--	--
6	No Trip	--	--
7	--	--	--
..	..	..	..
..	..	..	..
20	--	0.457	--
21	--	0.458	--
22	--	No Trip	--

Table 5.2. Trip signal vs. Nodes number for different BW when LAN is CSMA/AMP

Nodes Number	CSMA/AMP		
	0.1 gbps	1 gbps	10 gbps
1	0.452	0.452	0.450
2	--	--	--
3	--	--	--
4	0.855	--	--
5	No Trip	--	--
6	--	--	--
7	--	--	--
..	..	..	..
..	..	..	..
..	..	..	..
33	--	0.678	--

34	--	No Trip	--

5.3.4 The Noise Impact

In [140], it has been shown that the maximum value of noise covariance has been restricted to a maximum value of 4×10^{-4} to keep the Total Vector Error (TVE) of the phasor estimated by the KF within the allowable accuracy defined in IEEE C37.118 standard [95]. For testing the impact of noise covariance on the relay performance under IEC 61850-9-2 Process-Bus communication, three values (2×10^{-4}, 3×10^{-4} and 4×10^{-4}) of noise covariance have been arbitrary selected between zero and the maximum covariance value presented in [140]. Figure 5.8 shows the impact of increasing covariance (R) from $2*10^{-4}$ to maximum value of 4×10^{-4} on the relay integrator. The time needed for the integrator to reach its pre-set value of K is the highest when $R=4\times10^{-4}$.

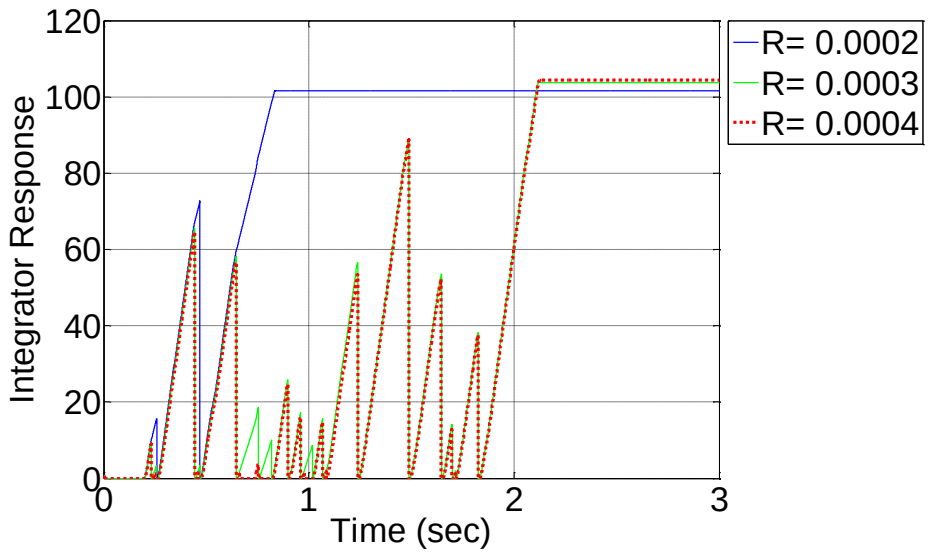


Figure 5.8. Integrator Response for Three Covariance Values

It's worth noting that the noise covariance has a stronger impact than BW on the relay performance whereas the delay in trip signals in case of maximum value of R has bigger value than that presented in case of assigning a minimum BW as shown in Figure 5.9 .

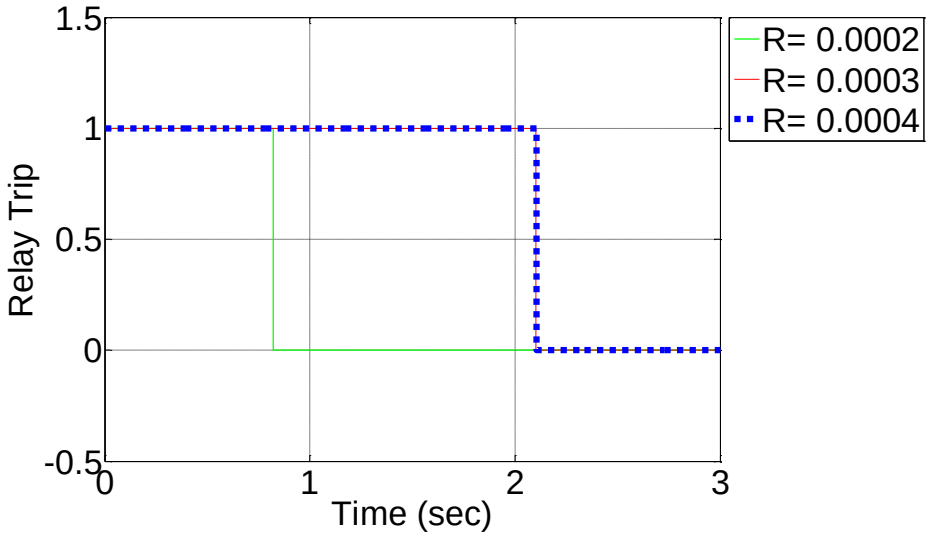


Figure 5.9. Trip Signal for Three Covariance (R) Values

To draw the relation between the covariance change and the relay trip, the length of the cable has been set to minimum of 10 meters to avoid its effect on the digital over-current relay trip time. Any increase in R, can lead to delay in trip signal in nonlinear manner. Figure 5.10 shows the amount of this delay for arbitrary values of 10^{-4}, 2×10^{-4}, 3×10^{-4} and 4×10^{-4}.

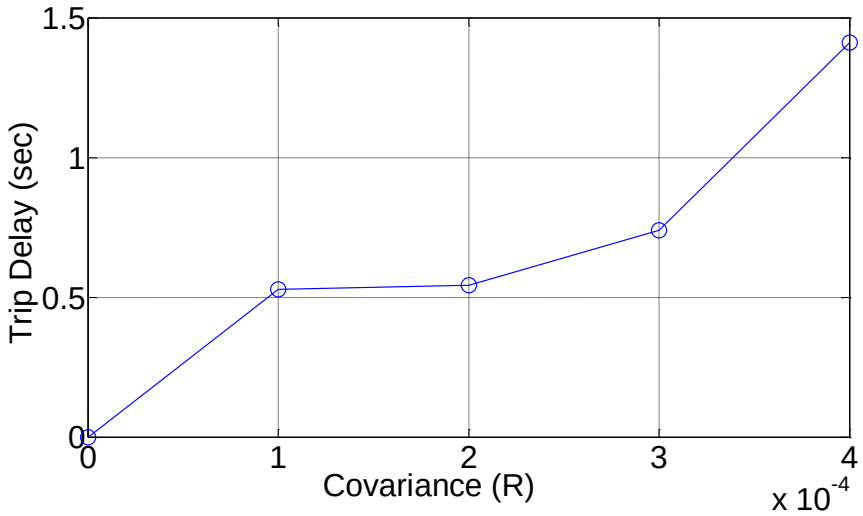


Figure 5.10. Trip Delays Introduced for Different Arbitrary Values of R

As can be seen from the Figure 5.10, the relation between the change in R and the trip delay is not linear. For each value, the total trip delay is 0.3286, 0.3436, 0.5387 and 1.2119 seconds, respectively. To determine the trip delay caused by KF for the curve in Figure 5.10, regression techniques (e.g. Polynomial Regression) can be applied. For the arbitrary values of 10^{-4}, 2×10^{-4}, 3×10^{-4} and 4×10^{-4}, the delays are shown in Table 5.3.

Table 5.3. Effect of R Increase on Trip Signal Delay

Covariance (R)	Total Delay (sec)	Trip delay (sec)	Delay by KF (sec)
0.0001	0.51	0.3286	0.0786
0.0002	0.52	0.3436	0.0936
0.0003	0.75	0.5387	0.2887
0.0004	1.4	1.2119	0.9619

The digital over-current relay shown in Figure 5.1 can be adapted to remove any delay that can be represented by the curve in Figure 5.10. This can be done by subtracting the trip time delay estimated by regression (T_{est}) at which fault occurred from the total delay time (T) according to Figure 5.11 and can also be represented in mathematical equation as follows:

From the general form of the polynomial regression that can be represented by Equation (5.4) below:

$$T_{est}(R) = p_0 + \sum_{j=1}^{k} p_j R^j \tag{5.4}$$

where p_0 is an optional constant term and p_1 through p_k are coefficients of increasing powers of R. We find the coefficients and solving for T_{est}. Substitute Equation (5.4) in Equation (5.1) yields:

$$K_{est} = \big((T_{est} - T) - T\big)\times T\times {I_c}^n \tag{5.5}$$

When R= 4×10^{-4}, the signal amplitude estimated by KF will has a higher distortion that can increase the delay in the trip signal to its maximum value. In this case, the delay will be very high compared to delay at R= 2×10^{-4} for example. This is shown in Figure 5.12. After removing the delay presented when R= 4×10^{-4} by KF and setting the constant K to the new one that is estimated by equation 5.6, the time of trip signal has been corrected from delay equals to T=1.4144 sec to a new value of T= 0.466 which is at K_{est}= 70.135 shown in Figure 5.13.

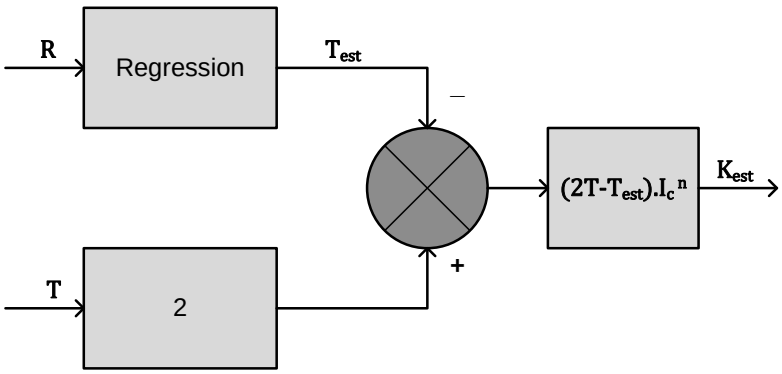


Figure 5.11. Adapted Model for Constant K

5.3.5 Impact of Cable Length

Ethernet LAN specification imposes that the length of cable can extend up to 100 meters. If more than 100 meters is required, a simple repeater (i.e. a hub, or a switch) needs to be installed for the extension of the cable. Alternative is fiber optic that can extend to something like 40 km with a high electrical insulation. For cases of unavailability of fiber optic link, Cat-6 Ethernet cable can be suitable for applications of protection in substation. The cable will be extended to cover the distance between the MU and digital relay and hence the impact of its length on relay performance should be tested.

For testing the impact of cable length on relay performance, different cable lengths of 10, 50 and 100 meters are selected. A worst case is considered by assigning a BW of 0.1

gbps which is the lower standardized value of BW. Figure 5.14 through Figure 5.16 show how the cable length can affect the performance of over-current relay. The effect of cable length appears to be a delay in trip signal. The delay had been increased from about 0.2 sec to 0.5sec to 1.1 sec for cable lengths of 10, 50 and 100 meters, respectively.

The effect of cable length on relay performance can be ignored if a higher value of the standardized BW is assigned. This is clear if we compare the graphs in Figure 5.14 with the graphs in Figure 5.17 which show how the delay is reduced when a BW of 10 gbps is assigned. The graph in Figure 5.16 represents the integrator response to a cable length of 100 metres, which behaves as a saw-toothed wave. This is because the increase in cable length makes delay to the SMVs received by KF. The delayed SMVs cause the peak values estimated by KF to fluctuate which can eventually cause the KF current graph to be distorted and not to be similar to the feeder current.

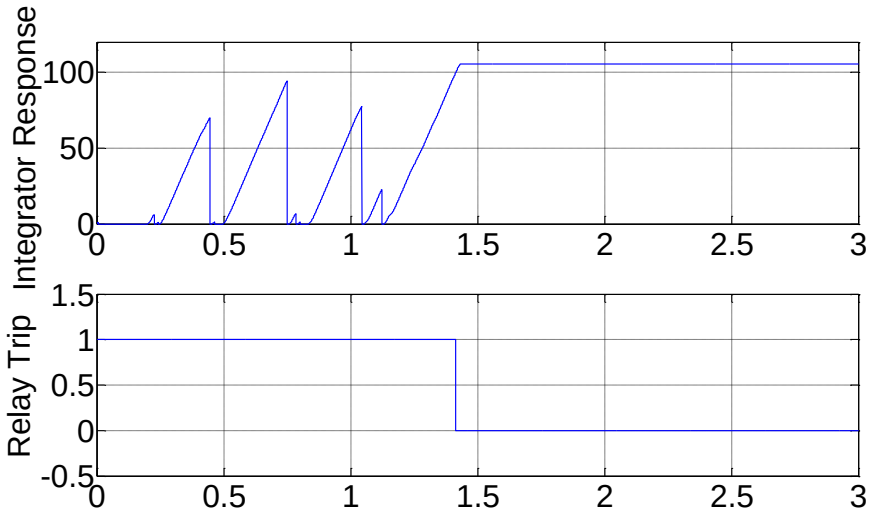


Figure 5.12. Delay in Trip Signal Before Adaption (K=99)

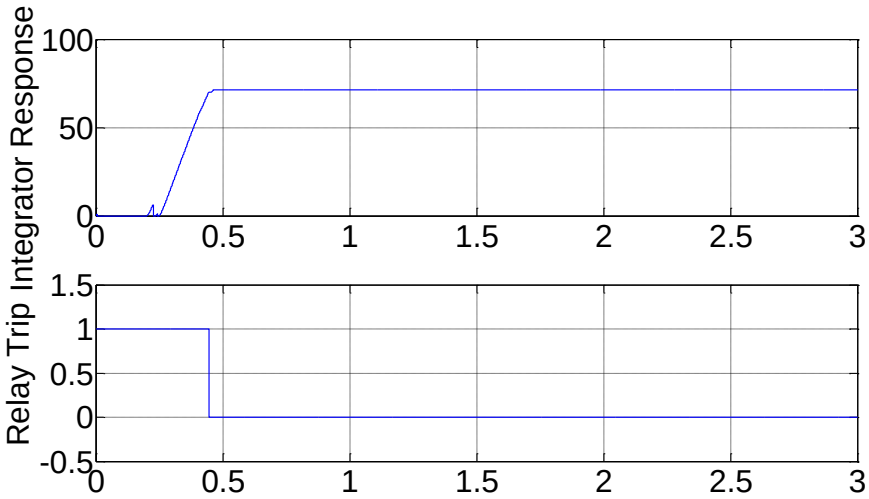


Figure 5.13. Delay in Trip Signal After Adaption (K=70.135)

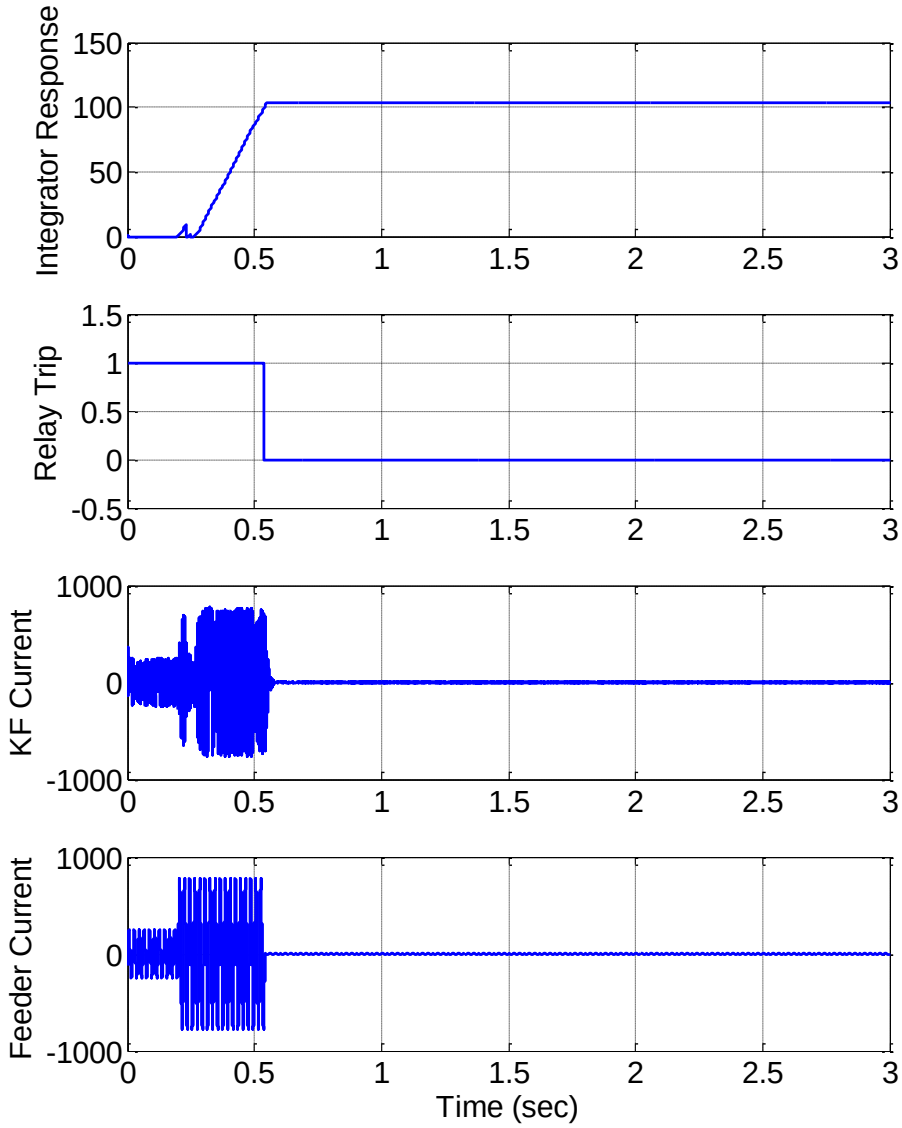


Figure 5.14. Evaluation for L=10 m and BW= 0.1 gbps

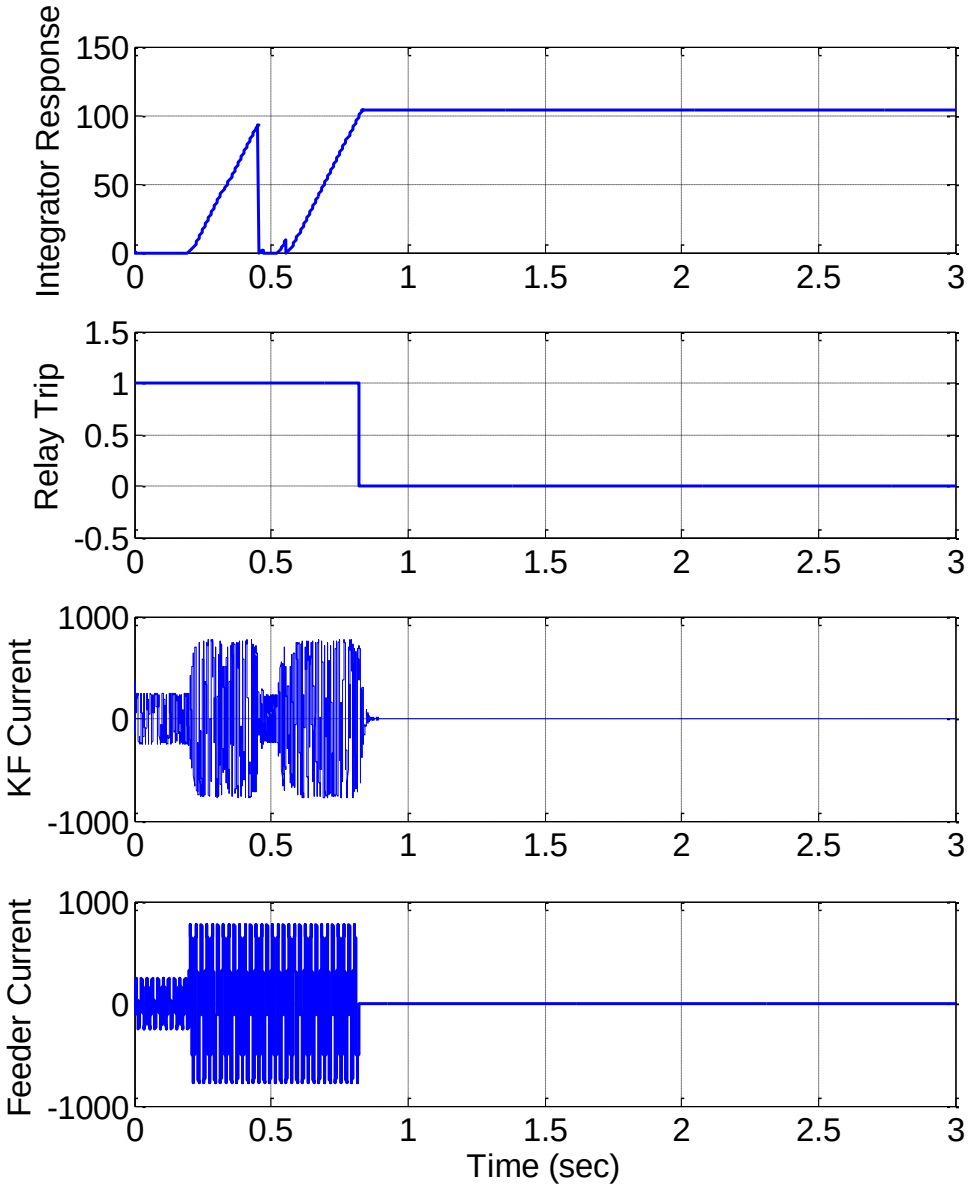


Figure 5.15. Evaluation for L=50 m and BW= 0.1 gbps

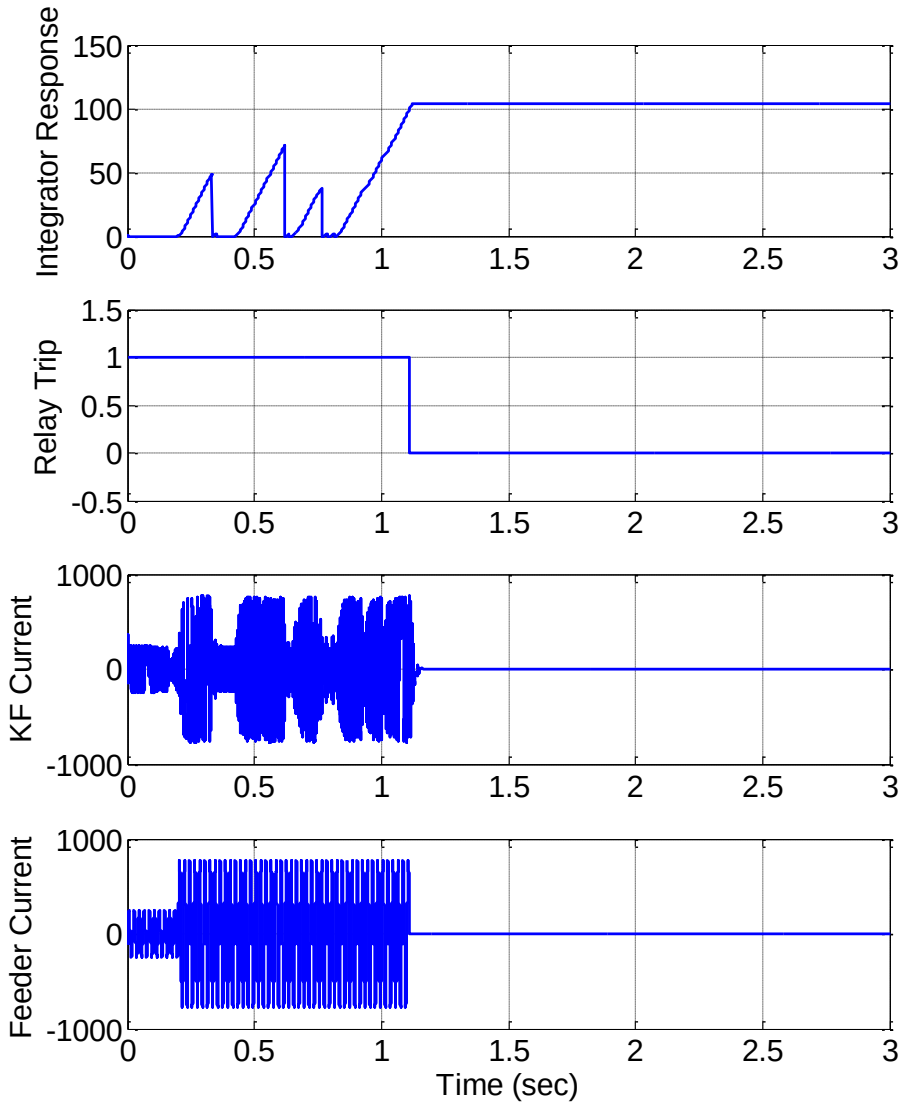


Figure 5.16. Evaluation for L=100 m and BW= 0.1 gbps

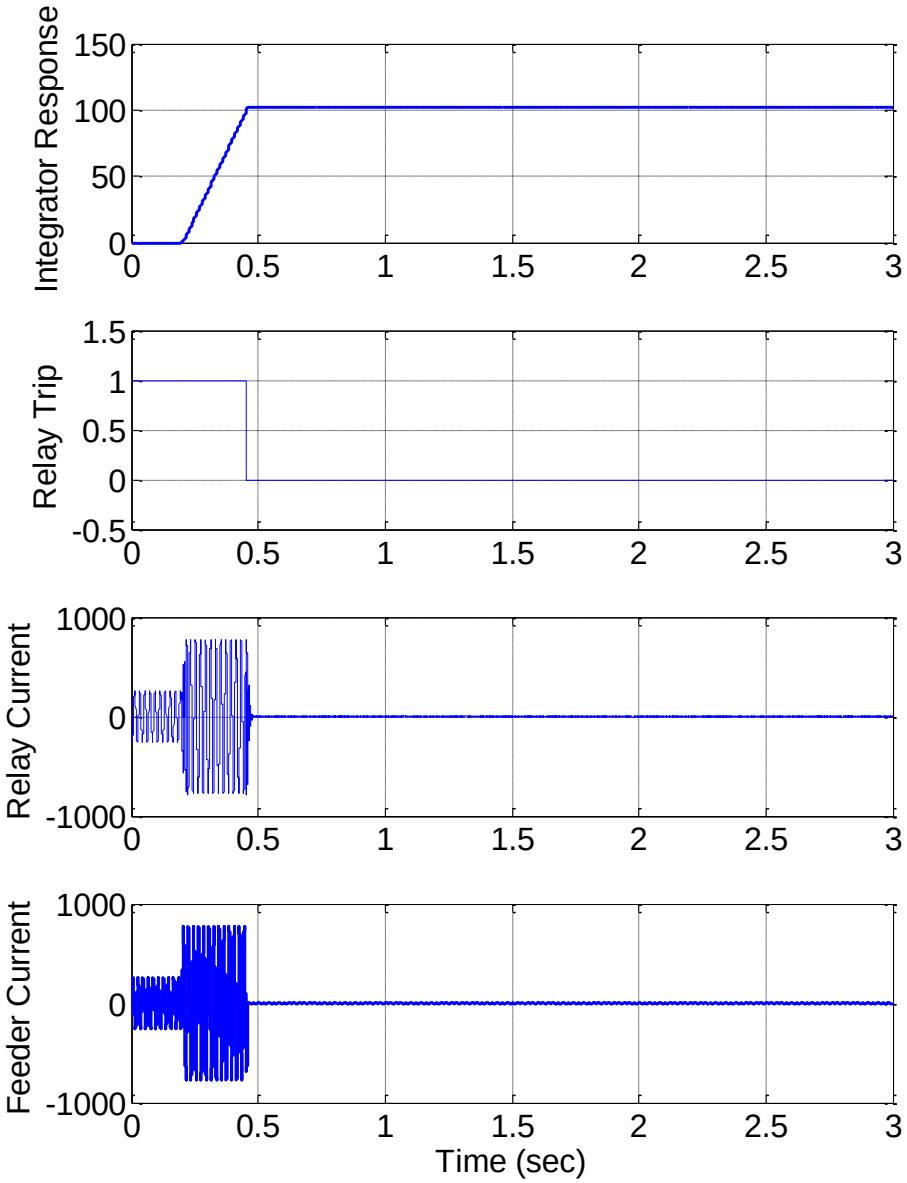


Figure 5.17. Evaluation for L=10 m and BW= 10 gbps

5.4 Summary

As the implementations of IEC 61850-9-2 Process-Bus grow into the substation automation, the performance of digital relays against normal and abnormal data received from MUs via IEC 61850-9-2 Process-Bus should be tested. Understanding the reaction of digital relays when subjected to abnormal IEC 61850-9-2 Process-Bus data is a critical in the design of digital relays to avoid any mal-operation in the whole system integration. Tests results showed that the impact of improper BW selection and cable length can contribute in the errors of the received data by relays. The integrator of over-current digital relay can be affected by these factors and then causing a delay in the trip signal. Also, tests results showed that the performance of over-current digital relay can be affected by different types of medium access packet transmission in LAN networks. Results showed that Switched Ethernet type can handle a larger number of nodes that others types of LAN can’t handle.

Chapter 6

6. Research Outcomes and Conclusions

The work of this research had addressed the major technical challenges related to IEC 61850-9-2 based Process-Bus which can affect the performance of phasors estimation and hence protection systems in substations. The detailed performance of dynamic phasor estimator implemented by Unscented Kalman filter and relying on the state of the art in substation measurements which is IEC 61850-9-2 SMVs is presented. Also, the performance and reliability analysis of an over-current protection function based on this dynamic phasor estimator are presented. The specific outcomes and conclusions of this research work will be discussed in this chapter.

6.1 Outcomes of the Research Work

This research work provides major contributions to the area of power and energy monitoring system, as well as protection of power system as follows:

1. Implementation of two codes, write and read for Sampled Measured Values using C programming in real time Linux system and integrate them with TrueTime to emulate the Merging Unit and IED as defend in IEC61850 standard.
2. Developing dynamic phasor estimator to estimate phasors dynamically of all buses at bay level instead of process level as the case in PMU, which can decrease the cost of deploying PMUs at all buses.
3. The proposed phasor estimator can produce phasors at bay level and can satisfy the allowable TVE at a reporting rate of 100 frames per second as recommended in IEEE C37.118 and can also satisfy higher accuracies than provided by the traditional PMUs, which are 0.01 degree and 0.01% for phase and magnitude, respectively.

4. The developed tools for the IEC61850-9-2 communication and phasor estimation can be considered simple and cost effective platforms for the detailed analysis of dynamic phasor estimator based on IEC 61850-9-2 Process-Bus as applied to over-current protection system in substation.
5. The developed platform can recognize various constraints and communication parameters of the practical IEC 61850-9-2 based Process-Bus networks, e.g. samples loss, samples delay, tape of LAN and channel band-width on dynamic estimated phasor.
6. The platform also provides a tool for analyzing the dynamic performance of any IEC 61850-9-2 Process-Bus network based protection system in typical substation. Considering and analyzing the effects of these communication parameters will help to design a robust protection and control system based on Process-Bus networks during the developing/ planning stages of existing/new substation automation projects.

6.2 Summary and Chapter Conclusions

For reducing the cost of engineering and wiring of long copper wires between the process level and bay level/control room in substation, the IEC 61850 standard proposed the Ethernet communication network based Process-Bus. In addition to the cost reduction, the Process-Bus also has architecture that offers simplicity, flexibility and interoperability, etc. The research work is to develop dynamic phasor estimator based on IEC 61980-9-2 Process-Bus products for protection functions in substation by addressing various technical challenges that are related to this new technology. These technical issues of Process-Bus have been from the review and the study of numerous literatures which include manufacturer's technical and white papers, standards developer committee publications, international journals and conference proceedings, etc., are all discussed in Chapter 1. The major recognized technical issues discussed in this chapter are the dynamic performance (delay/loss) of time critical SMV messages over The Ethernet based Process-Bus

communication network, and also the effects of SMV delay/loss on IEC 61850-9-2 based protection IEDs installed in substation. The work in this research is carried out to study and evaluate the effects of these major technical issues.

The significant features of the IEC 61850-9-2 based Process-Bus, such as the multicasting/retransmission of time critical messages (SMV/GOOSE), the time synchronization over the Process-Bus, and the fast Ethernet switched based networks are presented in Chapter 2. Moreover, a brief overview of many architectures for practical Ethernet switched networks as suggested by the IEEE (PSRC report) [96], which include cascaded architecture, ring architecture, star-ring architecture and redundant-ring architecture. The key time synchronization techniques, such as the IRIG-B and the IEEE 1588 based PTP protocol time synchronization over Ethernet based network are also presented in this Chapter. The implementation procedures that have been used to create the SMVs publisher/subscriber functions are explained in details in the same chapter. The programming for the encapsulation of different layers of SMV frame in C programming under Linux real time system and how the Process-Bus interface is embedded into the TrueTime Matlab environment are also documented in this Chapter.

In Chapter 3, the implementation of UKF Model-Based technique for phasor estimation and studying its frequency response performance are presented. The results showed that the frequency response of the proposed estimator indicated the capability to attenuate and reject harmonics at frequencies other than the fundamental frequency. From simulation results in the same Chapter, it can be seen that, at the fundamental frequency, the proposed model provides an accurate estimation of the magnitude. With a sample rate of 80 samples/cycle, the frequency response has sharper curve. This is an indication that, it has a higher attenuation on harmonics.

Chapter 4 presented an evaluation of a new architecture for phasor estimation within IEC 61850 substation Process-Bus. The raw IEC 61850-9-2 SMV streamed with high sampling rate from various Merging Units is proposed to be fed to dynamic estimator relying on Unsented Kalman Filter for the computation of Phasor at substation process level to enhance the state estimation of dynamic phenomena. The effects of IEC 61850-9-2

communication network and the time errors due to the Asynchrony between Merging Units on the performance of the estimated Phasor are also evaluated and presented in the Chapter. As the simulation results showed, the band-width assigned to the Process-Bus should be equal or higher than the nominal value of the band-width needed so that, the total band-width required by all MUs can be handled.

Also, time errors exist between the measured and control nodes due to asynchrony will have more effect on the estimated phase angle rather the estimated magnitude. The impacts of samples loss, nodes number and frame size on the Total Vector Error as defined in IEEE C37.118 have been shown in the same Chapter. The results have shown that, the Phasor estimator can handle a number of nodes equal to thirty nodes at a Band-Width of 10 Gbit/sec and a sampling frequency of 4 kHz without any problem related to the Total Vector Error. Also, it has been shown that the Frame Size has a deeper impact on the TVE if the frame has more than one ASDU at the maximum Nodes Number. Also, the results showed in the same Chapter that the proposed phasor estimator can produce phasors at bay level with reporting rate of 100 frames per second, which is encouraged in IEEE Std C37.118 and can satisfy higher accuracies than provided by the traditional Phasor Measurement Units, which are 0.01 degree and 0.01% for phase and magnitude, respectively.

Test results in Chapter 5 showed that, the impact of improper BW selection and cable length can contribute in the errors of the received data by relays. The integrator of over-current digital relay can be affected by these factors and then causing a delay in the trip signal. Also, test results showed that the performance of over-current digital relay can be affected by different types of medium access packet transmission in LAN networks.

References

[1] A. Monticelli, *State estimation in electric power systems: a generalized approach* vol. 507: Springer, 1999.

[2] J. Minkel, "The 2003 Northeast Blackout--Five Years Later," *Scientific American,* vol. 13, 2008.

[3] P. L. Anderson and I. K. Geckil, "Northeast blackout likely to reduce US earnings by $6.4 billion," *Anderson Economic Group,* 2003.

[4] E. J. Lerner, "What's wrong with the electric grid?," *Gravitational, Electric, and Magnetic Forces: An Anthology of Current Thought,* p. 41, 2005.

[5] J. Glanz and A. C. Revkin, "Energy Dept. Will Take Control of Blackout Investigation," *The New York Times,* vol. 20, 2003.

[6] J. G. Kappenman, "Systemic failure on a grand scale: The 14 August 2003 North American blackout," *Space Weather,* vol. 1, 2003.

[7] K. Zhu, L. Nordstrom, and L. Ekstam, "Application and analysis of optimum PMU placement methods with application to state estimation accuracy," in *Power & Energy Society General Meeting, 2009. PES'09. IEEE,* 2009, pp. 1-7.

[8] Z. Hong-Shan, L. Ying, M. Zeng-Qiang, and Y. Lei, "Sensitivity constrained PMU placement for complete observability of power systems," in *Transmission and Distribution Conference and Exhibition: Asia and Pacific, 2005 IEEE/PES,* 2005, pp. 1-5.

[9] B. Xu and A. Abur, "Observability analysis and measurement placement for systems with PMUs," in *IEEE PES power systems conference and exposition,* 2004, pp. 943-946.

[10] B. Milosevic and M. Begovic, "Nondominated sorting genetic algorithm for optimal phasor measurement placement," *Power Systems, IEEE Transactions on,* vol. 18, pp. 69-75, 2003.

[11] I. Kamwa and R. Grondin, "PMU configuration for system dynamic performance measurement in large, multiarea power systems," *Power Systems, IEEE Transactions on,* vol. 17, pp. 385-394, 2002.

[12] J. Chen and A. Abur, "Improved bad data processing via strategic placement of PMUs," in *Power Engineering Society General Meeting, 2005. IEEE,* 2005, pp. 509-513.

[13] A. Phadke and R. M. de Moraes, "The wide world of wide-area measurement," *Power and Energy Magazine, IEEE,* vol. 6, pp. 52-65, 2008.

[14] J. F. Hauer, N. B. Bhatt, K. Shah, and S. Kolluri, "Performance of" WAMS East" in Providing Dynamic Information for the North East Blackout of August 14, 2003 2."

[15] A. Phadke, "Synchronized phasor measurements in power systems," *Computer Applications in Power, IEEE,* vol. 6, pp. 10-15, 1993.

[16] A. Phadke, J. Thorp, and M. Adamiak, "A new measurement technique for tracking voltage phasors, local system frequency, and rate of change of frequency," *Power Apparatus and Systems, IEEE Transactions on,* pp. 1025-1038, 1983.

[17] H. Ma and A. A. Girgis, "Identification and tracking of harmonic sources in a power system using a Kalman filter," *Power Delivery, IEEE Transactions on,* vol. 11, pp. 1659-1665, 1996.
[18] V. M. Saiz and J. B. Guadalupe, "Application of Kalman filtering for continuous real-time tracking of power system harmonics," in *Generation, Transmission and Distribution, IEE Proceedings-*, 1997, pp. 13-20.
[19] P. Dash, A. Pradhan, and G. Panda, "Frequency estimation of distorted power system signals using extended complex Kalman filter," *Power Delivery, IEEE Transactions on,* vol. 14, pp. 761-766, 1999.
[20] S. Bittanti and S. M. Savaresi, "On the parametrization and design of an extended Kalman filter frequency tracker," *Automatic Control, IEEE Transactions on,* vol. 45, pp. 1718-1724, 2000.
[21] P. Dash, R. Jena, G. Panda, and A. Routray, "An extended complex Kalman filter for frequency measurement of distorted signals," *Instrumentation and Measurement, IEEE Transactions on,* vol. 49, pp. 746-753, 2000.
[22] P. Dash, A. Pradhan, G. Panda, and R. Jena, "Harmonics estimation in stressed electric power networks," in *Power Electronics and Drive Systems, 2001. Proceedings., 2001 4th IEEE International Conference on*, 2001, pp. 782-788.
[23] R. Aghazadeh, H. Lesani, M. Sanaye-Pasand, and B. Ganji, "New technique for frequency and amplitude estimation of power system signals," *IEE Proceedings-Generation, Transmission and Distribution,* vol. 152, pp. 435-440, 2005.
[24] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering,* vol. 82, pp. 35-45, 1960.
[25] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, 2000, pp. 153-158.
[26] IEC, "IEC 61850: IEC Standard for Communication Network and Systems in Substations," in
ed: The International Electrotechnical Commission (IEC), 2003.
[27] T. Sidhu and P. K. Gangadharan, "Control and automation of power system substation using IEC61850 communication," in *Control Applications, 2005. CCA 2005. Proceedings of 2005 IEEE Conference on*, 2005, pp. 1331-1336.
[28] L. Hossenlopp, "Engineering perspectives on IEC 61850," *Power and Energy Magazine, IEEE,* vol. 5, pp. 45-50, 2007.
[29] I. T. Equipment, "Systems—Part 5-104: Transmission Protocols—Network Access for IEC 60870-5-101 Using Standard Transport Profiles," *IEC Standard,* vol. 60870, 2006.
[30] D. U. GROUP, "Distributed Network Protocol," ed.
[31] I. Modbus, "Modbus application protocol specification v1. 1a," *North Grafton, Massachusetts (www. modbus. org/specs. php),* 2004.
[32] K.-P. Brand, V. Lohmann, and W. Wimmer, *Substation Automation Handbook*: Utility Automation Consulting Lohmann Bremgarten, 2003.

[33] M. G. Kanabar and T. S. Sidhu, "Reliability and availability analysis of IEC 61850 based substation communication architectures," in *Power & Energy Society General Meeting, 2009. PES'09. IEEE*, 2009, pp. 1-8.
[34] T. Skeie, S. Johannessen, and C. Brunner, "Ethernet in substation automation," *Control Systems, IEEE*, vol. 22, pp. 43-51, 2002.
[35] R. Mackiewicz, "Overview of IEC 61850 and Benefits," in *Power Systems Conference and Exposition, 2006. PSCE'06. 2006 IEEE PES*, 2006, pp. 623-630.
[36] UCA, "“ Introduction to UCA_R version 2.0”, Tech. rep.," IEEE1999.
[37] C. Brunner, "IEC 61850 for power system communication," in *Transmission and Distribution Conference and Exposition, 2008. T&# x00026; D. IEEE/PES*, 2008, pp. 1-6.
[38] "IEC 61850-9-2 LE: Implementation Guideline for Digital Interface to Instrument Transformers Using IEC 61850 9-2," ed: UCA International Users Group.
[39] IEC, "International Standard IEC 61850-9-2, Communication networks and systems in substations: Specific Communication System Mapping (SCSM) – Sampled Values over ISO/IEC 8802-3," ed, 2003.
[40] IEEE, "IEEE 1588 Standard for a Precision Clock Synchronization. Protocol for Networked Measurement and Control Systems ", ed, 2002.
[41] I. E. Commission, "International Standard IEC 60044-8," *Instrument transformers–part*, vol. 8.
[42] IEEE, "IEEE Standard for Information technology--Telecommunications and information exchange between systems--Local and metropolitan area networks," vol. 802.3, ed: IEEE, 2008.
[43] K. Brand, C. Brunner, and W. Wimmer, "Design of IEC 61850 based substation automation systems according to customer requirements," *Indian Journal of Power and River Valley Development*, vol. 61, p. 87, 2011.
[44] U. Anombem, H. Li, P. Crossley, R. Zhang, and C. McTaggart, "Process bus architectures for substation automation with life cycle cost consideration," in *Developments in Power System Protection (DPSP 2010). Managing the Change, 10th IET International Conference on*, 2010, pp. 1-5.
[45] M. Adamiak, B. Kasztenny, J. Mazereeuw, D. Mcginn, and S. Hodder, "Considerations for IEC 61850 Process Bus Deployment in Real-world Protection and Control Systems: a business analysis," *Paper B5-102, CIGRE 42d session, Paris*, 2008.
[46] *Benefits of IEC61850 networking*. Available: http://www.sisconet.com/downloads/ Last update 10/08/2014
[47] S. Pascal, "Solution of IEC 61850 process bus," *PAC World*, vol. 3, pp. 36-41, 2009.
[48] R. Hunt, "Process Bus: A Practical Approach," *PAC World, Spring*, 2009.
[49] D. Tholomier and D. Chatrefou, "IEC 61850 Process Bus-It is Real!," *PAC World Magazine*, pp. 48-53, 2008.
[50] J.-C. Tan, V. Green, and J. Ciufo, "Testing IEC 61850 based multi-vendor substation automation systems for interoperability," in *Power Systems Conference and Exposition, 2009. PSCE'09. IEEE/PES*, 2009, pp. 1-5.

[51] J. Holbach, J. Rodriguez, C. Wester, D. Baigent, L. Frisk, S. Kunsman, and L. Hossenlopp, "Status on the first IEC61850 based protection and control, multi-vendor project in the United States," in *Protective Relay Engineers, 2007. 60th Annual Conference for*, 2007, pp. 283-306.
[52] T. S. Sidhu, G. Kanabar, and P. P. Parikh, "Implementation issues with IEC 61850 based substation automation systems," in *Fifteenth National Power Systems Conference, str*, 2008, pp. 473-478.
[53] L. Hossenlopp, D. Chatrefou, D. Tholomier, and D. Bui, "Process bus: Experience and impact on future system architectures," *CIGRE Session, Paris, France*, 2008.
[54] L. Andersson, K.-P. Brand, and D. Fuechsle, "Optimized architectures for process bus with IEC 61850-9-2," *CIGRE Session Paris, France*, 2008.
[55] C. Brunner, "IEC 61850 Process Connection—A Smart Solution to Connect the Primary Equipment to the Substation Automation System," in *15th Power Systems Computation Conference, Liege, Belgium*, 2005.
[56] B. Kasztenny, J. Whatley, A. U. Eric, J. Burger, D. Finney, and M. Adamiak, "Unanswered questions about IEC 61850—What needs to happen to realize the vision?," in *32nd Annual Western Protective Relay Conference, Spokane, WA*, 2005.
[57] "IEC Standard for communication networks and systems in substation, Part 5: Communication requirement for functions and device models, IEC 61850 part-5," ed, 2003.
[58] T. S. Sidhu and Y. Yin, "Modelling and simulation for performance evaluation of IEC61850-based substation communication systems," *Power Delivery, IEEE Transactions on*, vol. 22, pp. 1482-1489, 2007.
[59] D. P. Bertsekas, R. G. Gallager, and P. Humblet, *Data networks* vol. 2: Prentice-Hall International, 1992.
[60] M. S. Thomas and I. Ali, "Reliable, fast, and deterministic substation communication network architecture and its performance simulation," *Power Delivery, IEEE Transactions on*, vol. 25, pp. 2364-2370, 2010.
[61] R. Kuffel, D. Ouellette, and P. Forsyth, "Real time simulation and testing using IEC 61850," in *Modern Electric Power Systems (MEPS), 2010 Proceedings of the International Symposium*, 2010, pp. 1-8.
[62] M. G. Kanabar and T. S. Sidhu, "Performance of IEC 61850-9-2 process bus and corrective measure for digital relaying," *Power Delivery, IEEE Transactions on*, vol. 26, pp. 725-735, 2011.
[63] O. M. Documentation, "OPNET Technologies," *Inc.[Internet] http://www. opnet. com*, 2003.
[64] *RTDS Technologies: Real Time Power System Simulation*. Available: www.rtds.com/
[65] F. K. James and W. R. Keith, "Computer networking a top-down approach featuring the internet," ed: Addison-Wesley, Reading, 2004.
[66] J. F. Kurose, *Computer networking: a top-down approach featuring the Internet*: Pearson Education India, 2005.
[67] C. Spurgeon, *Ethernet: the definitive guide*: " O'Reilly Media, Inc.", 2000.

[68] M. Felser, "Real-Time Ethernet--Industry Prospective," *Proceedings of the IEEE,* vol. 93, pp. 1118-1129, 2005.
[69] J.-D. Decotignie, "Ethernet-based real-time and industrial communications," *Proceedings of the IEEE,* vol. 93, pp. 1102-1117, 2005.
[70] E. Demeter, S. Faried, and T. Sidhu, "Power system protective functions performance over an ethernet-based process bus," in *Electrical and Computer Engineering, 2007. CCECE 2007. Canadian Conference on*, 2007, pp. 264-267.
[71] E. Demeter, T. S. Sidhu, and S. O. Faried, "An open-system approach to power system protection and control integration," *Power Delivery, IEEE Transactions on,* vol. 21, pp. 30-37, 2006.
[72] D. C. Robinson, "Using Bus Level IRIG Time Code Translators to Time Tag Data and Synchronize Multiple Processing Stations," *Division of Datum Inc,* 1999.
[73] W. Stallings, *Local and metropolitan area networks*: Macmillan Publishing Co., Inc., 1993.
[74] U. Premaratne, J. Samarabandu, T. Sidhu, R. Beresh, and J.-C. Tan, "Security analysis and auditing of IEC61850-based automated substations," *Power Delivery, IEEE Transactions on,* vol. 25, pp. 2346-2355, 2010.
[75] U. K. Premaratne, J. Samarabandu, T. S. Sidhu, R. Beresh, and J.-C. Tan, "An intrusion detection system for IEC61850 automated substations," *Power Delivery, IEEE Transactions on,* vol. 25, pp. 2376-2383, 2010.
[76] F. Steinhauser, "New challenges with substations utilizing communication networks," in *Proceedings of the 2003 IEEE Power Tech Conference*, 2003.
[77] IEC, "IEC Standard for Electromagnetic compatibility (EMC), IEC 6100," ed, 2001.
[78] J. Banting, T. Beckwith, J. Burnworth, H. Calhoun, D. Dawson, C. Downs, A. Giuliante, W. Kotheimer, L. Scharf, and J. Tengdin, "Summary ofWithstand capability of relay systems to radiated electromagnetic interference from transceivers'(ANSI/IEEE C37. 90.2-1987)," *Power Delivery, IEEE Transactions on,* vol. 6, pp. 103-108, 1991.
[79] D. Hou and D. Dolezilek, "IEC 61850-What It Can and Cannot Offer to Traditional Protection Shemes," *Schweitzer Engineering Laboratories, Inc,* vol. 20080912, 2008.
[80] *IRIG-B*. Available: http://www.irigb.com/
[81] E. Gressier, "A stochastic Petri net model for Ethernet," in *International Workshop on Timed Petri Nets*, 1985, pp. 296-303.
[82] M. P. Pozzuoli, "Ethernet in substation automation applications–issues and requirements," *Rugged-Com Inc., Woodbridge, ON, Canada, Tech. Rep,* 2004.
[83] I. S. T. C. Formats-B, "Range Commanders Council," *US Army White Sands Missile Range, New Mexico,* pp. 88002-5110, 2004.
[84] H. Farhangi, "The path of the smart grid," *Power and Energy Magazine, IEEE,* vol. 8, pp. 18-28, 2010.
[85] N. Framework, "Roadmap for Smart Grid Interoperability Standards," *NIST special publication,* vol. 1108, 2010.

[86] M. Janssen and A. Apostolov, "IEC 61850 impact on substation design," in *Transmission and Distribution Conference and Exposition, 2008. T&# x00026; D. IEEE/PES*, 2008, pp. 1-7.
[87] D. Steedman, *Abstract syntax notation one (ASN. 1): the tutorial and reference*: Technology appraisals, 1993.
[88] R. Seifert and J. Edwards, *The All-New Switch Book: The Complete Guide to LAN Switching Technology*: John Wiley & Sons, 2008.
[89] J. Postel, "User datagram protocol," *Isi*, 1980.
[90] K. C. Almeroth. Available: http://www.cs.ucsb.edu/~almeroth/classes/W01.176B/hw2/
[91] P. D. Buchan. (2013). Available: http://www.pdbuchan.com/rawsock/receive_arp.c
[92] D. C. Plummer, "An Ethernet Address Resolution Protocol—or—Converting Network Protocol Addresses to 48. bit Ethernet Address for Transmission on Ethernet Hardware, Nov. 1982,[online],[retrieved on Jan. 17, 2000]," *Network Working Group, Request for Comments*, vol. 826, 2003.
[93] C. C. Set, "7-bit American Standard Code for Information Interchange (ASCII), ANSI X3. 4-1986," *This standard is the specification of the US-ASCII charset*.
[94] "Wireshark," *Web page: http://www.wireshark.org*, 2014.
[95] IEEE, "IEEE Std C37.118™- Standard for Synchrophasor for Power Systems," ed, 2005.
[96] PSRC. Available: http://www.pes-psrc.org/
[97] A. Monticelli, "Electric power system state estimation," *Proceedings of the IEEE*, vol. 88, pp. 262-282, 2000.
[98] A. Abur and A. G. Exposito, *Power system state estimation: theory and implementation* vol. 24: CRC, 2004.
[99] C. P. Steinmetz, *Theory and calculation of alternating current phenomena* vol. 4: McGraw-Hill Book Company, inc., 1916.
[100] R. Burnett Jr, M. Butts, T. Cease, V. Centeno, G. Michel, R. Murphy, and A. Phadke, "Synchronized phasor measurements of a power system event," *Power Systems, IEEE Transactions on*, vol. 9, pp. 1643-1650, 1994.
[101] A. Phadke, B. Pickett, M. Adamiak, M. Begovic, G. Benmouyal, R. Burnett Jr, T. Cease, J. Goossens, D. Hansen, and M. Kezunovic, "Synchronized sampling and phasor measurements for relaying and control," *Power Delivery, IEEE Transactions on*, vol. 9, pp. 442-452, 1994.
[102] IEEE, "IEEE C37.118 Standard for Synchrophasor for Power Systems," ed, 2005.
[103] M. Gavrilas, "Recent advances and applications of synchronized phasor measurements in power systems," in *proc. 9th WSEAS/IASME International Conference on Electric Power Systems, High Voltages, Electric Machines, Budapest, Hungary*, 2009.
[104] S. Chen and R. Živanović, "Estimation of frequency components in stator current for the detection of broken rotor bars in induction machines," *Measurement*, vol. 43, pp. 887-900, 2010.

[105] R. Zivanovic and C. Cairns, "Implementation of PMU technology in state estimation: an overview," in *AFRICON, 1996., IEEE AFRICON 4th*, 1996, pp. 1006-1011.
[106] A. Abdolkhalig and R. Zivanovic, "Towards dynamic on-line state estimator based on IEC61850–9–2 Process Bus," in *Electrotechnical Conference (MELECON), 2012 16th IEEE Mediterranean*, 2012, pp. 261-264.
[107] R. Van Der Merwe and E. A. Wan, "The square-root unscented Kalman filter for state and parameter-estimation," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, 2001, pp. 3461-3464.
[108] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE,* vol. 92, pp. 401-422, 2004.
[109] J. Depablos, V. Centeno, A. G. Phadke, and M. Ingram, "Comparative testing of synchronized phasor measurement units," in *Power Engineering Society General Meeting, 2004. IEEE*, 2004, pp. 948-954.
[110] K. E. Martin, A. J. Faris, and J. F. Hauer, "Standardized testing of phasor measurement units," Pacific Northwest National Laboratory (PNNL), Richland, WA (US)2006.
[111] P. Komarnicki, C. Dzienis, Z. Styczynski, J. Blumschein, and V. Centeno, "Practical experience with PMU system testing and calibration requirements," in *Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*, 2008, pp. 1-5.
[112] J. Ren, M. Kezunovic, and G. Stenbakken, "Dynamic characterization of PMUs using step signals," in *Power & Energy Society General Meeting, 2009. PES'09. IEEE*, 2009, pp. 1-6.
[113] J. Ren, M. Kezunovic, and G. Stenbakken, "Characterizing dynamic behavior of PMUs using step signals," *European Transactions on Electrical Power,* vol. 21, pp. 1496-1508, 2011.
[114] M. Balabin, K. Görner, Y. Li, I. Naumkin, and C. Rehtanz, "Evaluation of PMU performance during transients," in *Power System Technology (POWERCON), 2010 International Conference on*, 2010, pp. 1-8.
[115] A. H. Jazwinski, "Stochastic Processes and Filtering Theory. 1970," ed: Academic Press, New York.
[116] P. Dienes, "The Taylor Series," 1957.
[117] E. Styvaktakis, I. Y. H. Gu, and M. H. J. Bollen, "Voltage dip detection and power system transients," in *Power Engineering Society Summer Meeting, 2001*, 2001, pp. 683-688.
[118] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *AeroSense'97*, 1997, pp. 182-193.
[119] H. Novanda, P. Regulski, F. M. González-Longatt, and V. Terzija, "Unscented Kalman Filter for frequency and amplitude estimation," in *PowerTech, 2011 IEEE Trondheim*, 2011, pp. 1-6.

[120] W. Duesterhoeft, M. W. Schulz, and E. Clarke, "Determination of instantaneous currents and voltages by means of alpha, beta, and zero components," *American Institute of Electrical Engineers, Transactions of the,* vol. 70, pp. 1248-1255, 1951.
[121] A. A. Girgis, W. B. Chang, and E. B. Makram, "A digital recursive measurement scheme for online tracking of power system harmonics," *Power Delivery, IEEE Transactions on,* vol. 6, pp. 1153-1160, 1991.
[122] R. G. Brown, *Introduction to random signal analysis and Kalman filtering* vol. 8: Wiley New York, 1983.
[123] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*: John Wiley & Sons, 2006.
[124] F.-L. Lian, J. R. Moyne, and D. M. Tilbury, "Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet," *Control Systems, IEEE,* vol. 21, pp. 66-83, 2001.
[125] D. J. Cimino and E. Z. DeSouza, "Media access controller," ed: Google Patents, 1995.
[126] D. Hristu-Varsakelis and W. S. Levine, *Handbook of networked and embedded control systems*: Springer, 2005.
[127] M. Andersson, D. Henriksson, A. Cervin, and K. Arzen, "Simulation of wireless networked control systems," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, 2005, pp. 476-481.
[128] L. R. Rabiner and B. Gold, "Theory and application of digital signal processing," *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975. 777 p.,* vol. 1, 1975.
[129] Z. Huang, B. Kasztenny, V. Madani, K. Martin, S. Meliopoulos, D. Novosel, and J. Stenbakken, "Performance evaluation of phasor measurement systems," in *Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*, 2008, pp. 1-7.
[130] P. M. Kanabar, M. G. Kanabar, W. El-Khattam, T. S. Sidhu, and A. Shami, "Evaluation of communication technologies for IEC 61850 based distribution automation system with distributed energy resources," in *Power & Energy Society General Meeting, 2009. PES'09. IEEE*, 2009, pp. 1-8.
[131] Y. G. Paithankar and S. Bhide, *Fundamentals of power system protection*: PHI Learning Pvt. Ltd., 2010.
[132] S. H. Horowitz and A. G. Phadke, *Power system relaying*: John Wiley & Sons, 2013.
[133] A. M. El-Hadidy and A. A. Mahfouz, "Impact of decaying DC component on the characteristics of over-current protective relays," 2009.
[134] P. M. Donohue and S. Islam, "The Effect of Nonsinusoidal Current Waveforms on Electromechanical and Solid-State Overcurrent Relay Operation," *Industry Applications, IEEE Transactions on,* vol. 46, pp. 2127-2133, 2010.
[135] S. Suresh Kumar, V. Subbiah, A. Kandaswaray, G. Dinesh Kumar, R. Sujay, and S. Manoharan, "A state of the art STATCON for instantaneous VAr compensation and harmonic suppression to enhance power quality," in *Quality and Security of Electric Power Delivery Systems, 2003. CIGRE/PES 2003. CIGRE/IEEE PES International Symposium*, 2003, pp. 86-90.

[136] IEC, "IEC 60255 Standard for Electrical Relays," ed: The International Electrotechnical Commission, 1989.
[137] V. M. M. Saiz and J. B. Guadalupe, "Application of Kalman filtering for continuous real-time tracking of power system harmonics," in *Generation, Transmission and Distribution, IEE Proceedings-*, 1997, pp. 13-20.
[138] H. W. Sorenson, *Kalman filtering: theory and application* vol. 38: IEEE press New York, 1985.
[139] A. Abdolkhalig and R. Zivanovic, "Performance Evaluation of Phasor Estimator within IEC 61850-9-2 Communication Network," in *The International Conference on Electrical and Electronics Engineering, Clean Energy and Green Computing (EEECEGC 2013)*, 2013, pp. 113-119.
[140] A. Abdolkhalig and R. Zivanovic, "Phasor measurement based on IEC 61850-9-2 and Kalman–Filtering," *Measurement*, vol. 50, pp. 126-134, 2014.

Appendix A: SMV traffic writer

```
// This code is part one of two codes for writting and reading of SMV frames.
// This part is to write (send out) SMV packet continuously through the
Ethernet port.
// This code is based on the code example on:
// "P. D. Buchan. (2013). Available: http://www.pdbuchan.com/rawsock/"

//------------------------------------------------------------------------

// Packet sockets are used to receive or send raw packets at the device driver
(OSI7 Layer 2) level
#include <sys/socket.h>   // To define the Internet Protocol family.
#include <netpacket/packet.h>
#include <net/ethernet.h> // The Layer 2 protocols.
//------------------------------------------------------------------------

#include <stdio.h>        // C library to perform Input/Output operations.
#include <netdb.h>        // Definitions for network database operations.
#include <arpa/inet.h>    // Definitions for internet operations.
#include <netinet/ip.h>   // Definitions for internet protocol.
#include <string.h>       // Definitions for several functions to manipulate C
strings and arrays.
#include <time.h>         // Definitions for functions to get and manipulate
date and time information.
#include <stdlib.h>       // For exit.
#include <ifaddrs.h>      // For getting the interface addresses.
#include <sys/ioctl.h>    // For net device.
#include <sys/time.h>     // For time.
#include <math.h>         // For sine wave.
#include <netinet/in.h>   // To define Internet address family.
#include <sys/types.h>    // To define data types.
#include <net/if.h>       // To define sockets local interfaces.
#include <endian.h>       // Generic definitions for little- and big-endian
systems.
# define phi 3.1415926/3
//------------------------------------------------------------------------

void ErrorHandling(char *e){
       perror(e);
       exit(1);
}

static int SmpCount= 0x0000;

int print_pkt(char *buf, int len){
       int j = 0;
       for(j = 0; j < len; j++ )
       {
              if((j%16) == 0 && j != 0 )
              printf("\n");
              printf("%02x ",*(buf+j)& 0xFF );
       }
       printf("\n====Packet END====\n");
       return 0;
}

// Starting the program
int main(int argc, char *argv[]){
```

```
	struct sockaddr_ll dsti;
	struct sockaddr_in *hostAddr, TrgtIP, SndIP;
	int packet_socket,sock,i=0;
	int8_t *buffer;
	int32_t l;
	struct ifreq ifr;

	// Define the 4th layer
	// One Dataset "SMV_dataset"
	struct SMV_dataset {
	// Instantaneous magnitude of Currents (Phase A, B, C and neutral) in
Ampere
		int32_t InnATCTR1_Amp_instMag_i;//Magnitude
		int32_t InnATCTR1_Amp_q;        //Quality
		int32_t InnBTCTR2_Amp_instMag_i;//Magnitude
		int32_t InnBTCTR2_Amp_q;        //Quality
		int32_t InnCTCTR3_Amp_instMag_i;//Magnitude
		int32_t InnCTCTR3_Amp_q;        //Quality
		int32_t InnNTCTR4_Amp_instMag_i;//Magnitude
		int32_t InnNTCTR4_Amp_q;
	// Instantaneous magnitude of Voltages (Phase A, B, C and neutral) in
Volt
		int32_t UnnATVTR1_Vol_instMag_i;//Magnitude
		int32_t UnnATVTR1_Vol_q;        //Quality
		int32_t UnnBTVTR2_Vol_instMag_i;//Magnitude
		int32_t UnnBTVTR2_Vol_q;        //Quality
		int32_t UnnCTVTR3_Vol_instMag_i;//Magnitude
		int32_t UnnCTVTR3_Vol_q;        //Quality
		int32_t UnnNTVTR4_Vol_instMag_i;//Magnitude
		int32_t UnnNTVTR4_Vol_q;        //Quality
	}__attribute__((__packed__));

	// Define the 3rd layer
	// Application Service Data Unit (ASDU)
	// ASDU is coded according to 9-2 LE and ASN.1 encoding rules
	struct SMV_ASDU {
		int8_t SequenceASDU_tag;     //0x30
		int8_t SequenceSMV_ASDUngth; //0x01
		int8_t svID_tag;             //0x80
		int8_t svID_Length;          //0x0A
		int8_t svID[11];             //according to the naming rules is
in the IEC 61869-9
		int8_t smpCnt_tag;           //0x82
		int8_t smpCnt_Length;        //0x02
		int16_t smpCnt;              //Samples Counter
		int8_t confRev_tag;          //0x83
		int8_t confRev_Length;       //0x04
		int32_t confRev;             //Configuration revision number
		int8_t smpSynch_tag;         //0x85
		int8_t smpSynch_Length;      //0x01
		int8_t smpSynch;             //synchronization, 2-> global 1->
local 0-> not sych
		int8_t smpRate_tag;          //0x86
		int8_t smpRate_Length;       //0x02
		int16_t smpRate;             //80 samples per cycle (0x0050 in
hex)
		int8_t SequenceofData_tag;   //0x87
		int8_t SequenceofData_Length;//0x40
		struct SMV_dataset DataSet;
	}__attribute__((__packed__));
```

```
	// Define the 2nd layer
	// Application Protocol Data Unit  (APDU)
	struct SMV_APDU {
		int8_t savPDUtag;
		int8_t savPDULength;  //sizeof SMV_APDU
		int8_t noASDUtag;
		int8_t noSMVASDUngth; //sizeof noASDU
		int8_t noASDU;        //One ASDU (0x01 hex)
		int8_t SequenceofASDUtag;
		int8_t SequenceofSMVASDUngth;//sizeof all ASDU
		struct SMV_ASDU SMV_ASDU1;
	}__attribute__((__packed__));

	// Sampled Measrued Value frame structure
	struct SMV_FRM {
		//Header MAC
		int8_t destination_mac[6];
		int8_t source_mac[6];
		//Start Priority tagged
		int16_t TPID;     // For 802.1Q VLAN, its 0x8100
		int16_t TCI;      // 0x4000
		// Start Ethertype PDU
		int16_t Ethertype;// For SMV, its 0x88BA
		int16_t APPID;    // 0x4000 application identifier
		int16_t Length;   // After APPID
		int16_t reserved1;
		int16_t reserved2;
		//Adding the SMV_APDU (the application protocol data unit)
		struct SMV_APDU SMV_apdu;
		//pad bytes if necessary
		//Frame check sequence
	}__attribute__((__packed__));

	struct SMV_FRM FRM;
	memset(&FRM, 0, sizeof FRM);

	printf("====Initializing parameters succeessed====\n");

	// Check input argument
	if (argc!=2)
		ErrorHandling("To run: ./SMV_s + <host interface name>\n");

	// Open a socket for ioctl
	sock = socket(AF_INET, SOCK_STREAM, 0);
	if(sock == -1)
		ErrorHandling("open socket for interface");
	memcpy(ifr.ifr_name, argv[1], IF_NAMESIZE);

	// Get MAC address of the interface, after call this hostAssr will be
reset
	if ((ioctl(sock, SIOCGIFHWADDR, &ifr, sizeof ifr)) == -1)
		ErrorHandling("HW addr");
	// Display
	printf("Sender MAC address: %02x:%02x:%02x:%02x:%02x:%02x \n",

	ifr.ifr_hwaddr.sa_data[0]&0xFF,

	ifr.ifr_hwaddr.sa_data[1]&0xFF,

	ifr.ifr_hwaddr.sa_data[2]&0xFF,

	ifr.ifr_hwaddr.sa_data[3]&0xFF,
```

```
        ifr.ifr_hwaddr.sa_data[4]&0xFF,

        ifr.ifr_hwaddr.sa_data[5]&0xFF);

        printf("Formating SMV packet...\n");
        //Formatting the SMV packet
        //The MAC Header
        //The destination MAC Address, should be in the range of
01:0C:CD:04:XX:XX
        memset(FRM.destination_mac, 0xBC, sizeof FRM.destination_mac[0]);
        memset(FRM.destination_mac+1, 0x30, sizeof FRM.destination_mac[0]);
        memset(FRM.destination_mac+2, 0x5B, sizeof FRM.destination_mac[0]);
        memset(FRM.destination_mac+3, 0xA1, sizeof FRM.destination_mac[0]);
        memset(FRM.destination_mac+4, 0x7A, sizeof FRM.destination_mac[0]);
        memset(FRM.destination_mac+5, 0x1C, sizeof FRM.destination_mac[0]);
    //The source MAC Address
        memset(FRM.source_mac, (ifr.ifr_hwaddr.sa_data[0]&0xFF), sizeof
FRM.source_mac[0]);
        memset(FRM.source_mac+1, (ifr.ifr_hwaddr.sa_data[1]&0xFF), sizeof
FRM.source_mac[0]);
        memset(FRM.source_mac+2, (ifr.ifr_hwaddr.sa_data[2]&0xFF), sizeof
FRM.source_mac[0]);
        memset(FRM.source_mac+3, (ifr.ifr_hwaddr.sa_data[3]&0xFF), sizeof
FRM.source_mac[0]);
        memset(FRM.source_mac+4, (ifr.ifr_hwaddr.sa_data[4]&0xFF), sizeof
FRM.source_mac[0]);
        memset(FRM.source_mac+5, (ifr.ifr_hwaddr.sa_data[5]&0xFF), sizeof
FRM.source_mac[0]);

        FRM.TPID = htons(0x8100);      //Priority tag
        FRM.TCI = htons(0x4000);
        FRM.Ethertype = htons(0x88BA); //Ethertype PDU
        FRM.APPID = htons(0x4000);
        FRM.Length = htons(0x006D);
        FRM.reserved1 = htons(0x0000); //htons(0x8000)
        FRM.reserved2 = htons(0x0000);

        // Define the SMV_APDU
        FRM.SMV_APDU.savPDUtag = 0x60;
        FRM.SMV_APDU.savPDULength = 0x63;
        FRM.SMV_APDU.noASDUtag = 0x80;
        FRM.SMV_APDU.noSMVASDUngth = 0x01;
        FRM.SMV_APDU.noASDU = 0x01;
        FRM.SMV_APDU.SequenceofASDUtag = 0xA2;
        FRM.SMV_APDU.SequenceofSMVASDUngth = 0x5E;
        // Define the SMV_ASDU1
        FRM.SMV_APDU.SMV_ASDU1.SequenceASDU_tag = 0x30;
        FRM.SMV_APDU.SMV_ASDU1.SequenceSMV_ASDUngth = 0x5C;
        FRM.SMV_APDU.SMV_ASDU1.svID_tag = 0x80;
        FRM.SMV_APDU.SMV_ASDU1.svID_Length = 0x0B;
        FRM.SMV_APDU.SMV_ASDU1.SequenceASDU_tag = 0x30;
        FRM.SMV_APDU.SMV_ASDU1.SequenceSMV_ASDUngth = 0x5C;
        FRM.SMV_APDU.SMV_ASDU1.svID_tag = 0x80;
        FRM.SMV_APDU.SMV_ASDU1.svID_Length = 0x0B;
        memset(FRM.SMV_APDU.SMV_ASDU1.svID, 0x5F, sizeof
FRM.SMV_APDU.SMV_ASDU1.svID[0]);   // _
        memset(FRM.SMV_APDU.SMV_ASDU1.svID+1, 0x41, sizeof
FRM.SMV_APDU.SMV_ASDU1.svID[0]); // A
        memset(FRM.SMV_APDU.SMV_ASDU1.svID+2, 0x48, sizeof
FRM.SMV_APDU.SMV_ASDU1.svID[0]); // H
```

```
	memset(FRM.SMV_APDU.SMV_ASDU1.svID+3, 0x4D, sizeof
FRM.SMV_APDU.SMV_ASDU1.svID[0]); // M
	memset(FRM.SMV_APDU.SMV_ASDU1.svID+4, 0x45, sizeof
FRM.SMV_APDU.SMV_ASDU1.svID[0]); // E
	memset(FRM.SMV_APDU.SMV_ASDU1.svID+5, 0x44, sizeof
FRM.SMV_APDU.SMV_ASDU1.svID[0]); // D
	memset(FRM.SMV_APDU.SMV_ASDU1.svID+6, 0x5F, sizeof
FRM.SMV_APDU.SMV_ASDU1.svID[0]); // _
	memset(FRM.SMV_APDU.SMV_ASDU1.svID+7, 0x53, sizeof
FRM.SMV_APDU.SMV_ASDU1.svID[0]); // S
	memset(FRM.SMV_APDU.SMV_ASDU1.svID+8, 0x56, sizeof
FRM.SMV_APDU.SMV_ASDU1.svID[0]); // V
	memset(FRM.SMV_APDU.SMV_ASDU1.svID+9, 0x31, sizeof
FRM.SMV_APDU.SMV_ASDU1.svID[0]); // 1
	memset(FRM.SMV_APDU.SMV_ASDU1.svID+10, 0x5F, sizeof
FRM.SMV_APDU.SMV_ASDU1.svID[0]);// _
	FRM.SMV_APDU.SMV_ASDU1.smpCnt_tag = 0x82;
	FRM.SMV_APDU.SMV_ASDU1.smpCnt_Length = 0x02;
	FRM.SMV_APDU.SMV_ASDU1.confRev_tag = 0x83;
	FRM.SMV_APDU.SMV_ASDU1.confRev_Length = 0x04;
	FRM.SMV_APDU.SMV_ASDU1.confRev = htonl(0x00000001);
	FRM.SMV_APDU.SMV_ASDU1.smpSynch_tag = 0x85;
	FRM.SMV_APDU.SMV_ASDU1.smpSynch_Length = 0x01;
	FRM.SMV_APDU.SMV_ASDU1.smpSynch = 0x00; // 0->none; 1->local; 2->global
	FRM.SMV_APDU.SMV_ASDU1.SequenceofData_tag = 0x87;
	FRM.SMV_APDU.SMV_ASDU1.SequenceofData_Length = 0x40;
	// End of formatting

	printf("Formating successed...\n");
	 printf("SMV packet is...\n");
	 printf("\n");
	print_pkt((char *)&FRM, sizeof FRM);

	// To get the interface index
	if ((ioctl(sock, SIOCGIFINDEX, &ifr, sizeof ifr)) == -1)
		ErrorHandling("index");

	// To set the destination in sockaddr_ll
	memset(&dsti, 0, sizeof dsti);
	dsti.sll_family=AF_PACKET;
	dsti.sll_protocol=htons(ETH_P_8021Q); //it also works with ETH_P_ARP
	dsti.sll_ifindex=ifr.ifr_ifindex;

	// To open the  layer 2 socket
	packet_socket = socket(AF_PACKET, SOCK_RAW, htons(ETH_P_8021Q));
	if(packet_socket == -1)
		ErrorHandling("open layer 2 socket");
	else printf("open layer 2 socket succeed\nsending SMV packets...\n");

	// To send the ARP to dsti
	int cnt=0;
	 double Harmonics_A=0,Harmonics_B=0,Harmonics_C=0;

	// To send the SMV frames continuously
	while(1){
	struct timeval tv1,tv2;
	struct timezone tz1,tz2;
	int def=0,slp=0,cmpst=0;
	int Ia,Ib,Ic,Va,Vb,Vc;
	double CurrentA,VoltageA,CurrentB,VoltageB,CurrentC,VoltageC;
```

```
            for (i=0;i<=3999;i++)

               {
                   if (SmpCount==0x0FA0)
                      {//0x3200
                          SmpCount=0x0000;
                          cnt++;
                        }


            FRM.SMV_APDU.SMV_ASDU1.smpCnt = htons(SmpCount);
                   SmpCount++;
                   if (cnt==0)

                  {//fault current will appear after 15 seconds

Harmonics_A=sin(2*100*3.1415926*i*1/4000+phi/3)+sin(5*100*3.1415926*i*1/4000+ph
i/4);
        Harmonics_B=sin(2*100*3.1415926*i*1/4000-3.1415926*2/3+phi/3)+
sin(5*100*3.1415926*i*1/4000-3.1415926*2/3+phi/4);
        Harmonics_C=sin(2*100*3.1415926*i*1/4000+3.1415926*2/3+phi/3)+
sin(5*100*3.1415926*i*1/4000+3.1415926*2/3+phi/4);
                   }


                   if (cnt==1)
                      {//fault current will be removed after 20 seconds
                                Harmonics_A=0;Harmonics_B=0;Harmonics_C=0;
                               }


            // The sinusoidal values
        CurrentA = (sin(1*100*3.1415926*i*1/4000+phi/2)+Harmonics_A)*1000;//
The current scale factor for SMV is 0.001*/
        CurrentB = (sin(1*100*3.1415926*i*1/4000-3.1415926*2/3+phi/2)+
Harmonics_B)*1000;
        CurrentC = (sin(1*100*3.1415926*i*1/4000+3.1415926*2/3+phi/2)+
Harmonics_C)*1000;

        VoltageA = 33000*sin(100*3.1415926*i*1/4000)*100;                  //
The voltage scale factor for SMV is 0.01*/
        VoltageB = 33000*sin(100*3.1415926*i*1/4000+3.1415926*2/3)*100;
           VoltageC = 33000*sin(100*3.1415926*i*1/4000-3.1415926*2/3)*100;

        // To assign values into the SMV_dataset*/
           FRM.SMV_APDU.SMV_ASDU1.DataSet.InnATCTR1_Amp_instMag_i =
htonl(CurrentA); // Phase A
            FRM.SMV_APDU.SMV_ASDU1.DataSet.InnATCTR1_Amp_q =
htonl(0x00000000);
            FRM.SMV_APDU.SMV_ASDU1.DataSet.InnBTCTR2_Amp_instMag_i =
htonl(CurrentB); // Phase B
            FRM.SMV_APDU.SMV_ASDU1.DataSet.InnBTCTR2_Amp_q =
htonl(0x00000000);
            FRM.SMV_APDU.SMV_ASDU1.DataSet.InnCTCTR3_Amp_instMag_i =
htonl(CurrentC); // Phase C
            FRM.SMV_APDU.SMV_ASDU1.DataSet.InnCTCTR3_Amp_q =
htonl(0x00000000);
            FRM.SMV_APDU.SMV_ASDU1.DataSet.InnNTCTR4_Amp_instMag_i =
htonl(CurrentA+CurrentB+CurrentC);//neutral
            FRM.SMV_APDU.SMV_ASDU1.DataSet.InnNTCTR4_Amp_q =
htonl(0x00002000);
```

```
            FRM.SMV_APDU.SMV_ASDU1.DataSet.UnnATVTR1_Vol_instMag_i =
htonl(VoltageA); // Phase A
            FRM.SMV_APDU.SMV_ASDU1.DataSet.UnnATVTR1_Vol_q =
htonl(0x00000000);
            FRM.SMV_APDU.SMV_ASDU1.DataSet.UnnBTVTR2_Vol_instMag_i =
htonl(VoltageB); // Phase B
            FRM.SMV_APDU.SMV_ASDU1.DataSet.UnnBTVTR2_Vol_q =
htonl(0x00000000);
            FRM.SMV_APDU.SMV_ASDU1.DataSet.UnnCTVTR3_Vol_instMag_i =
htonl(VoltageC); // Phase C
            FRM.SMV_APDU.SMV_ASDU1.DataSet.UnnCTVTR3_Vol_q =
htonl(0x00000000);
            FRM.SMV_APDU.SMV_ASDU1.DataSet.UnnNTVTR4_Vol_instMag_i =
htonl(VoltageA+VoltageB+VoltageC);//neutral
            FRM.SMV_APDU.SMV_ASDU1.DataSet.UnnNTVTR4_Vol_q =
htonl(0x00002000);

          if(i == 0)

          {// For getting the computer locol time when sending of SMV starts
                    gettimeofday(&tv1, &tz1);
                }

                if(sendto(packet_socket, &FRM, sizeof FRM, 0, (struct
sockaddr *)&dsti, sizeof dsti)==-1)
                    ErrorHandling("sendto()");


                gettimeofday(&tv2, &tz2);// To get the time after each
sent out of SMV

                // For calculating the witing time
                def=1000000/4000*i-((tv2.tv_sec-
tv1.tv_sec)*1000000+(tv2.tv_usec-tv1.tv_usec))+cmpst;

                   if(def<0)
                   {
                       slp = 0;
                       cmpst = def;
                            }

                   else

                     {
                       slp = def;
                       cmpst = 0;
                         }
                usleep(slp); // TO suspend the thread for ensuring the
next SMV is sent out after 250us from the previous
            }
      }


       printf("====Packet sent====\n");
      print_pkt((char *)&FRM, sizeof FRM);
      // To close the sockets
      close(packet_socket);
      close(sock);
}
```

Appendix B: SMV traffic read

```
// This code is part two of two codes for writting and reading of SMV frames.
// This code is based on the code example on:
// "P. D. Buchan. (2013). Available: http://www.pdbuchan.com/rawsock/"

// This part is to read (receive) SMV packet continuously through the Ethernet port
// and unwrapping the data included. In addition, the code contains subroutine
// which is responsible to estimate the phasors based on UKF with a reporting rate
// of 100 frames per second as defined in IEEE C37.118
//------------------------------------------------------------------------

// Packet sockets are used to receive or send raw packets at the device driver (OSI7
Layer 2) level
#include <sys/socket.h>   // To define the Internet Protocol family.
#include <netpacket/packet.h>
#include <net/ethernet.h> // The Layer 2 protocols.
//------------------------------------------------------------------------

#include <stdio.h>          // Standard Library of Input and Output
#include <complex.h>        // Standart Library of Complex Numbers
#include <math.h>           // For sine wave and other mathematical functions
#include <stddef.h>         // Definitions for several types implicitly generated
#include <limits.h>         // Defines constants with the limits of fundamental
integral types
#include <sys/types.h>      // Definitions for data types
#include <netdb.h>          // Definitions for network database operations
#include <netinet/ip.h>     // Definitions for internet protocol...IP_MAXPACKET
(65535)
#include <arpa/inet.h>      // Definitions for internet operations.
#include <endian.h>         // Definitions for little- and big-endian systems.
#include <float.h>          // Describes the characteristics of floating types for
the specific system
#include <sys/types.h>      // Needed for socket()
#include <sys/socket.h>     // Needed for socket()
#include <stdlib.h>         // For exit*/
#include <ifaddrs.h>        // For getting the interface addresses.
#include <string.h>         // For strcpy, memset(), and memcpy()
#include <unistd.h>         // For close()
#include <sys/ioctl.h>      // For netdevice
#include <net/if.h>         // For sockets local interfaces
#include <time.h>           // For time
#include <sys/time.h>       // For time
#include <linux/if_ether.h> // ETH_P_ARP = 0x0806, ETH_P_ALL = 0x0003
#include <errno.h>          // errno, perror()
#include <netinet/in.h>     // Internet address family

# define TRUE (1U)
# define FALSE (0U)
# define C double
# define phi  3.1415926/3
# define PI  3.1415926


//------------------------------------------------------------------------------
//Start of subroutine which is responsible to estimate the phasors based on UKF

void b_sum(const double b_x[40],double y[2])
{
  int iy;
```

```
  int ixstart;
  int j;
  int ix;
  double s;
  int k;
  iy = -1;
  ixstart = -1;
  for (j = 0; j < 2; j++) {
    ixstart++;
    ix = ixstart;
    s = b_x[ixstart];
    for (k = 0; k < 19; k++) {
      ix += 2;
      s += b_x[ix];
    }

    iy++;
    y[iy] = s;
  }
}

void sum(const double b_x[200], double y[10])
{
  int iy,ixstart,j,ix,k;
  double s;

  iy = -1;
  ixstart = -1;
  for (j = 0; j < 10; j++) {
    ixstart++;
    ix = ixstart;
    s = b_x[ixstart];
    for (k = 0; k < 19; k++) {
      ix += 10;
      s += b_x[ix];
    }

    iy++;
    y[iy] = s;
  }
}


void chol(double A[100])
{
  int jmax,colj,j,jj,ix,iy,coljp1,i1,iac,i2,ia;
  unsigned char exitg1;
  double ajj,c;

  jmax = -1;
  colj = 0;
  j = 0;
  exitg1 = 0U;
  while ((exitg1 == 0U) && (j + 1 < 11)) {
    jj = colj + j;
    ajj = 0.0;
    if (j < 1) {
    } else {
      ix = colj;
      iy = colj;
      for (coljp1 = 1; coljp1 <= j; coljp1++) {
        ajj += A[ix] * A[iy];
```

```
        ix++;
        iy++;
      }
    }

    ajj = A[jj] - ajj;
    if (ajj > 0.0) {
      ajj = sqrt(ajj);
      A[jj] = ajj;
      if (j + 1 < 10) {
        jj += 10;
        coljp1 = colj + 11;
        if (j == 0) {
        } else {
          iy = jj;
          i1 = coljp1 + 10 * (8 - j);
          for (iac = coljp1; iac <= i1; iac += 10) {
            ix = colj + 1;
            c = 0.0;
            i2 = (iac + j) - 1;
            for (ia = iac; ia <= i2; ia++) {
              c += A[ia - 1] * A[ix - 1];
              ix++;
            }

            A[iy] += -c;
            iy += 10;
          }
        }

        ajj = 1.0 / ajj;
        i1 = (jj + 10 * (8 - j)) + 1;
        while (jj + 1 <= i1) {
          A[jj] *= ajj;
          jj += 10;
        }

        colj = coljp1 - 1;
      }

      j++;
    } else {
      A[jj] = ajj;
      jmax = j;
      exitg1 = 1U;
    }
  }

  if (jmax + 1 == 0) {
    jmax = 10;
  }

  for (j = 0; j + 1 <= jmax; j++) {
    for (jj = j + 1; jj + 1 <= jmax; jj++) {
      A[jj + 10 * j] = 0.0;
    }
  }
}
// End of chol() function
```

```
// Start of UKF Routine

struct PHASOR
{
    double AMP1,PHS1;
   };

struct PHASOR UKF(int ct,double VALa, double VALb, double VALc)
{
  int i0,r2,ia,ib,k;
  double a21,a22;
  double Amp0,Amp2,Amp5,ang0,ang2,ang5,aang0,aang2,aang5,Phs0,Phs2,Phs5;
  double complex V1_,V2_,V3_,V4_,V5_,V6_,Ang0,Ang2,Ang5;
  double xo[10];
  double Pm[100];
  double A[100];
  double b[200];
  double b_A[200];
  double X[200];
  double xm[10];
  double a[10];
  double mx[10];
  double z[2];
  double Pz[4];
  double my[2];
  double b_b[2];
  double Pxz[20];
  double c_A[4];
  double B[20];
  double Y[20];
  double b_VALa[3];
  double dv0[100];


  static  double x[10] = {0,0,0,0,0,0,0,0,0,0};
  static const double d_A[6] = { 0.66666666666666663,         0.0         , -
0.33333333333333331,
                                 0.57735026918962573, -0.33333333333333331, -
0.57735026918962573 };

  static  double P[100] = { 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,10, 0, 0, 0, 0, 0, 0,
0, 0,
                            0, 0,10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,10, 0, 0, 0, 0,
0, 0,
                            0, 0, 0, 0,10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,10, 0, 0,
0, 0,
                            0, 0, 0, 0, 0, 0,10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,10,
0, 0,
                            0, 0, 0, 0, 0, 0, 0, 0,10, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,10 };

  static  double Qk[100] = {    1.0E-9, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
                                0.0, 1.0E-9, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
                                0.0, 0.0, 1.0E-9, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
                                0.0, 0.0, 0.0, 1.0E-9, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
                                0.0, 0.0, 0.0, 0.0, 1.0E-9, 0.0, 0.0, 0.0, 0.0, 0.0,
                                0.0, 0.0, 0.0, 0.0, 0.0, 1.0E-9, 0.0, 0.0, 0.0, 0.0,
                                0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0E-9, 0.0, 0.0, 0.0,
                                0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0E-9, 0.0, 0.0,
                                0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0E-9, 0.0,
                                0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0E-9
};
```

```
  static double Rk[4] = { 1.0E-6, 0.0, 0.0, 1.0E-6 };

  static  double Fund[100] = { 0.996917333733128, 0.0784590957278449, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0,
                              -0.0784590957278449, 0.996917333733128, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0,
                               0.0, 0.0, 0.987688340595138,  0.156434465040231, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0,
                               0.0, 0.0, -0.156434465040231, 0.987688340595138, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0,
                               0.0, 0.0, 0.0, 0.0, 0.972369920397677,
0.233445363855905, 0.0, 0.0, 0.0, 0.0,
                               0.0, 0.0, 0.0, 0.0, -0.233445363855905,
0.972369920397677, 0.0, 0.0, 0.0, 0.0,
                               0.0, 0.0, 0.0, 0.0, 0.0, 0.0,  0.951056516295154,
0.309016994374947, 0.0, 0.0,
                               0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -0.309016994374947,
0.951056516295154, 0.0, 0.0,
                               0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.923879532511287,   0.38268343236509,
                               0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,-
0.38268343236509 ,  0.923879532511287 };

  static  double Xm[200] =
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0};

  static  double Z[40] =
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};


  //-----------------------------------------------------------------
  //  Find sigma points X
  //  R = chol(A) where R'*R = A

  for (r2 = 0; r2 < 100; r2++) {
    Pm[r2] = 10.0 * P[r2];
  }

  chol(Pm);
  for (r2 = 0; r2 < 10; r2++) {
    for (ia = 0; ia < 10; ia++) {
      A[ia + 10 * r2] = Pm[r2 + 10 * ia];
    }
  }

  ib = 0;
  for (r2 = 0; r2 < 20; r2++) {
    ia = 0;
    for (k = 0; k < 10; k++) {
      b[ib] = x[ia];
      ia++;
```

```
    ib++;
  }
}

for (r2 = 0; r2 < 10; r2++) {
  memcpy((void *)&b_A[10 * r2], (void *)&A[10 * r2], 10U * sizeof(double));
}

for (r2 = 0; r2 < 10; r2++) {
  for (ia = 0; ia < 10; ia++) {
    b_A[ia + 10 * (r2 + 10)] = -A[ia + 10 * r2];
  }
}

for (r2 = 0; r2 < 20; r2++) {
  for (ia = 0; ia < 10; ia++) {
    X[ia + 10 * r2] = b_A[ia + 10 * r2] + b[ia + 10 * r2];
  }

  //  Size of the state vector multiplied by 2
  //  Predicted states for next time step computed from the sigma points X
  for (ia = 0; ia < 10; ia++) {
    Xm[ia + 10 * r2] = 0.0;
    for (ib = 0; ib < 10; ib++) {
      Xm[ia + 10 * r2] += Fund[ia + 10 * ib] * X[ib + 10 * r2];
    }
  }
}

//  Find the prediction mean and covariance matrix
sum(Xm, xm);
for (r2 = 0; r2 < 10; r2++) {
  xm[r2] /= 20.0;
}

memset((void *)&Pm[0], 0, 100U * sizeof(double));

// size n
for (k = 0; k < 20; k++) {
  for (r2 = 0; r2 < 10; r2++) {
    a[r2] = Xm[r2 + 10 * k] - xm[r2];
    mx[r2] = Xm[r2 + 10 * k] - xm[r2];
  }

  for (r2 = 0; r2 < 10; r2++) {
    for (ia = 0; ia < 10; ia++) {
      Pm[ia + 10 * r2] += a[ia] * mx[r2];
    }
  }
}

for (r2 = 0; r2 < 100; r2++) {
  Pm[r2] = Pm[r2] / 20.0 + Qk[r2];
}

// Predicted measurements sigma points
for (k = 0; k < 20; k++) {

  a21 = Xm[10 * k];
  a22 = Xm[1 + 10 * k];
  for (ib = 0; ib < 4; ib++) {
    a21 += Xm[((ib << 1) + 10 * k) + 2];
```

```
    a22 += Xm[((ib << 1) + 10 * k) + 3];
  }

  Z[k << 1] = a21;
  Z[1 + (k << 1)] = a22;
}

// Find mean and covariance
b_sum(Z, z);
for (r2 = 0; r2 < 2; r2++) {
  z[r2] /= 20.0;
}

for (r2 = 0; r2 < 4; r2++) {
  Pz[r2] = 0.0;
}

/*  size n */
for (k = 0; k < 20; k++) {
  for (r2 = 0; r2 < 2; r2++) {
    my[r2] = Z[r2 + (k << 1)] - z[r2];
    b_b[r2] = Z[r2 + (k << 1)] - z[r2];
  }

  for (r2 = 0; r2 < 2; r2++) {
    for (ia = 0; ia < 2; ia++) {
      Pz[ia + (r2 << 1)] += my[ia] * b_b[r2];
    }
  }
}

for (r2 = 0; r2 < 4; r2++) {
  Pz[r2] = Pz[r2] / 20.0 + Rk[r2];
}

//  2*n
sum(Xm, mx);
for (r2 = 0; r2 < 10; r2++) {
  mx[r2] /= 20.0;
}

b_sum(Z, my);
for (r2 = 0; r2 < 2; r2++) {
  my[r2] /= 20.0;
}

/*  size n */
memset((void *)&Pxz[0], 0, 20U * sizeof(double));
for (k = 0; k < 20; k++) {
  for (r2 = 0; r2 < 10; r2++) {
    a[r2] = Xm[r2 + 10 * k] - mx[r2];
  }

  for (r2 = 0; r2 < 2; r2++) {
    b_b[r2] = Z[r2 + (k << 1)] - my[r2];
  }

  for (r2 = 0; r2 < 2; r2++) {
    for (ia = 0; ia < 10; ia++) {
      Pxz[ia + 10 * r2] += a[ia] * b_b[r2];
    }
  }
```

```
}

for (r2 = 0; r2 < 20; r2++) {
  Pxz[r2] /= 20.0;
}

//  Kalman gain matrix */
for (r2 = 0; r2 < 2; r2++) {
  for (ia = 0; ia < 2; ia++) {
    c_A[ia + (r2 << 1)] = Pz[r2 + (ia << 1)];
  }
}

for (r2 = 0; r2 < 10; r2++) {
  for (ia = 0; ia < 2; ia++) {
    B[ia + (r2 << 1)] = Pxz[r2 + 10 * ia];
  }
}

if (fabs(c_A[1]) > fabs(c_A[0])) {
  ib = 1;
  r2 = 0;
} else {
  ib = 0;
  r2 = 1;
}

a21 = c_A[r2] / c_A[ib];
a22 = c_A[2 + r2] - a21 * c_A[2 + ib];
for (k = 0; k < 10; k++) {
  Y[1 + (k << 1)] = (B[r2 + (k << 1)] - B[ib + (k << 1)] * a21) / a22;
  Y[k << 1] = (B[ib + (k << 1)] - Y[1 + (k << 1)] * c_A[2 + ib]) / c_A[ib];
}

for (r2 = 0; r2 < 2; r2++) {
  for (ia = 0; ia < 10; ia++) {
    Pxz[ia + 10 * r2] = Y[r2 + (ia << 1)];
  }
}

//  Find the state estimates
b_VALa[0] = VALa;
b_VALa[1] = VALb;
b_VALa[2] = VALc;
for (r2 = 0; r2 < 2; r2++) {
  a21 = 0.0;
  for (ia = 0; ia < 3; ia++) {
    a21 += d_A[r2 + (ia << 1)] * b_VALa[ia];
  }

  my[r2] = a21 - z[r2];
}

for (r2 = 0; r2 < 10; r2++) {
  a21 = 0.0;
  for (ia = 0; ia < 2; ia++) {
    a21 += Pxz[r2 + 10 * ia] * my[ia];
  }

  x[r2] = xm[r2] + a21;
}
```

```
// Estimation error covariance matrix
for (r2 = 0; r2 < 10; r2++) {
  for (ia = 0; ia < 2; ia++) {
    B[r2 + 10 * ia] = 0.0;
    for (ib = 0; ib < 2; ib++) {
      B[r2 + 10 * ia] += Pxz[r2 + 10 * ib] * Pz[ib + (ia << 1)];
    }
  }
}

for (r2 = 0; r2 < 10; r2++) {
  for (ia = 0; ia < 10; ia++) {
    a21 = 0.0;
    for (ib = 0; ib < 2; ib++) {
      a21 += B[r2 + 10 * ib] * Pxz[ia + 10 * ib];
    }

    P[r2 + 10 * ia] = Pm[r2 + 10 * ia] - a21;
  }
}

// Prevent numerical problem
for (r2 = 0; r2 < 10; r2++) {
  for (ia = 0; ia < 10; ia++) {
    dv0[ia + 10 * r2] = 0.5 * (P[ia + 10 * r2] + P[r2 + 10 * ia]);
  }
}

for (r2 = 0; r2 < 10; r2++) {
  memcpy((void *)&P[10 * r2], (void *)&dv0[10 * r2], 10U * sizeof(double));
  xo[r2] = x[r2];
}


 // The the fundamental harmonic
 Amp0=sqrt((xo[0]*xo[0])+(xo[1]*xo[1]));
 ang0=(atan2(xo[1],xo[0]))+(3.1415926/2);
 V1_=cos(ang0)+((sin(ang0))*I);
 V2_=cos(2*3.1415926*50*1/4000*ct)+sin(2*3.1415926*50*1/4000*ct)*I;
 Ang0=V1_/V2_;
 aang0=atan2(cimag(Ang0),creal(Ang0));
 Phs0=aang0*180/3.1415926;

 // The the second harmonic
 /*Amp2=sqrt((xo[2]*xo[2])+(xo[3]*xo[3]));
 ang2=(atan2(xo[3],xo[2]))+(3.1415926/2);
 V3_=cos(ang2)+((sin(ang2))*I);
 V4_=cos(2*2*3.1415926*50*1/4000*ct)+sin(2*2*3.1415926*50*1/4000*ct)*I;
 Ang2=V3_/V4_;
 aang2=atan2(cimag(Ang2),creal(Ang2));
 Phs2=aang2*180/3.1415926;

 // The the fifth harmonic
 Amp5=sqrt((xo[8]*xo[8])+(xo[9]*xo[9]));
 ang5=(atan2(xo[9],xo[8]))+(3.1415926/2);
 V5_=cos(ang5)+((sin(ang5))*I);
 V6_=cos(5*2*3.1415926*50*1/4000*ct)+sin(5*2*3.1415926*50*1/4000*ct)*I;
 Ang5=V5_/V6_;
 aang5=atan2(cimag(Ang5),creal(Ang5));
 Phs5=aang5*180/3.1415926;*/

 printf("SMV_No=%d   Fund_Amp= %.2f pu  Fund_Phase_Angle= %.2f\n",ct,Amp0,Phs0 );
```

```
   struct PHASOR retval ={Amp0,Phs0};
   return retval;

}// End of UKF Routine


//-------------------------------------------------------------------------
// Start of SMV Routine
int print_pkt(char *buf, int len)// printout SMV frame

{
       int j = 0;
       for(j = 0; j < len; j++ )
       {
              if((j%16) == 0 && j != 0 )
              printf("\n");
              printf("%02x ",*(buf+j)& 0xFF );
       }
       printf("\n====Packet END====\n");
       return 0;
}


//-----------------------------------------------
// Define the 4th layer
// One Dataset "SMV_dataset"

       struct SMV_dataset {
       // Instantaneous magnitude of Currents (Phase A, B, C and neutral) in Ampere
              int32_t InnATCTR1_Amp_instMag_i;//Magnitude
              int32_t InnATCTR1_Amp_q;        //Quality
              int32_t InnBTCTR2_Amp_instMag_i;//Magnitude
              int32_t InnBTCTR2_Amp_q;        //Quality
              int32_t InnCTCTR3_Amp_instMag_i;//Magnitude
              int32_t InnCTCTR3_Amp_q;        //Quality
              int32_t InnNTCTR4_Amp_instMag_i;//Magnitude
              int32_t InnNTCTR4_Amp_q;
       // Instantaneous magnitude of Voltages (Phase A, B, C and neutral) in Volt
              int32_t UnnATVTR1_Vol_instMag_i;//Magnitude
              int32_t UnnATVTR1_Vol_q;        //Quality
              int32_t UnnBTVTR2_Vol_instMag_i;//Magnitude
              int32_t UnnBTVTR2_Vol_q;        //Quality
              int32_t UnnCTVTR3_Vol_instMag_i;//Magnitude
              int32_t UnnCTVTR3_Vol_q;        //Quality
              int32_t UnnNTVTR4_Vol_instMag_i;//Magnitude
              int32_t UnnNTVTR4_Vol_q;        //Quality
       }__attribute__((__packed__));

//-------------------------------------------------
// Define the 3rd layer
// Application Service Data Unit (ASDU)
// ASDU is coded according to 9-2 LE and ASN.1 encoding rules

       struct SMV_ASDU {
              int8_t SequenceASDU_tag;     //0x30
              int8_t SequenceSMV_ASDUngth; //0x01
              int8_t svID_tag;             //0x80
              int8_t svID_Length;          //0x0A
              int8_t svID[11];             //according to the naming rules is in the
IEC 61869-9
              int8_t smpCnt_tag;           //0x82
              int8_t smpCnt_Length;        //0x02
```

```
                int16_t smpCnt;              //Samples Counter
                int8_t confRev_tag;          //0x83
                int8_t confRev_Length;       //0x04
                int32_t confRev;             //Configuration revision number
                int8_t smpSynch_tag;         //0x85
                int8_t smpSynch_Length;      //0x01
                int8_t smpSynch;             //synchronization, 2-> global 1-> local
0-> not sych
                int8_t smpRate_tag;          //0x86
                int8_t smpRate_Length;       //0x02
                int16_t smpRate;             //80 samples per cycle (0x0050 in hex)
                int8_t SequenceofData_tag;   //0x87
                int8_t SequenceofData_Length;//0x40
                struct SMV_dataset DataSet;
        }__attribute__((__packed__));


//-------------------------------------------------------
// Define the 2nd layer
// Application Protocol Data Unit  (APDU)
        struct SMV_APDU {
                int8_t savPDUtag;
                int8_t savPDULength;  //sizeof SMV_APDU
                int8_t noASDUtag;
                int8_t noSMVASDUngth; //sizeof noASDU
                int8_t noASDU;        //One ASDU (0x01 hex)
                int8_t SequenceofASDUtag;
                int8_t SequenceofSMVASDUngth;//sizeof all ASDU
                struct SMV_ASDU SMV_ASDU1;
        }__attribute__((__packed__));



// Sampled Measrued Value frame structure
        struct SMV_FRM {
                //Header MAC
                int8_t destination_mac[6];
                int8_t source_mac[6];
                //Start Priority tagged
                int16_t TPID;     // For 802.1Q VLAN, its 0x8100
                int16_t TCI;      // 0x4000
                // Start Ethertype PDU
                int16_t Ethertype;// For SMV, its 0x88BA
                int16_t APPID;    // 0x4000 application identifier
                int16_t Length;   // After APPID
                int16_t reserved1;
                int16_t reserved2;
                //Adding the SMV_APDU (the application protocol data unit)
                struct SMV_APDU SMV_apdu;
                //pad bytes if necessary
                //Frame check sequence
        }__attribute__((__packed__));


typedef struct SMV_FRM FRM;
// End of SMV Routine


 int main (int argc, char **argv)
 {
  int i, sd, status,smpCnt;
  unsigned char *ether_frame;
```

```
  FRM  *FRM;
  void *tmp;
  int CurrentA_r,CurrentB_r,CurrentC_r,VoltageA_r,VoltageB_r,VoltageC_r;
  double CurrentA,CurrentB,CurrentC,VoltageA,VoltageB,VoltageC;
  static timer=0;
  double mat_smv[4000][7];

  FILE* smv_phs;// for printing out to file


// Allocate memory for various arrays.

  tmp = (unsigned char *) malloc (IP_MAXPACKET * sizeof (unsigned char));
  if (tmp != NULL) {
    ether_frame = tmp;}
  else {
    fprintf (stderr, "ERROR: Cannot allocate memory for array 'ether_frame'.\n");
    exit (EXIT_FAILURE);
  }
   memset (ether_frame, 0, IP_MAXPACKET * sizeof (unsigned char));
   printf ("Ethernet frame Size:%lu\n",sizeof ether_frame);
// Submit request for a raw socket descriptor.
  if ((sd = socket (PF_PACKET, SOCK_RAW, htons (ETH_P_ALL))) < 0) {
    perror ("socket() failed ");
    exit (EXIT_FAILURE);
  }

// Listen for incoming ethernet frame from socket sd.
// The  ethernet frame expected is in the form:
// MAC (6 bytes) + MAC (6 bytes) + ethernet type (2 bytes)+...
// ... ethernet data (ARP header) (28 bytes)
// Keep at it until we get a reply.

   FRM=(FRM *)(ether_frame);



  while (timer<3999)
{
   struct timeval tv1,tv2;
   struct timezone tz1,tz2;
   double t_rec;


 if ((status = recv (sd, ether_frame, IP_MAXPACKET, 0)) < 0)
      {
        perror ("recv() failed:");
        exit (EXIT_FAILURE);
      }

else

/*get the locol time of the computer when SV is recieved*/
   gettimeofday(&tv1, &tz1);
   t_rec=(tv1.tv_sec*1000000)+tv1.tv_usec;

// Uncomment if you want to printout  the SMV frame content
/*
//Print out contents of received ethernet frame.
  printf ("\nEthernet frame header:\n");
  printf ("Destination MAC (this node): ");
  for (i=0; i<5; i++)
```

```
  {
    printf ("%02x:", ether_frame[i]);
  }
  printf ("%02x\n", ether_frame[5]);
  printf ("Source MAC: ");
  for (i=0; i<5; i++)
  {
    printf ("%02x:", ether_frame[i+6]);
  }
  printf ("%02x\n", ether_frame[11]);
  printf ("Ethernet type code: 0x88BA (35002 = IEC 61850 ): %u\n", ntohs(FRM-
>Ethertype));

// Print out contents of Ethertype PDU.
  printf ("\nContents of Ethertype PDU:\n");
  printf ("APPID: %u\n", ntohs(FRM->APPID));
  printf ("Length: %u\n", ntohs(FRM->Length));
  printf ("reserved1: %u\n", ntohs(FRM->reserved1));
  printf ("reserved2: %u\n", ntohs(FRM->reserved2));

// Print out contents of SMV_APDU.
  printf ("\nContents of SMV_APDU:\n");
  printf ("savPDU_tag: 0x%x\n", (FRM->SMV_APDU.savPDU_tag));
  printf ("savPDULength: 0x%x\n", (FRM->SMV_APDU.savPDULength));
  printf ("noASDUtag: 0x%x\n", (FRM->SMV_APDU.noASDUtag));
  printf ("noSMVASDUngth: 0x%x\n", (FRM->SMV_APDU.noSMVASDUngth));
  printf ("noASDU: 0x%x\n", (FRM->SMV_APDU.noASDU));
  printf ("SequenceofASDUtag: 0x%x\n", (FRM->SMV_APDU.SequenceofASDUtag));
  printf ("SequenceofSMVASDUngth: 0x%x\n", (FRM->SMV_APDU.SequenceofSMVASDUngth));

// Print out contents of SMV_ASDU1.

  printf ("\nContents of SMV_ASDU1:\n");
  printf ("SequenceASDU_tag: 0x%x\n", (FRM->SMV_APDU.SMV_ASDU1.SequenceASDU_tag));
  printf ("SequenceSMV_ASDUngth: 0x%x\n", (FRM-
>SMV_APDU.SMV_ASDU1.SequenceSMV_ASDUngth));
  printf ("svID_tag: 0x%x\n", (FRM->SMV_APDU.SMV_ASDU1.svID_tag));
  printf ("svID_Length: 0x%x\n", (FRM->SMV_APDU.SMV_ASDU1.svID_Length));
  printf ("svID: %s\n", (FRM->SMV_APDU.SMV_ASDU1.svID));
  printf ("smpCnt_tag: 0x%x\n", (FRM->SMV_APDU.SMV_ASDU1.smpCnt_tag));
  printf ("smpCnt_Length: 0x%x\n", (FRM->SMV_APDU.SMV_ASDU1.smpCnt_Length));
  printf ("smpCnt: %u\n", ntohs(FRM->SMV_APDU.SMV_ASDU1.smpCnt));
  printf ("confRev_tag: 0x%x\n", (FRM->SMV_APDU.SMV_ASDU1.confRev_tag));
  printf ("confRev_Length: 0x%x\n", (FRM->SMV_APDU.SMV_ASDU1.confRev_Length));
  printf ("confRev: 0x%u\n", ntohl(FRM->SMV_APDU.SMV_ASDU1.confRev));
  printf ("smpSynch_tag: 0x%x\n", (FRM->SMV_APDU.SMV_ASDU1.smpSynch_tag));
  printf ("smpSynch_Length: 0x%x\n", (FRM->SMV_APDU.SMV_ASDU1.smpSynch_Length));
  printf ("smpSynch: 0x%x\n", (FRM->SMV_APDU.SMV_ASDU1.smpSynch));
  printf ("SequenceofData_tag: 0x%x\n", (FRM-
>SMV_APDU.SMV_ASDU1.SequenceofData_tag));
  printf ("SequenceofData_Length: 0x%x\n", (FRM-
>SMV_APDU.SMV_ASDU1.SequenceofData_Length));

// Print out contents of Dataset (Measurments).

  printf ("Current A= %d\n", ntohl(FRM-
>SMV_APDU.SMV_ASDU1.DataSet.InnATCTR1_Amp_instMag_i));
  printf ("Current B= %u\n", ntohl(FRM-
>SMV_APDU.SMV_ASDU1.DataSet.InnBTCTR2_Amp_instMag_i));
  printf ("Current C= %u\n", ntohl(FRM-
>SMV_APDU.SMV_ASDU1.DataSet.InnCTCTR3_Amp_instMag_i));
```

```
  printf ("Current N= %u\n", ntohl(FRM-
>SMV_APDU.SMV_ASDU1.DataSet.InnNTCTR4_Amp_instMag_i));

  printf ("Voltage A= %u\n", ntohl(FRM-
>SMV_APDU.SMV_ASDU1.DataSet.UnnATVTR1_Vol_instMag_i ));
  printf ("Voltage B= %u\n", ntohl(FRM-
>SMV_APDU.SMV_ASDU1.DataSet.UnnBTVTR2_Vol_instMag_i ));
  printf ("Voltage C= %u\n", ntohl(FRM-
>SMV_APDU.SMV_ASDU1.DataSet.UnnCTVTR3_Vol_instMag_i ));
  printf ("Voltage N= %u\n", ntohl(FRM-
>SMV_APDU.SMV_ASDU1.DataSet.UnnNTVTR4_Vol_instMag_i));
*/

  CurrentA_r=ntohl(FRM->SMV_APDU.SMV_ASDU1.DataSet.InnATCTR1_Amp_instMag_i);
  CurrentB_r=ntohl(FRM->SMV_APDU.SMV_ASDU1.DataSet.InnBTCTR2_Amp_instMag_i);
  CurrentC_r=ntohl(FRM->SMV_APDU.SMV_ASDU1.DataSet.InnCTCTR3_Amp_instMag_i);
  VoltageA_r=ntohl(FRM->SMV_APDU.SMV_ASDU1.DataSet.UnnATVTR1_Vol_instMag_i );
  VoltageB_r=ntohl(FRM->SMV_APDU.SMV_ASDU1.DataSet.UnnBTVTR2_Vol_instMag_i );
  VoltageC_r=ntohl(FRM->SMV_APDU.SMV_ASDU1.DataSet.UnnCTVTR3_Vol_instMag_i );
  smpCnt=ntohs(FRM->SMV_APDU.SMV_ASDU1.smpCnt);

  CurrentA=CurrentA_r*0.001;
  CurrentB=CurrentB_r*0.001;
  CurrentC=CurrentC_r*0.001;
  VoltageA=VoltageA_r*0.01;
  VoltageB=VoltageB_r*0.01;
  VoltageC=VoltageC_r*0.01;


  struct PHASOR  phasor = UKF(smpCnt,CurrentA,CurrentB,CurrentC);
  printf("AMP1: %.2f\n", phasor.AMP1);
  printf("PHS1: %.2f\n", phasor.PHS1);


  mat_smv[timer][1]=smpCnt;
  mat_smv[timer][2]=t_rec;
  mat_smv[timer][3]=CurrentA;
  mat_smv[timer][4]=CurrentB;
  mat_smv[timer][5]=CurrentC;
  mat_smv[timer][6]=phasor.AMP1;
  mat_smv[timer][7]=phasor.PHS1;

  timer++;
}



  smv_phs=fopen("SMV.txt","w");
     if(smv_phs!=NULL)
    {

        for(i=0;i<=timer;i++)
        fprintf(smv_phs,"%f   %.4f   %.4f   %.4f   %.4f   %.4f
%.4f\n",mat_smv[i][1],mat_smv[i][2],
mat_smv[i][3],mat_smv[i][4],mat_smv[i][5],mat_smv[i][6],mat_smv[i][7] );
    }
    else
        printf("ERROR!\nThe FILE SMV.txt or the PHASOR.txt didn't open
successfully!\n");
```

```
}//here, is the end of the program
```