Synchronized Phasor Measurement Units

Implementation Of Phasor Measurement Unit Function On An HVDC Control Platform

SAMARTH DEO

KTH Electrical Engineering

Master's Degree Project
Stockholm, Sweden December 28, 2013

XR-EE-ICS 2013:017

ABSTRACT

Power Systems most often operate in the margins of stability limits. With power grids becoming even more automated now, Phasor Measurement Units (PMU's) and Phasor Data Concentrators (PDC's) become essential for real-time control of the system. Since PMU's are time synchronized, the phasors can be compared at the substations, studied for any faults and analyzed at the same time. These PMU's report the measured magnitude, phasor angle of voltages and currents in real synchronized time in different locations. One good way to measure these quantities is to use the Discrete Fourier Transform (DFT) and analyze the signal as a digital signal. However with transients and noise present in the input signal, DFT might not be the best approach for measurement and/or protection. With the IEEE C37.118 (Standard for Synchrophasor for Power Systems) 2011 version emphasizing on the importance of the two classes of PMU's- P-class for Protection and M-class for Measurement. There is a high precision required for M-class PMU's whereas a good reporting rate for the P-class. Recently The North-American Synchrophasor Initiative (NASPI) and Western Electricity Coordinating Council (WECC) also gave filtering specifications and frequency response of an industry compliant PMU. This thesis discusses the various frequency estimation algorithms, which are compliant with the NASPI/WECC standards. Further such an algorithm is implemented on ABB proprietary hardware and tested against dynamic tests.

Keywords: PMU, PDC, HVDC, WAMS, GPS, IEEE C37.118.2005.

ACKNOWLEDGMENTS

I would like to start by thanking ABB for letting me do this thesis with them on their control platform. This thesis was carried out at 'ABB AB HVDC', Ludvika between May and September 2013. I would like to thank Hans Bjorklund, my advisor at ABB for his guidance throughout the project. I would also like to thank my manager Bjorn Falldin for providing me with all the resources and making me comfortable during my work. A special thanks to Gunnar Rydh, Fredrik Tinnerholm and Meenakshi Karthick for their guidance and helping me with several problems all throughout. Also to the entire team at PKC Department, ABB HVDC, Ludvika for giving me a nice, motivated environment.

I would like to express my kind and sincere gratitude to my examiner Prof. Lars Nordström for his continuous support to my master thesis research and study. I would also like to thank my adviser Zhu Kun at School of Electrical Engineering, Industrial Information and Control Systems Department, for his insightful comments, possible problems and suggestions. I appreciate his patience, motivation and immense knowledge. His continuous guidance helped me sail throughout the journey of study and research. This thesis wouldn't have been possible without him.

Last but not the least, I take this opportunity to express my profound gratitude to my parents and brother for their love, continuous support- both materially and spiritually and also for this wonderful opportunity to study abroad.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ROCOF	Rate of Change of Frequency iv
PMU	Phasor Measurement Units vi
DFT	Discrete Fourier Transform 15
MACH	Modular Advanced Control for HVDC vi
IEEE C37.118	Standard for Synchrophasors for Power Systems™ 1
NASPI	North American Synchrophasor Initiative 1
HVDC	High Voltage Direct Current 2
TVE	Total Vector Error 6
IEEE	Institute of Electrical and Electronics Engineers® 5
UTC	Coordinated Universal Time 6
RMS	Root Mean Square 6
GPS	Global Positioning System 6
PPS	Pulse Per Second 7
PTP	Precise Time protocol 14
PDC	Phasor Data Concentrator 10
TCP	Transmission Control Protocol 11
UDP	User Datagram Protocol 11
FPGA	Field Programmable Gate Array 16
DRAM	Dynamic random-access memory 16
WECC	Western Electricity Coordinating Council 3
FIR	Finite Impulse Response 18
ADC	Analog to Digital Converters 27
RTOS	Real Time Operating System 27
TFR	Transient Fault Recorder 37
IIR	Infinite Impulse Response 35
WAMS	Wide-Area Monitoring System 20

1 INTRODUCTION

This chapter gives a basic introduction to Phasor Measurement Units. It defines the scope of the thesis and the structure that the report follows, i.e., explaining what is covered in each chapter and also highlights the various tasks performed in the thesis.

1.1 BACKGROUND

Phasor Measurement Units (PMU's) have been of interest to power system engineers for quite sometime now. Synchronized Voltage and Current phasors provide engineers with enough details to analyze an entire Power Grid to perform closed-loop controls at a time frame of fractions of a second. PMU's offer an opportunity to analyze the cause in the power system units that lead to black-outs. The state of power systems can be analyzed in the simplest way by analyzing the positive sequence measurements. As of 2008, there were 24 commercial manufacturers of PMU. Apparently, the performance of the intended application is highly dependent on the quality of the synchrophasors computed by the PMU's. The measurement bias can be traced back to the input with off-nominal frequency, harmonic distortion and power system transients. There are many research initiatives to propose new algorithms with the purpose to improve the measurement quality in hostile environments, for example the work presented in [2], [9], [17], [10] and [6]. Particularly in industry, efforts have also been made to propose metric to evaluate the performance of the PMU. Two synchrophasor accuracy class are specified in Standard for Synchrophasors for Power Systems™ (IEEE C37.118) standard depending on the input frequency deviations from the nominal values which also covers the dynamic response of a PMU. In [5], North America Synchrophasor Initiative North American Synchrophasor Initiative (NASPI) proposes a more complete framework, where requirements in power system steady and transient state are both covered. This, by far, provides the most comprehensive requirements regarding the PMU performance, and more importantly, this report is also endorsed by several power utilities.

A generalized PMU is represented as in the Figure 1.1 on page 2.

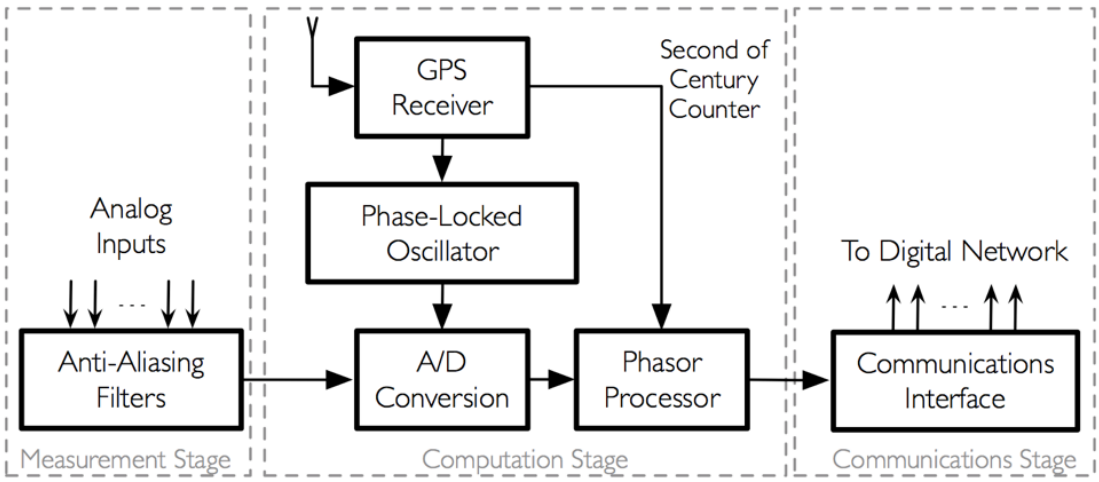

Figure 1.1: A generalized PMU

As seen above, a PMU has three major components: the measurement part(consisting of Anti-aliasing filters), the Phasor Computation part(Analog to Digital Converters, GPS Receivers, and a Phasor Processor) and the Communication part(to stream the computed data in C37.118 defined format.)

1.2 SCOPE, THESIS GOALS AND OBJECTIVES

This thesis focuses on the investigation of the compliance of the available synchrophasor algorithms to the NASPI PMU performance requirements, and also elicitation of their requirements on the hardware platforms, i.e., computation power, storage size and etc. Given the input from the first phase, the most suitable algorithm is chosen, and implemented in the ABB platform to control High Voltage Direct Current (HVDC) systems-MACH3™. This includes doing an in-depth literature survey, testing the implemented model on MATLAB and SIMULINK. Thereafter implementing on the MACH3 computer.

The scope does not cover implementation of the IEEE C37.118-2005(and later versions) library and also not bringing this thesis to a full-fledged product.

The objectives of the thesis can be summarized as:

- Literature survey of methods to compute synchrophasors. A comprehensive study of eight different algorithms with sufficient details (to the level that the algorithm can be implemented) provided by the articles.
- A chapter in report concerning study of the compliance of these methods to the NASPI PMU calibration report. Such an investigation is expected to be performed in a simulation environment, for example Matlab.

- A chapter in report introducing the ABB HVDC control platform, the focus is on its hardware capacities and software tools that are relevant to the implementation.
- Implementation of synchrophasor algorithm.
- Master thesis report

1.3 STRUCTURE

- Chapter 1 of this report covers the introductory part of the thesis, the scope and the remaining structure of this report.
- Chapter 2 of this report covers the literature survey required and done for this project. A comprehensive set of 10 papers was chosen to analyze and infer before taking any decision on implementation. A section of the report describes about the IEEE C37.118 Standard and the definitions for synchrophasors. Finally it contains a section on potential applications of a PMU.
- Chapter 3 talks in detail about the implementation of the proposed algorithm, first in MATLAB to verify the outputs and then in the target hardware. A section of this chapters deals with tests and results performed.
- Chapter 4 contains conclusion and the future possible work.

1.4 RESEARCH METHODOLOGY

The implementation of the PMU algorithm is done in four steps.

- Elicitation of the requirements of a PMU.
  1. IEEE C37.118 Standard.
  2. Industry Requirements from NASPI and Western Electricity Coordinating Council (WECC).
- Review of the available algorithms; their pros and cons.
- Validation of algorithm on MATLAB/Simulink.
- Implementation in ABB MACH3 Controllers.

2 RELATED LITERATURE

This chapter reviews the relevant literature's to establish the research context necessary for the thesis. It includes a part on the IEEE C37.118 standard, a part on time synchronization and a large part on the existing research and papers on synchrophasor measurements.

The following sections aim to explore the developments in the field of phasor measurement units. Further, the background work of the thesis project is also presented.

The materials needed for this sake are mostly among the scientific papers, standards documentations and industrial projects publications. The emphasis is to discuss different aspects of the PMU's by combining knowledge with the materials reviewed and scored from mentioned sources. The literature review in our thesis is basically composed of an overview of the different algorithms, problems and results by different authors.

There have been several papers discussing the merits and demerits of algorithms related to PMU's. While a simple algorithm like discrete Fourier transform can give satisfactory results in absence of any distortion, harmonics, discussions in this section are related to algorithms which take care of above mentioned problems and also are efficient enough to be run on general hardware's, in the present case- ABB propriety hardware- PS700, which is ABB's proprietary control hardware.

2.1 IEEE C37.118 STANDARDS

The first synchrophasor standard was Institute of Electrical and Electronics Engineers® (IEEE) 1344-1995 and was formed in 1995. After almost a decade, this standard was revised and a new standard named IEEE C37.118.2005 came into existence. This has been further revised to IEEE C37.118.2011 in 2011. The synchrophasor standards act like a tool for defining PMU performance requirements and standardizing quality of measurement of PMU's. The IEEE C37.118.2005 provides a measurement standard under steady state conditions, improves time stamping methods defined in the previous standards and also introduces a method of determining measurement precision.

The current version of the protocol also lays emphasis on the ability and performance of a PMU in dynamic conditions, not included in the original standard. Additional

clarifications on the definitions of the phasors and synchrophasor have been made. The concepts of Total Vector Error (TVE) and compliance tests have been expanded with more clarity. Tests related to temperature variations have been added. Over all, the requirements for dynamic tests have been detailed and the theoretical limits on frequency and Rate of Change of Frequency ROCOF have been articulated.

2.1.1 Synchrophasor Definition

A phasor is magnitude and phase represented in form of a sinusoidal signal. Phasors become an important quantity from the fact that the active power in a power line is directly proportional to the sine of the angle difference between voltages at the two terminals of any power line. When these phasors are synchronized to the Universal Time Coordinated (Coordinated Universal Time (UTC)), they are termed synchrophasors. Hence UTC is the common basis of time for all PMU's. Therefore all the angle measurements by PMU's are directly comparable. This leads to valuable information for several electric power network based applications including wide area monitoring, protection and control.

Phasor is a representation of a sinusoidal waveform which is time invariant in frequency and amplitude. A sinusoidal wave(voltage for instance) can be given by:

$$V(t) = A * \cos(\omega t + \theta)$$

A phasor represents this function as a complex number V with a magnitude A and a phase angle θ. In general, the Root Mean Square (RMS) value is considered for amplitude hence a factor of $\sqrt{\frac{1}{2}}$ comes into picture, making it:

$$V(t) = A/\sqrt{2} * \cos(\omega t + \theta)$$

A synchronized phasor is a phasor tagged with a unique time stamp. Hence, a synchronized phasor can be defined as the magnitude and cosine of signal as referenced to an absolute point of time.

The time reference is usually given by a highly accurate clock such as Global Positioning System (GPS)- clock. The angle is measured by the phase difference between the peaks of the sinusoidal and the angle at reporting time, as explained in Figure 2.1. By synchronizing the sampling processes for different signals - which may be hundreds of miles apart, it is possible to put their phasors on the same phasor diagram. Figure 2.1 also illustrates the phase angle/UTC time relationship.

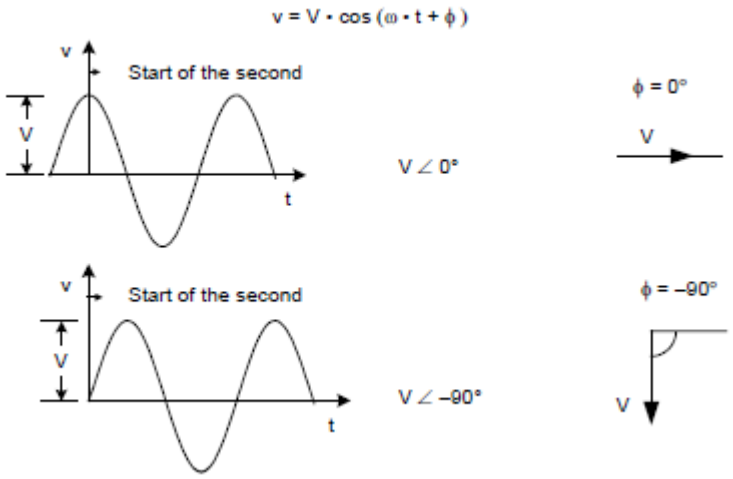

Figure 2.1: Convention for the Synchrophasor Representation

With reference to, Figure 2.1, the first figure has 0° phase difference, the reporting times phase is the peak of sinusoidal. While in the second figure, the phase difference is −90°.

2.1.2 Measurement Reporting Rate

The IEEE C37.118 standard provides PMU reporting rates at sub-multiples of the nominal power frequency which ranges from 10 frames/second upto the system nominal frequency. The actual rate to be adopted during implementation will be user select-able. It is this rate at which Frequency, (ROCOF) and TVE estimates will be made. Each frame is a combination of synchrophasor, frequency, ROCOF and these all correspond to the same time stamp by GPS. The required rate for 50Hz and 60Hz are given below in the table:

Table 2.1: Synchrophasor Reporting Rates

System Frequency	50Hz			60Hz					
Reporting Rates(F_s-frames per second)	10	25	50	10	12	15	20	30	60

2.1.3 Measurment Reporting Time

For a reporting rate N frames/second, the reporting times must be evenly spaced through each second with frame number 0 (numbered 0 through N–1) coincident with the UTC second rollover (usually the 1 Pulse Per Second (PPS) provided by GPS). These reporting times (time-tags) are to be used for determining the instantaneous values of the synchrophasor. This is illustrated in Figure 2.2, where the reporting times are at

0, T_0, $2T_0$, $3T_0$, $4T_0$, etc. If rates lower than 1/s are used, there shall be one report on the hour (xx:00:00) and evenly spaced thereafter with an integer number of seconds between reports according to the chosen rate in the absence of leap seconds. If a leap second occurs, the last interval in the hour shall be shorter or longer by that leap second.

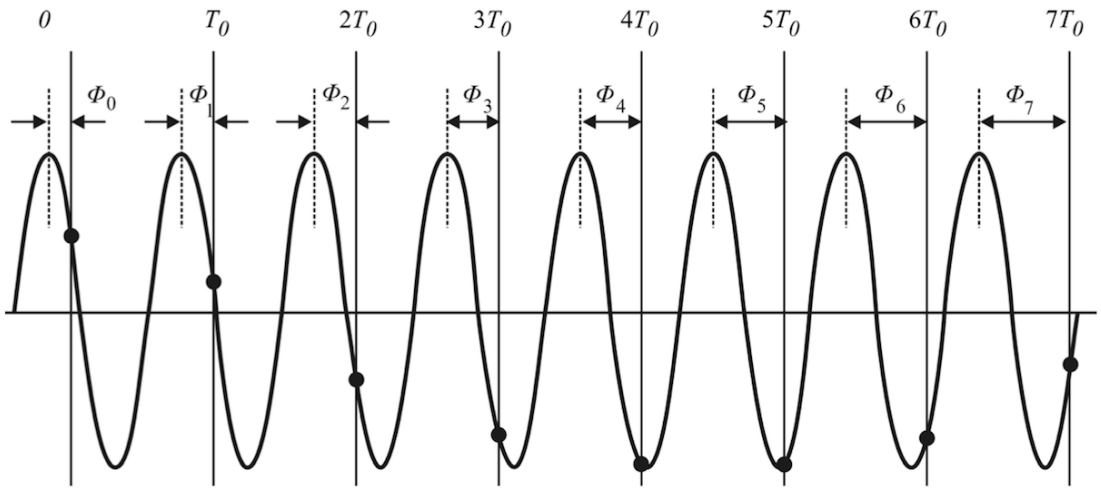

Figure 2.2: Time Tagging the Phasors

With reference to Figure 2.2, a sinusoidal with a frequency $f > f_0$ is over-served at instants that are T_0, seconds apart- the phase angle phi increases uniformly in relation to the frequency different $f\text{-}f_0$.

2.1.4 Evaluation of Measurement

The IEEE C37.118 Standard defines a term- Total Vector Error TVE) as a parameter for evaluation of measurement. TVE is defined as the difference between the estimated phasor from the unit under test and a perfect phasor of a theoretical synchrophasor, with both the phasor samples with the same time stamp. Simply put, TVE is the relative difference between a reference phasor and the output of a PMU. The equation used to compute TVE is:

$$\text{TVE} = \sqrt{\frac{(\hat{X}_r(n) - X_r(n))^2 + (\hat{X}_i(n) - X_i(n))^2}{X_r(n)^2 + X_i(n)^2}} \tag{2.1}$$

Where Xr(n) and Xi(n) are sequence of estimates given by the actual unit under test. Both being referenced to the same instance of time. TVE is an error based on the magnitude and phase of the unit in test. If TVE is 1%(maximum allowed for a good system) and there is no phase error between the two measurements at the same

time, it would imply that there is a magnitude error of 1%. Alternatively, if there is 0 magnitude error, 1% TVE would be because of 1% phase error. This corresponds to 0.01 radians or 0.57° and to 31μSec time error for a 50 Hz system.

2.1.5 Frequency and ROCOF Measurement Evaluation

Frequency measurement error(FE) and ROCOF errors are the absolute value of the difference between the theoretical values and the estimated values given in Hz and Hz/sec respectively. Both, the actual measured value and the expected phasor values have the same time tag. Frequency of any system is a key indicator of the load-generator balance. The quantity of generation loss is directly related to the frequency deviation. System frequency is also a good indicator of integrity of an interconnection during system events involving islanding. (For example, if a bus frequency at a point is 49.5 Hz and at a different place, far away, is 50.5 for a few minutes, it is an indication of system separation. Hence by looking at the entire interconnection, it is easy for the operator to identify islands and system separation points.)

The equations for calculating Frequency and ROCOF are given by:

Frequency Measurement Error

$$\text{FE} == |f_{\text{true}} - f_{\text{measured}}| = |\Delta f_{\text{true}} - \Delta f_{\text{measured}}| \tag{2.2}$$

Rate of Change of Frequency Error:

$$\text{RFE} == |(\mathrm{d}f/\mathrm{d}t)_{\text{true}} = (\mathrm{d}f/\mathrm{d}t)_{\text{measured}}| \tag{2.3}$$

2.1.6 Compliance with standards

The IEEE C37.118 requires that all PMU's be classified under one of the two classes of PMU's- P or M. P class PMU's refers to the Protection application where a very fast response is required. M class refers to the Measurement class where the speed of computation is not as important as the accuracy of measurement. Most vendors provide user the option of choosing this.

2.1.7 Measurement Reporting Latency

Since a PMU involves calculations at every step, there is a fare chance that there is a delay in the processed signals. General sources of errors include- filtering, windowing and estimation methods. The reporting rate and the performance class are the important factors since they determine the above said factors- filtering, windowing and estimation methods. The standard defines reporting latency as: The maximum time interval between the data report time as indicated by the data time stamp and the time when the data becomes available at the PMU output.

2.1.8 Communication Protocol for Phasors

The IEEE C37.118.2005 defines a data communication protocol for phasors and also explains the data format for the communication. Any communication system or media can be used for transmission of data. For identification of the phasor at the receiving end, which is Phasor Data Concentrator (PDC), the PMU has its own ID Code and status which are crafted together with the data for proper interpretation of the measured data. The library has been earlier implemented on Linux at the university. However, implementing the same on Windows was out of scope of this thesis work. The data format is explained below. It basically implements crafting the IEEE C37.118 messages in the following format:

- **Data Frame** carrying data measurements, e.g., current, voltage, frequency and other quantities.
- **Configuration Frame** carrying meta-data context for the data messages, i.e., carrying details or information about the phasor data in the data messages.
- **Header Frame** carrying user-defined information.
- **Command Frame**- Since the PMU only receives command messages (carrying commands to control the operation and configuration of the PMU), the library does not support crafting command messages. However, the library contains the necessary functions to check the sanity and validity of command messages. These functions, which are used by the PMU before accepting and executing a received command, allow the validation of the following specific information:

  1. The frame is well crafted, non-corrupted, and is sound. That is, the frame starts with a sync byte of 0xAA as the first byte, the frame type is indeed a command frame, the protocol number is the one that the PMU understands, the Cyclic Redundancy Code carried within the frame is identical to the computed one out from the received frame, and the carried frame size is correct.
  2. The frame is indeed addressed to the respective PMU, i.e., the ID code carried in the frame matches with the ID code assigned to and stored in the PMU.
  3. The intended command is a one that is supported by the PMU. Currently, the PMU understands only four commands, which are:
     – Turn OFF transmission of data,
     – Turn ON transmission of data,
     – Send the header frame,
     – Send the FIRST configuration frame, and
     – Send the SECOND configuration frame.

The entire packet of data, configuration and header frame, together with IDCODE is threaded in the following way:

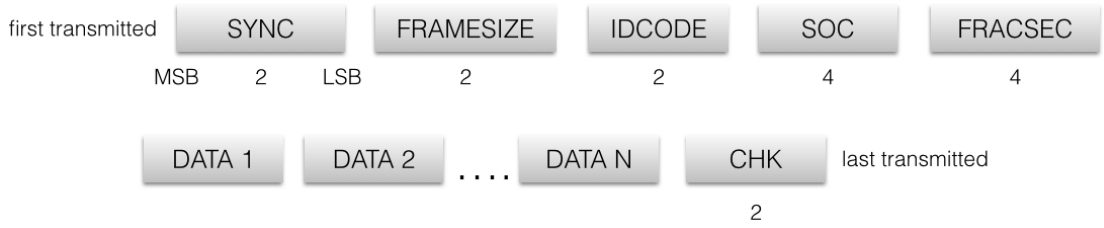

Figure 2.3: Example of frame transmission order

With respect to Figure 2.3, all message frames start with a 2-byte SYNC word followed by a 2-byte FRAMESIZE word, a 2-byte IDCODE, and a time stamp consisting of a 4-byte SOC and 4-byte fraction of second (FRACSEC), which includes a 24-bit fraction-of-second integer and an 8-bit Time Quality. The SYNC word provides synchronization and frame identification. IDCODE positively identifies the unit sending or receiving the message. Bits 4–6 in the SYNC word designate the frame type.

The simplest implementation of the IEEE C37.118 library is to split the entire task into two threads. In this case, it is assumed that the IEEE C37.118 library is running on a Linux machine and accepts the phasor stream over Transmission Control Protocol (TCP) from MATLAB/Simulink/HiDraw which is running on Windows. The library crafts the incoming message into the correct frames and sends them to a third computer over TCP/User Datagram Protocol (UDP) which runs the PDC.

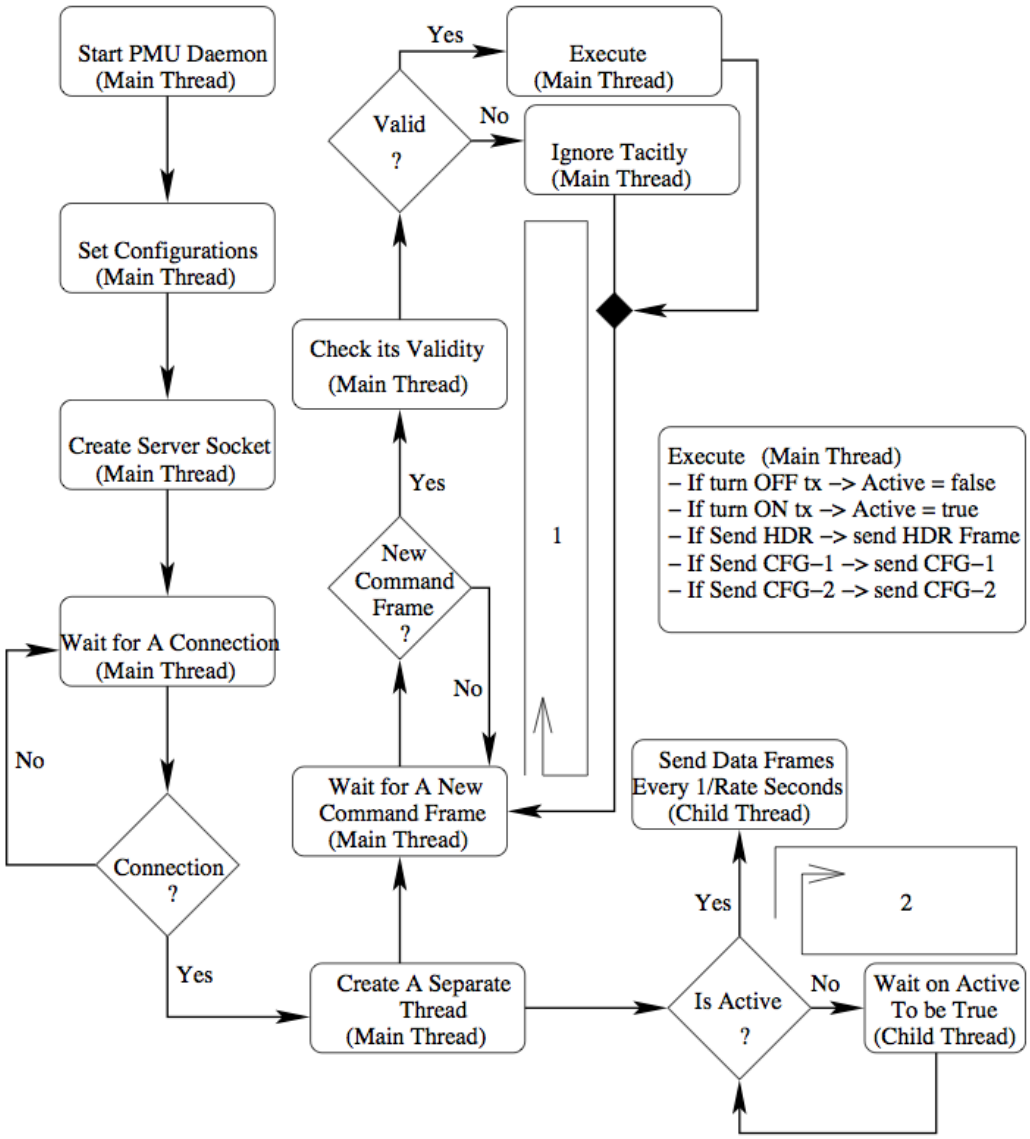

Figure 2.4: Flowchart for PMU communication

According to Figure 2.4 above, the PMU daemon is started first which sets the configuration for the communication. This daemon waits for a command frame from the receiver. When a command to stream data is received, a second thread is created which starts to stream data to the receiver. The configurations for the communication are basically the IP address, device IDCODE, port number, the reporting rate and the nominal frequency.

2.2 GPS SYNCHRONIZATION

This section covers the basic concepts of GPS Synchronization. Why is it needed in a PMU and how is it actually done!!

GPS stands for Global Positioning System. It is a satellite based navigation system developed by the U.S. Department of Defense [3]. It is run by 24 set of satellites and was deployed to overcome the problems of the previous available navigation systems. It came into full use and operations in 1994. Some uses of GPS include precise object tracking, vehicular navigation, space shuttle navigation, mining etc. among others. The 24 satellites orbit around in such a way that at any given time, from any given location on earth, four satellites can be accessed. Precise location is made from these four satellites.

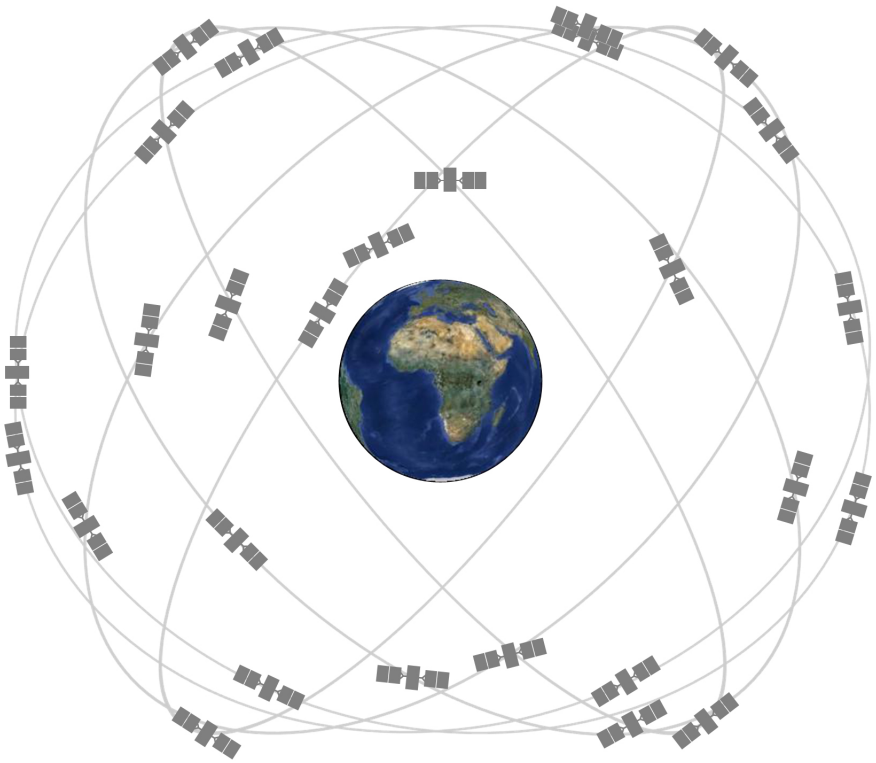

Figure 2.5: Constellation of GPS Satellites.

The satellites in the GPS constellation are arranged into six equally-spaced orbital planes surrounding the Earth. Each plane contains four "slots" occupied by baseline satellites. This 24-slot arrangement ensures there are at least four satellites in view from virtually any point on the planet. Source: `www.gps.gov`

In general, to compare any two quantities in the world, a common scale of measurement is needed. In case of Phase Measureing Units, to measure the angle difference, the two phasors measured at two points have to be on the same time scale. Hence

PMU's need to be time synchronized. The IEEE C37.118.2005 standard for synchrophasor requires a time accuracy of 1µSec. Further, a tolerance of 1 millisecond translates to an uncertainty of 20 degrees. Hence it becomes very important that the time accuracy of measurement unit be less than 1µSec. There are multiple industry standards developed to achieve this.

2.2.1 Pulse Per Second(PPS)

PPS stands for Pulse Per Seconds. The most common use of PPS is to connect to a low latency computer and synchronize it with the GPS time. It is notable that PPS does not specify time, it just specifies the start of second so it cant be used as a stand alone source of correct time. It has to be used with some other time source to increase the accuracy of the same.

2.2.2 IEEE 1588

IEEE 1588 and Precise Time protocol (PTP): PTP stands for Precise Time Protocol. PTP was first defined in the IEEE 1588-2002 standard and is designed to synchronize devices and computers over a network. According to the specifications, over a network, it can achieve a clock accuracy in the sub-micro second range. This makes this apt for measurement and control systems. IEEE 1588 has a master-slave architecture where the GPS is treated as master and the device(s) connected as slave(s). However if the GPS clock is not able to get the satellites signals, the slave having the most recent time starts to act as the master. More details on this in next chapter- Implementation in HiDraw.

2.3 PMU ALGORITHMS

2.3.1 [Phadke83] [11]

A.G. Phadke, J.S.Thorp, M.G.Adamiak - *"A New Measurement Technique for Tracking Voltage Phasors, Local System Frequency and Rate of Change of Frequency in Power Apparatus and Systems, IEEE Transactions on, vol.PAS-102, no.5, pp.1025-1038, May 1983"*.

Published in 1983 by A.G. Phadke, also known as father of PMU's, this was the one of the first papers which discussed the phasor measurement units in detail and bring out an efficient way to calculate phasors from the three phase source. However, written way before IEEE C37.118 came into picture, this does not necessarily comply with all the specifications of the standards including reporting rates. The algorithm is based on the following flowchart (Figure 2.6).

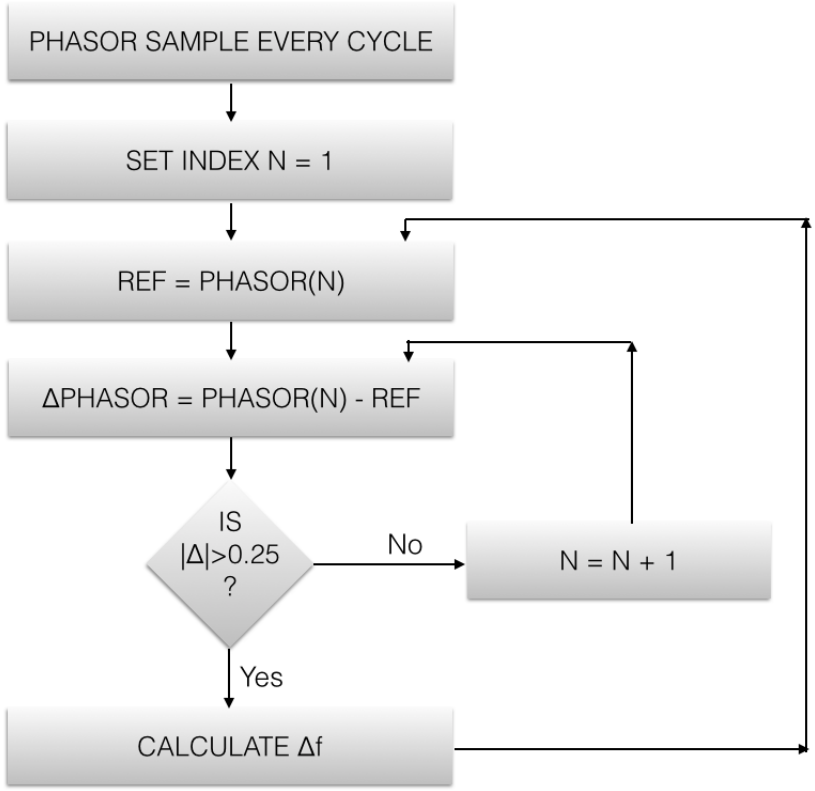

Figure 2.6: Flowchart showing the recursive DFT Algorithm

While the implementation of the algorithm is relatively easy, brings down the computation burden bringing out one phasor per sample, it is sensitive to harmonics and is inaccurate when the input deviates from the fundamental frequency. As a matter of fact for small frequency deviation from fundamental(in the range of 0.5Hz), the algorithm could take up to 3.2 seconds. For a +5 Hz deviation, the algorithm would take about 0.064 seconds.

2.3.2 [Paolone2011] [10]

M. Paolone, A. Borghetti and C.A. Nucci- *"A synchrophasor estimation algorithm for the monitoring of active distribution networks in steady state and transient conditions, in proc of PSCC 2011, Stockholm, Aug, 2011"*

The paper deals with the transient behavior of networks phasor quantities. There are several key features to the paper which include active rejection of harmonics, relatively low value of synchrophasor estimation; removal of anti-aliasing filters by adopting a sampling rate of 100 KHz, it is very difficult to achieve such a high sampling rate in real-time. It could be classified as a two-step algorithm where first step is to do a Discrete Fourier Transform (DFT) based analysis of the input signal. The second step is to improvise the synchrophasor phase at the fundamental frequency calculated in the

first step.

The algorithm is implemented in a National Instruments Embedded Real Time Micro-controller running at 400 MHz and a 2GB storage, 128 MB Dynamic random-access memory (DRAM) linked with a 3Mgate Field Programmable Gate Array (FPGA).

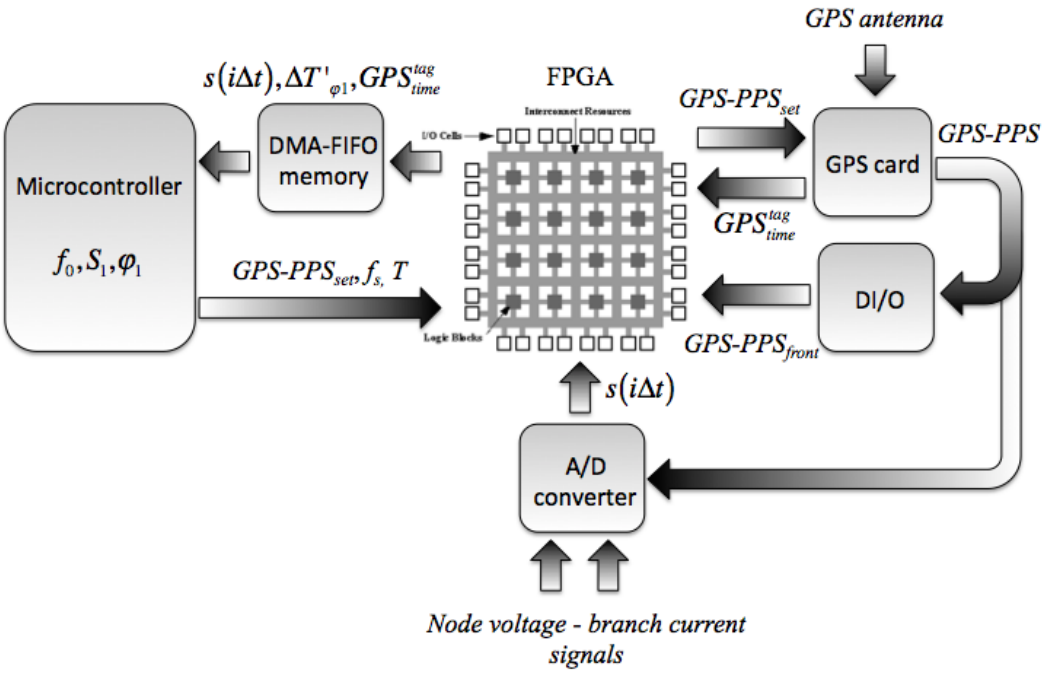

Figure 2.7: Implementation of the algorithm in NI Microcontroller

The FPGA in the center takes care of all the input pre-processing whereas the A/D Converters take the analog inputs. The micro-controller calculates the phasor quantities and writes back to the memory.

2.3.3 | [Tsinghua04] [17]

Maohai Wang, Yuanzhang Sun - *"A practical, precise method for frequency tracking and phasor estimation," Power Delivery, IEEE Transactions on , vol.19, no.4, pp. 1547- 1552, Oct. 2004"*

This paper describes a simple yet precise algorithm for frequency estimation. The authors have given the algorithm in detail and it utilizes an adaptive method to suppress the effects of harmonics. It is based on calculation of the DFT of the first window with no phase shift in the exponent then calculating the DFT of the same window with proper phase shift in the exponent. This is important because when the second window is shifted by one sample then the basis of the DFT will also have to rotate by one sample. Thereafter calculating the DFT of the current window twice, one with zero phase shift

and second with proper phase shift. The basis of DFT has to be rotated with every increment of the sample for the second DFT of each window.

2.3.4 [Brahma09] [15]

Sukumar M. Brahma, Phillip L. De Leon, Rajesh G. Kavasseri- *"Investigating the Option of Removing the Anti-aliasing Filter From Digital Relays, IEEE TRANSACTIONS ON POWER DELIVERY, VOL. 24, NO. 4, OCTOBER 2009"*

Anti-aliasing filters are used widely in most of the algorithms related to receiving analog inputs from an external output source. This is done to remove any aliasing in the waveform. However, this has inherent disadvantages. In general these are low pass filters so they do introduce a phase delay in the wave. This paper talks about a way of getting rid of such filters as they save significant "cost" and "space." This, as claimed by the authors, is achieved by oversampling the analog waveform.

2.3.5 [IITK12] [1]

Bannerjee P.; Srivastava S.C.- *"A Subspace-Based Dynamic Phasor Estimator for Synchrophasor Application," Instrumentation and Measurement, IEEE Transactions on , vol.61, no.9, pp.2436-2445, Sept. 2012"*

Published recently, this is a classical algorithm which approaches the problems in an analog waveform without the use of anti-aliasing filters using subspace methods. Specifically, the authors talk about two such approaches- 1) The total least square estimation of signal parameters via rotational in-variance and 2) The propagator method. To compensate for the dynamic nature of power system oscillations, a separate approach is proposed. The only drawback of this algorithm is that it requires excessive computation power and requires special techniques for large matrix multiplications. The proposed algorithm, named- ESPRIT+PM utilizes a one cycle window, comprising of 320 samples. It has been tested against off-nominal frequencies, frequency ramps and multimode amplitude and phase oscillations and claims of TVE $< 1\%$ for all the tests.

2.3.6 [HQ11] [6]

Kamwa I.; Pradhan A.K.; Joos G - *"Adaptive Phasor and Frequency-Tracking Schemes for Wide-Area Protection and Control," Power Delivery, IEEE Transactions on , vol.26, no.2, pp.744-753, April 2011"*

This paper presented in 2011 is the first of its kind which addresses the compliance of a PMU output with reference to NASPI and WECC requirements. The requirements are elicited in the next section.

The paper claims for a good response to step and ramp inputs though the frequency requirements are difficult to reach to be implemented on a normal microprocessor. But

implementing in MATLAB, the algorithm gave good outputs which are comparable to the claims in the paper. The algorithm is implemented in multiple steps. At first a “slow frequency tracking" consisting of a recursive DFT is used to estimate the fundamental frequency. There after, depending on the frequency deviation, a "frequency " algorithm or an “adaptive bandpass filtering method is used.” An adaptive bandpass filter is one which can adjust its center frequency in a way to filter out only the component not required.

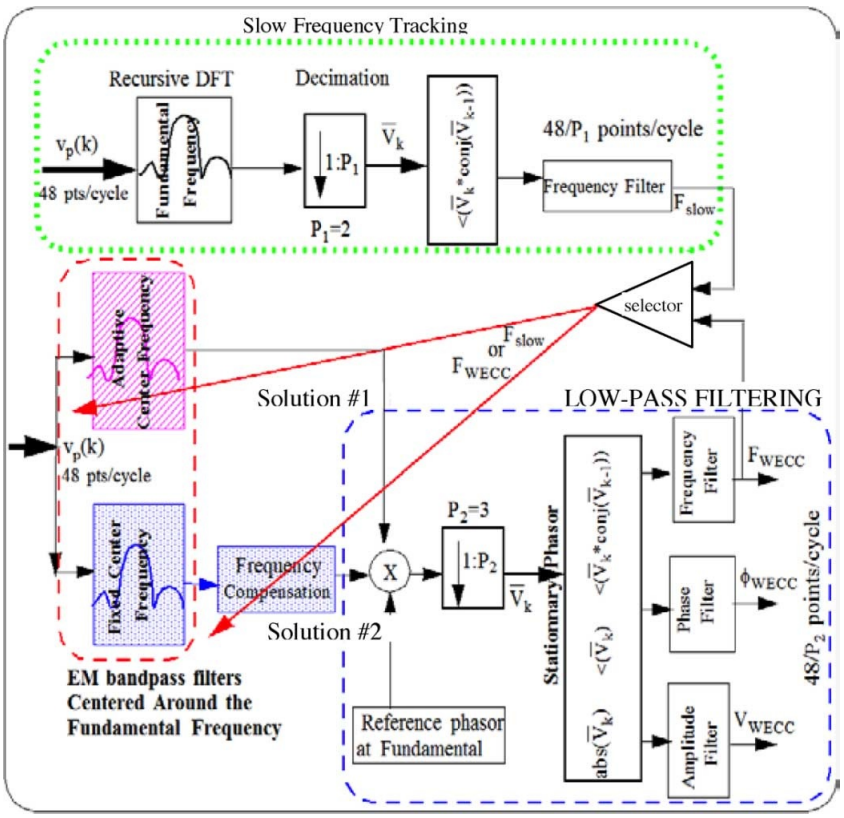

Figure 2.8: Overview of the proposed algorithm

2.3.7 | [HQ12] [7]

Kamwa I.; Pradhan A.K.; Joos G.- *"Compliance Analysis of PMU Algorithms and Devices for Wide-Area Stabilizing Control of Large Power Systems; IEEE Transactions On Power Systems, vol.28, no.2, pp.1766 - 1778, May 2012”,*

IEEE C37.118.2005 was the second IEEE Standard on Synchrophasors and PMU’s in general, released in 2005. In 2011, this standard was updated and was named IEEE C37.118.1-2011. It introduced several new classifications for a PMU and in general talked about the two classed of PMU’s- class -P and -M. This was done to ensure operation over a wide frequency range. The authors talk about an Finite Impulse Response (FIR) based

approach first with adaptive band-pass filtering with no overshoot in either the phase or the amplitude step responses.

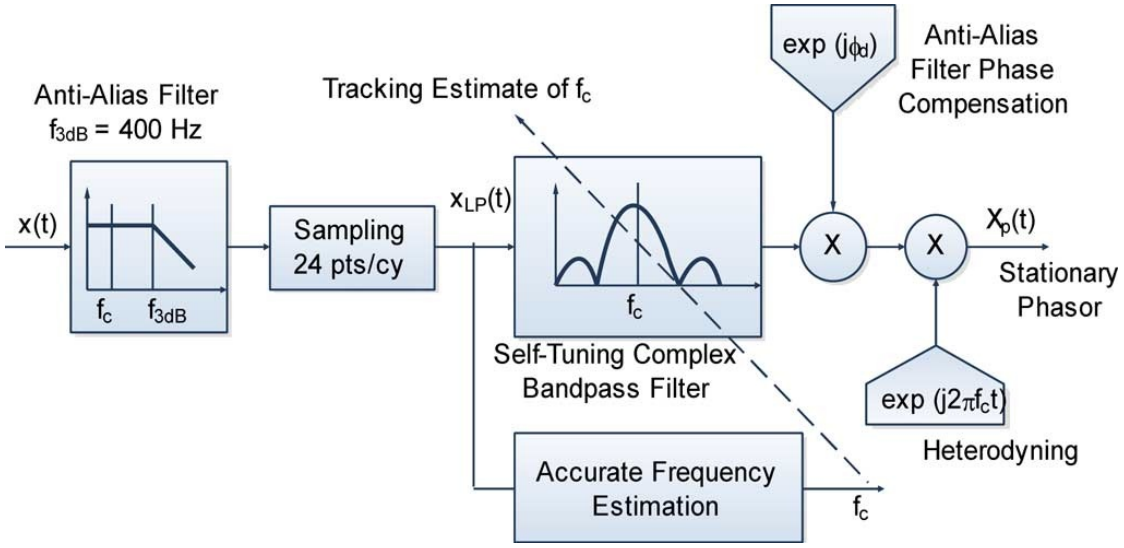

Figure 2.9: FIR based appoach for a Class -M PMU

The paper discusses results from PMU's, with comparison of the same algorithms, from three different vendors and the results don't seem to be consistent with each other in the Class -M. Subjected to the same step excitation's, the output of the three PMU's under test resulted in quite different behaviors. Further, the authors suggest that in the coming updates to the standard, it would be a good idea to mention the appropriate group delay from each of the vendors to compensate for them in the Phasor Data Concentrator.

The paper also brings out the fact that the term TVE is more subjective and cannot be used to characterize or compensate the problems discussed(group delay, transient behaviors) during data alignment.

2.3.8 [Sidhu99] [14]

Tarlochan S. Sidhu- *"Accurate Measurement of Power System Frequency Using a Digital Signal Processing Technique; IEEE Transactions On Instrumentation And Measurement, VOL. 48, NO. 1, February 1999"*,

Since this paper was written in 1999, the concept of P and M-class PMU hadn't come into picture. But this paper describes a Digital Signal Processing based approach to get extremely accurate results when dealing with nominal, off-nominal and near-nominal frequencies in about 25 mSec. This makes it apt to be fit for a P-class PMU. First, orthogonal FIR filters are described to filter any noise/harmonics in the input signal. This means a Real and Imaginary filter are designed separately which generate

waveform which are shifted by 90°. A sampling rate of around 1KHz is used. With these generated waveform, frequency is calculated and the fundamental frequency is updated. This process is done iteratively until the frequency estimated is equal to the fundamental frequency assumed to design the filters. Every iteration, the fundamental frequency is updated and a new set of coefficients are calculated.

The algorithm gives promising results on MATLAB. However, as described, the algorithm needs to design new filters at every iterations. This requires intensive multiplications and additions. Specially if the number of samples taken is big. Hence the authors have suggested to pre-calculate the filter coefficients for a range of 34Hz to 75Hz and store in a tabular form so that they can be directly used instead of generating at run time. This has been a single disadvantage of this algorithm. However, this has been implemented in this thesis on MATLAB and efforts to implement this on ABB MACH3 were also done but were not successful owing to the computations involved on a Real-Time Operating Systems.

2.4 NASPI/WECC REQUIREMENTS FOR AN INDUSTRY COMPLIANT PMU

In efforts to enhance the quality of measurements from Wide-Area Monitoring System (WAMS), the WECC listed the minimal requirements for monitoring applications. These requirements were endorsed by NASPI in 2007. The authors in [4] have summarized the frequency response of a NASPI/WECC compliant PMU as:

- Is -3 dB or greater at 5 Hz;
- Does not exceed -40 dB at frequencies above the Nyquist frequency(a limit of -60 dB is preferred);
- Does not exceed -60 dB at frequencies that are harmonics of the actual power frequency;
- Does not produce excessive ringing in records for step disturbances.

The above requirements are presented in the figure below:

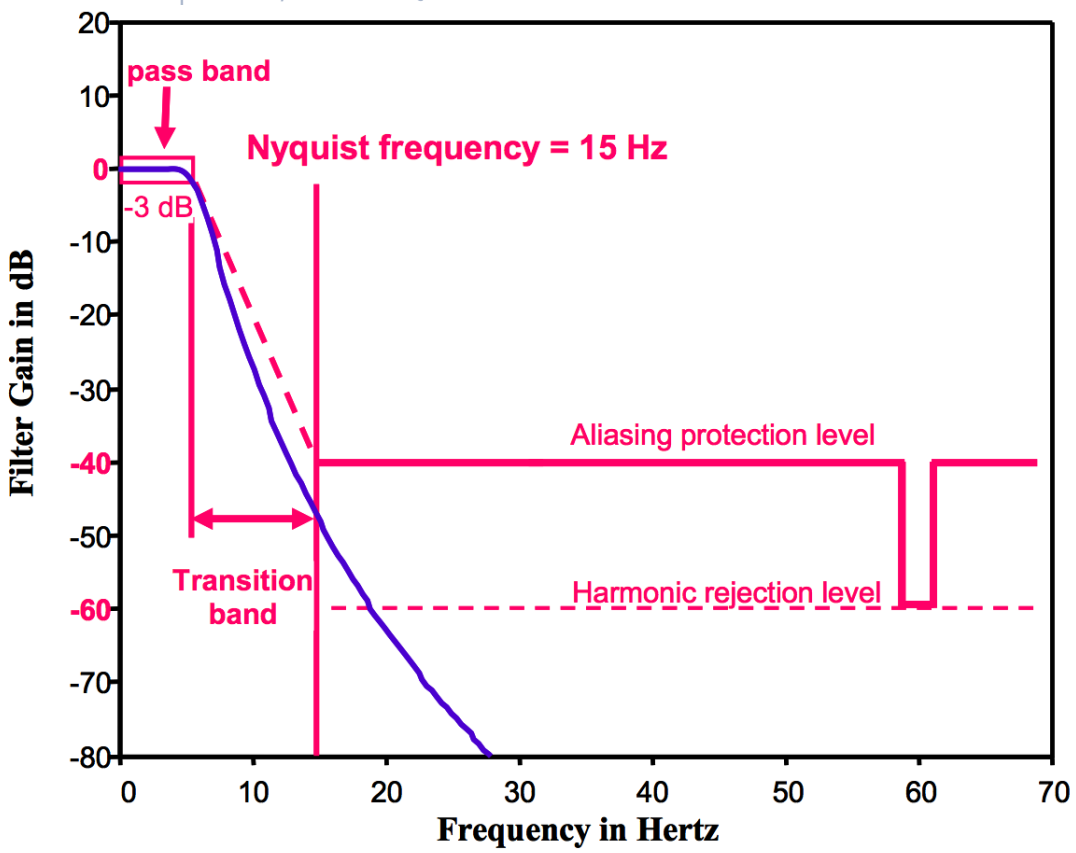

Figure 2.10: WECC Filtering Requirements

With resepct to Figure 2.10 a sixth order Butter-worth filter with a 6-Hz bandwidth versus WECC filtering standard(sample rate = 30 samples per second) is presented.

Further, specifications on data sampling rates, ADC bit resolution and documentation have also been presented. They are summarized below:

- **Sampling Rate:** To make all monitors compatible, a sample rate of 30 sps or multiples of 30 sps is preferred.
- **Numerical Resolution:** The analog-to-digital conversion process should have a bit resolution of 16 or higher. Scaling the signals entering the converter should also assure that 12 to 14 bits are actively used for representation.
- **Measurement Noise:** Noise levels for frequency transducers based on zero-crossing logic are unacceptable.
- **Documentation:** Must be sufficiently detailed that overall quality of the acquisition system can be analyzed. It must also be detailed enough so that the acquired data records can be compensated for attenuation and phase lags introduced by the acquisition system in use.
- **Data Storage:** The monitor(s) store data continuously and retains the data of the last 10 days without any operator intervention.

- **Time Synchronization:** The device is synchronized to UTC to a 100µSec or better.
- **Data Access:** Data access is by network with software for storage, transfer and archiving. Data formats are well defined and reasonable.

However, the above requirements are indicative of sufficient guidelines to meet the WECC needs; but not all necessary in several cases. There are multiple assumptions and there is ample room for engineering judgment.

2.5 POTENTIAL APPLICATIONS

This section covers the potential applications of the implementation. It includes some references to the book from A.G.Phadke [12] and a deeper discussion on the future of PMU's.

PMU's find application both for real time monitoring as well as for off-line network analysis. Depending on the uses case, a P-Class or an M-Class PMU is designed. Wide-area and local monitoring of networks are specific situation awareness applications. Some of the desired monitoring application include:

- **Wide Area Situation Analysis:** One of the key learning statements from cascading outages of the 1996 Western Blackout and the 2003 Eastern Blackouts in the US was that early detection of fault is an important factor in avoiding such events. If PMU's would have been deployed, it would have provided early warning to operators by revealing angular stress and providing visibility of the power oscillations. This would have alarmed the operators and the outages would have been avoided.
- **Voltage and Current Stability Monitoring:** As stated PMU's can be used to measure voltage, frequency and phase angles of a voltage signal in dynamic conditions. It is noteworthy that many systems are designed with voltage limits. In such cases, measuring and controlling the system would be of high importance because it is very easy to cross the stability limits. Voltage instability can occur when there is not adequate reserves of power being generated and demand is high OR it can happen due to a fault in the transmission line when appropriate power doesn't reach from the source to the needed place. PMU's can be deployed in substations to monitor the load/voltage fluctuations and alarm the operator.
- **State Estimation:** By feeding the data(Voltage and Current Measurements) from PMU's directly into the state estimator measurement vector and the Jacobian matrix, the state estimator uses to solve the network. Because PMU data is redundant in ways, state estimator makes quite accurate network solutions based on data both from conventional measurements and PMU's simultaneously.
- **Adaptive Outage Restoration after Blackouts:** It becomes of utmost importance to restore the power soon after blackouts occur, not leading to any damage

to a normal life of thousands of people in the particular area. Even though from previous incidents, there have been many learning statements and many techniques have been designed theoretically, it is hard to implement it in a ready-to-go fashion because the actual reason and actual state of system differs from the estimated earlier theoretically. Since PMU's provide real time measurements, it provides an excellent opportunity to determine restoration strategy. At present, the strategy provided by PMU's is suggested to the operator and he implements it with the suggestions. In future, it is estimated that PMU's will deal with such problems on their own.

- **Differential Protection of Transmission Lines:** A well established protection system in power systems is one based on differential protection schemes. But true differential protection was not possible until PMU's came into picture. This becomes of primary importance for series compensated and tapped lines. Easiest way to perform differential protection is to combine current over a communication channel and compare them.

- **Smart Grid based Island Monitoring:** Applications supporting real time grid operations to provide wide-area visualizations and increased state-estimation.

- **Off line Usages:** Responsive Control Systems that can use the real time data collected from PMU's and take automated control actions on power systems can be designed using PMU data analysis. This can also be used for static system calibration and validation, event analysis, power plant model validation, load characterization, alarm setting and operator training.

3 IMPLEMENTATION

This chapter covers the implementation details of the concept, the algorithms. All the simulations were first carried out in MATLAB® and Simulink®, testing and verifying; then moving to HiDraw® which is ABB's proprietary modeling and designing software for Power System Applications.

3.1 VALIDATION IN MATLAB®

As referred in Figure 1.1 on page 2, a PMU consists of the following blocks:

- Anti-aliasing filter: To avoid any aliasing errors, the IEEE C37.118 standards makes it compulsory to use this filter which band limits the input signals to below half the sampling frequency chosen.
- Analog-to-Digital Converters: This is used to convert the analog inputs to digital for further calculations/processing.
- A GPS Receiver: For time tagging the digital samples.
- A Phasor Processor: To accurately estimate the frequency response of the input waveform, convert it to IEEE C37.118 standard data format.
- Communication Interface: To transmit the data computer over the network.

In the remaining part of this chapter, first a similar interface in MATLAB® will be presented. Then a look at the compliance of this implementation with regards to the standards. Next an overview of the same implementation in ABB HiDraw is given. Finally the results of simulation on HiDraw® are presented.

3.1.1 Implementation in MATLAB/Simulink

Simulink is a graphical programming language developed by MATHWORKS®. It is bundled together with MATLAB releases and is mostly provided by the university working under (in this case KTH). For all the modeling and simulation experiments, Matlab v2013a is used. Three different algorithms were implemented, tested and verified before moving to HiDraw.

Figure 3.1 below shows an implementation of a PMU algorithm. A low pass filter is used as an anti-aliasing filter. Thereafter a sample and hold is used to convert the

waveform into digital. A band-pass filter centered at 50 Hz is used for filtering all the harmonics and DC bias (low-pass filter removes harmonics, High-pass removes any bias). A "discrete 3-phase positive sequence fundamental value" is used to compute the positive sequence phasor from the three-phase input.

For testing purposes, a simulated sine wave is used at 50±2 Hz. Together, a harmonics is inserted right in the beginning for dynamic tests. The system is run at 4 kHz. A low pass Butter-worth filter of order 5 is used as an anti-aliasing filter with pass-band frequency of 2*pi*400 rad/sec. The discrete measurement, positive sequence fundamental value block is developed at Power Systems Simulations Laboratory, IREQ, Hydro-Quebec and performs a Park transformation to estimate the phasors.

As a reference phasor (required for computation of TVE), a plain sine wave is converted to frequency domain using DFT, their phase and magnitude is calculated.

This setup estimates phasors with good accuracy. As a lab setup, phasors calculated by this model were sent out over a specified IP to a Linux machine. On the Linux machine, a pre-implemented IEEE C37.118 library was used to convert the raw phasors in to IEEE C37.118 format. This was finally sent over to a third machine running a Phasor Data Concentrator (PDC), in my case- PMU Connection Tester, available for free at [13]. The setup and results have been reported in an accepted paper at the IEEE Power Energy Society General Meeting in 2013 by the same author(along with others) [8].

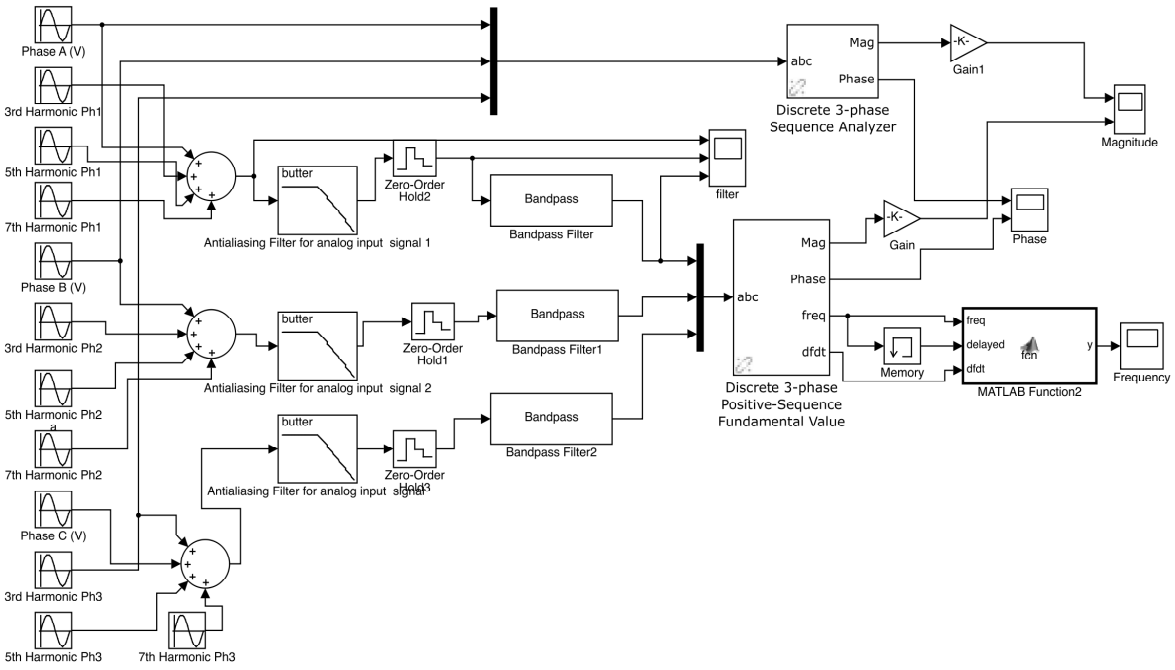

Figure 3.1: MATLAB Implementation of the PMU Algorithm

With reference to Figure 3.1 above, Simulink has blocks made for band-pass filter which is self adaptive in nature. Depending on the harmonics, it chooses the "lowest order", which gives a good output. Anti-aliasing filter for the sake of completing the model was a low pass filter of 4th order. A "sample and hold" block gives the equivalent of an Analog to Digital Converters (ADC) converter. In the end, a 3-Phase Postivie Sequence Block (designed by Hydro-Quebec) is used to compute accurate phasors. Inside the 3-Phase Positive Sequence Analyzer is an implementation of Park Transformation (ABC to dq0 transformation.) This gives fairly accurate results. However, the adaptive band-pass FIR filter designs a filter of the order around 500. While this is possible to implement in MATLAB, this is hard to implement in a hardware or a Real Time Operating System (RTOS).

On running the above setup, the following results are achieved. For all the inputs, a sinusoidal waveform at 50 Hz, magnitude of 8(peak to peak) is used with 3rd, 5th and 7th harmonics inserted right in the beginning and also some bias.

1) The FIR filter response:

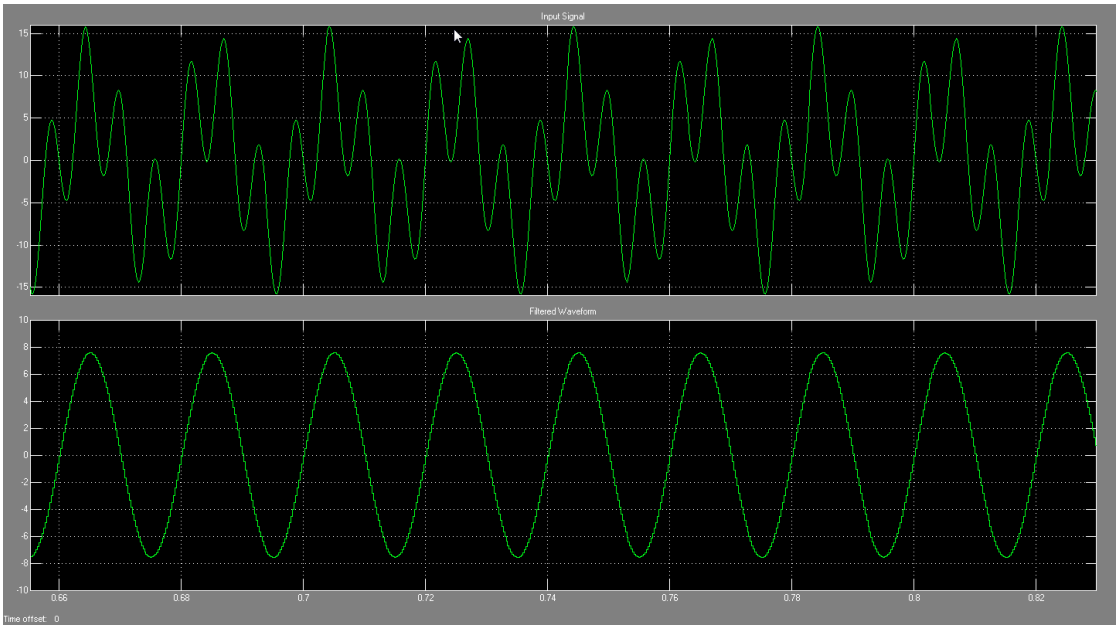

Figure 3.2: Filtered Output

With reference to Figure 3.2 above, the input is a sinusoidal waveform of magnitude 8 peak-to-peak with 3rd, 5th and 7th harmonics inserted in the beginning and at a frequency of 50Hz. The FIR filters being self adaptive in nature, removes all the noise and gives a plane sinusoidal signal after filtering. The signal is also digitized because of sample and hold block. However, a small element of delay is added because of the intensive filtering to the input. This adds up in the error calculation as given by Equation 2.1 on page 8.

2) The Frequency measured:

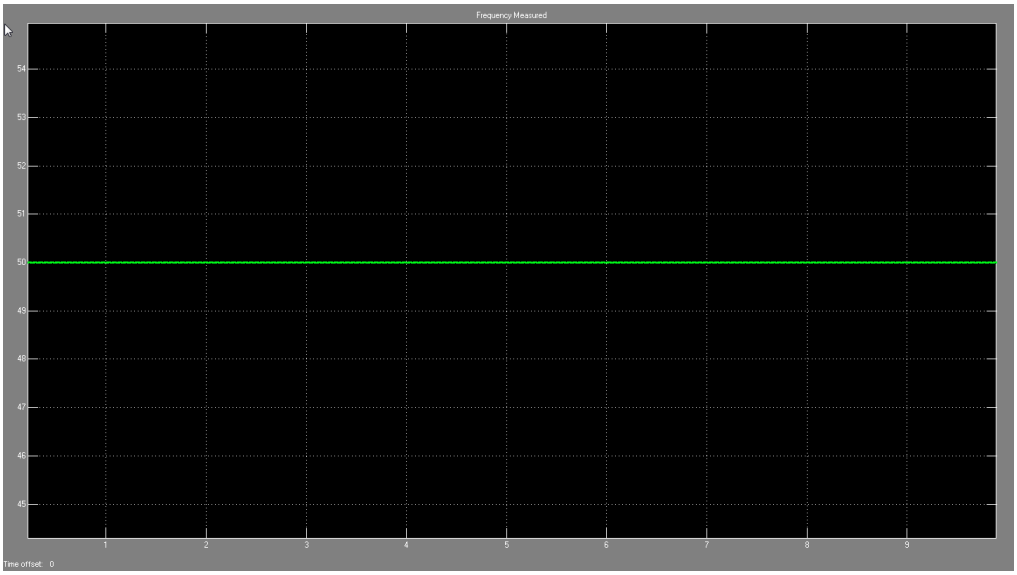

Figure 3.3: Filter Frequency Measured

With reference to Figure 3.3, the frequency is accurately measured as 50Hz. This infers that the model is insensitive to any noise/bias in the input.

3) The ROCOF:

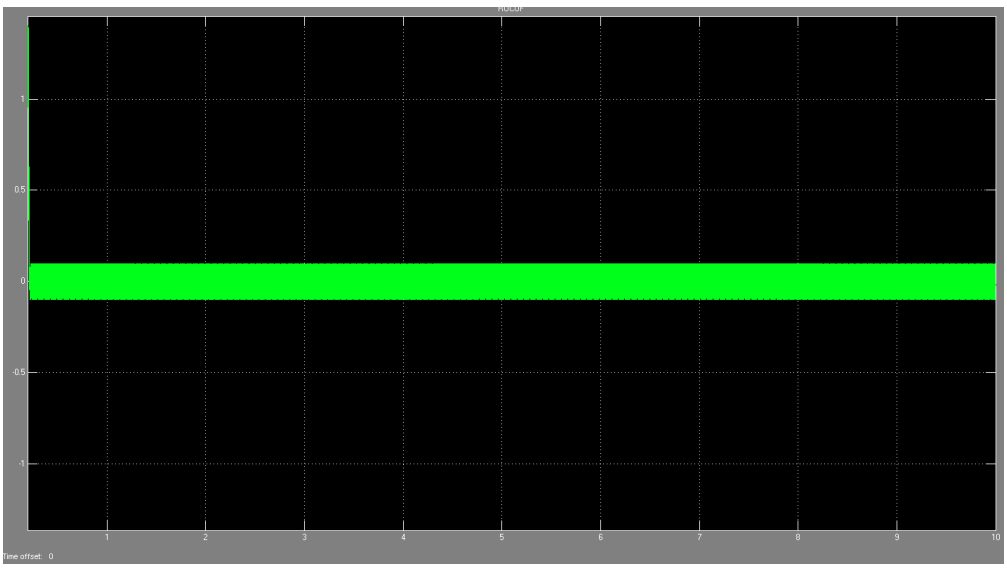

Figure 3.4: ROCOF

With reference to Figure 3.4, the ROCOF is as small as ± 0.5 Hz. This is acceptable

for an P-class PMU.

4) The Magnitude Measured:

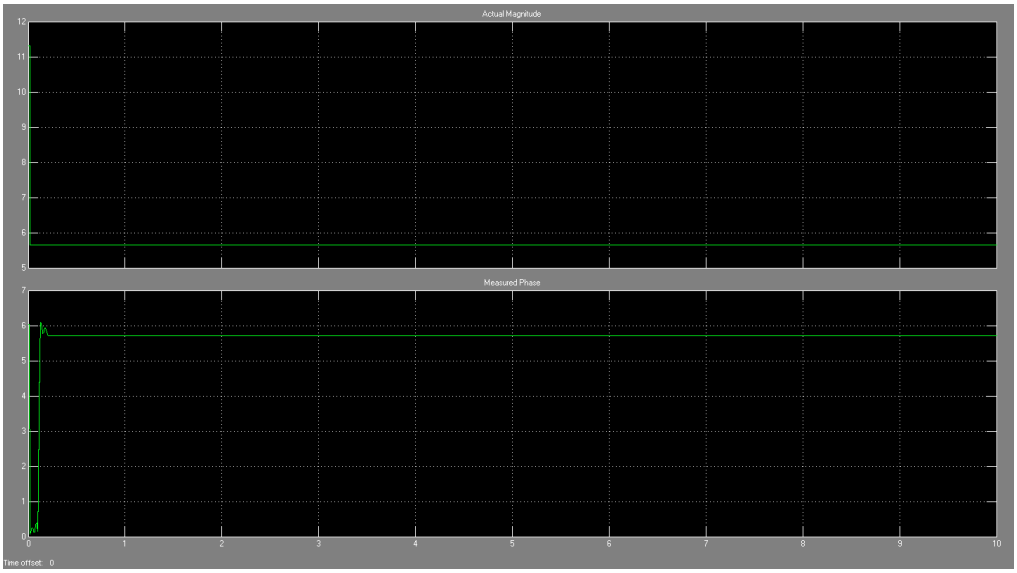

Figure 3.5: Magnitude Measured

The magnitude (RMS value) measured is equal to the actual magnitude/sqrt(2). The actual magnitude is 8(peak-to-peak)

5) The Phase angle measured is:

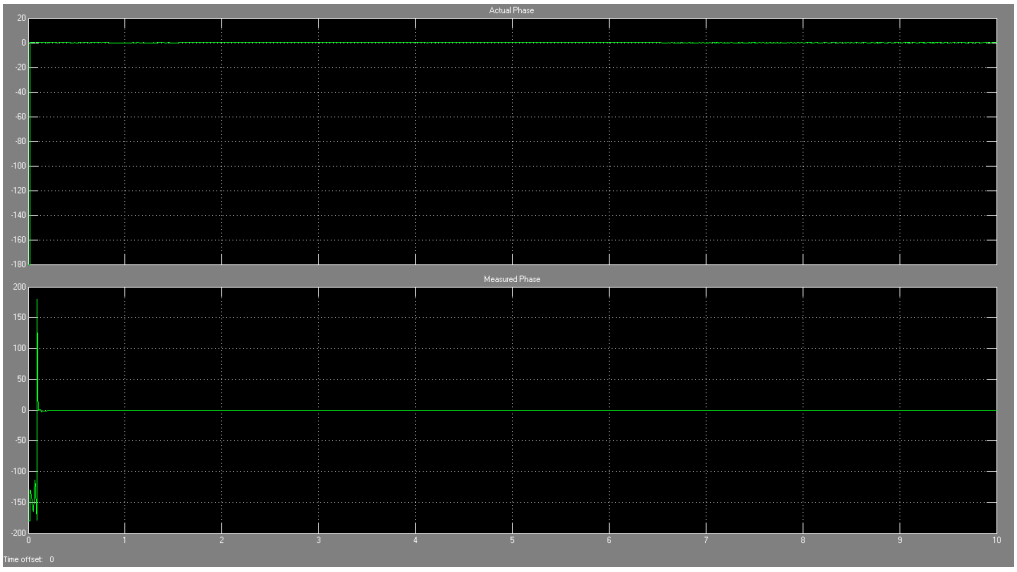

Figure 3.6: Phase Measured

The phase angle measured is 0° as seen in Figure 3.6. To be noted, that the phase measured will always be 0° if the reference frequency is equal to the fundamental frequency. However, at off-nominal, near-nominal frequencies, the phase angle starts to fluctuate between +180° and -180°.

6) Finally the Total Vector Error:

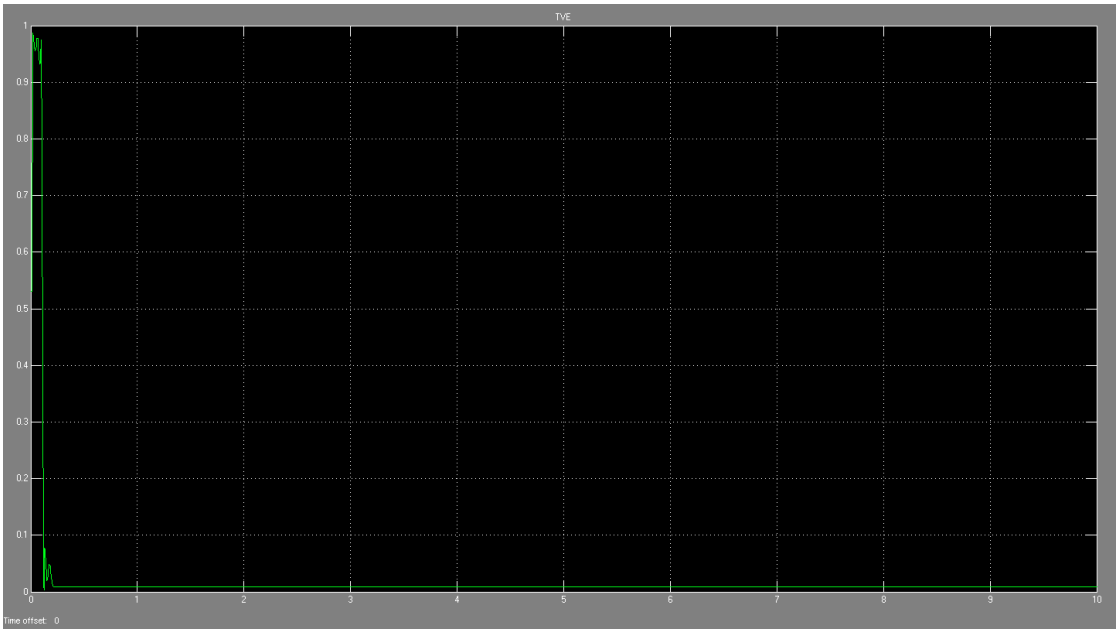

Figure 3.7: Total Vector Error(TVE)

TVE is a measure of the difference between the measurements made by the actual PMU device in test and the theoretical value calculated. The mathematical equation to calculate TVE is given by Equation 2.1 on page 8. For compliance, TVE should be less than 1. In the above figure (3.7), TVE is less than 1.

3.2 IMPLEMENTATION IN HIDRAW

3.2.1 Software

ABB Power Systems uses HiDraw- a fully graphical programming language as an application programming interface, which is similar to Mathworks Simulink in terms of look and feel; and HiBug, a graphical debugger. Both HiDraw and HiBug are designed for Windows environment. Owing to the similarity to Simulink, HiDraw opts the click-drag-and-drop feature for designing and modeling. Depending on the target hardware, there are specific libraries with the well-known blocks available. These blocks/symbols are placed on a drawing (a “drawing” is an equivalent of “model” in Simulink or “vi” in LabView.) These blocks are then connected by using “signals”

(which are equivalent of "connectors" in Simulink.) When HiDraw Drawing is compiled, code is generated from the schematic which is either in a assembly language or a high level language(like ANSI standard C, C++, PL/M or FORTRAN). As stated every hardware in the MACH™(Modular Advanced Control for HVDC) family has a library or pre-defined and pre-assembled blocks. Blocks which are not available can be designed in HiDraw32 (where HiDraw drawings can be modified) or imported from a different existing library. This coding is done in HDF (HighDraw Definition File). Hdf-file is where the source code for HiDraw symbols is stored. The generated code is used directly for transient studies. Models can also be used to study this way.

The HiDraw generated code is a well-documented C/C++ application code, which is easily understandable. The source code in itself is in the form of functional blocks. Each block has a different block of code. For debugging purposes in the target hardware, HiBug, which is a fully graphical debugger, is used. HiBug is integrated in HiDraw Studio. When connected to the Hardware in real time, simple double clicking the signal shows the current value. Values can even be changed during run time. Owing to its user-friendliness, HiBug is an excellent maintenance tool as well.

MACH is a fully computerized control and protection system. The main features of the MACH are the high degree of functional integration and the open systems interface approach. The open systems strategy is reflected both in the use of industrial standard serial and parallel communication buses, as well as in the use of standard formats for all collected data (such as events, alarms and disturbance data). This makes it possible to take advantage of the latest achievements in electronic engineering, thus ensuring timely development of more advanced functions and enhancement of performance for a long time to come. Integrated with the MACH control and protection equipment is the Station Control and Monitoring (SCM) system. Workstations (PC's) are interconnected by an Ethernet local area network. A field bus network builds up the distributed system for remote I/O, for control as well as for process interfacing with the SCM system. Converter control and protection principles are based on the well-proven ABB HVDC control system that is now in operation in over 80 converters around the world. The Base Design provides a complete well proven platform for the implementation of control and protection for an HVDC project. Experience from previous projects is fed back to ensure continuous improvement [16].

3.2.2 Hardware

ABB uses its proprietary Hardware, named PS700 for all its control applications and uses PS741 for Voltage Measurements whereas PS745 for Current Measurements.

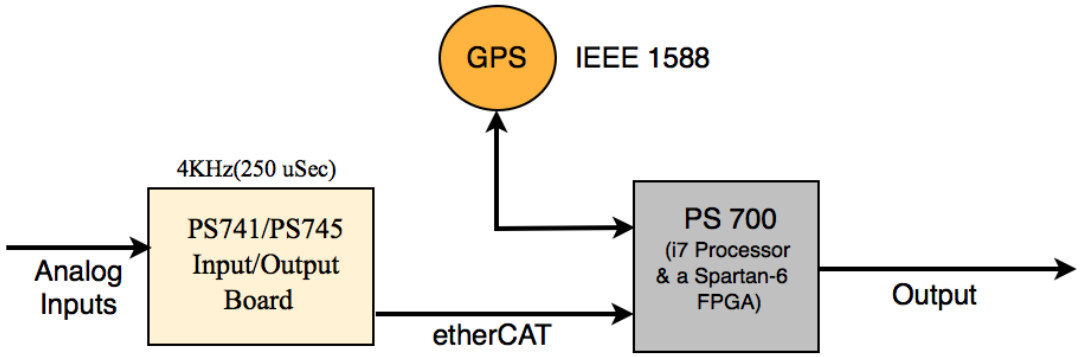

Figure 3.8: Overview of Setup at ABB

Inside of PS700 is an Intel™ i7 processor coupled with one(or sometimes two) Spartan 6 FPGA's. The FPGA performs the input/output tasks and communicates with external devices. This communication takes place over EtherCAT. A generalized setup is shown in 3.8.

EtherCAT is short form for "Ethernet for Control Automation Technologies", the ethernet based field bus system. EtherCAT overcomes the bottlenecks of the traditional Ethernet protocol, for e.g.- usable data rate being low. With EtherCAT, real time applications can overcome the problems of high jitters and provide short data update time. Unlike Ethernet, the packets are not received, processed in EtherCAT. It follows a Master-Slave architecture where the Slave reads the data addressed to it while the telegram passes through it. The theoretical delay in EtherCat is estimated to be lower than 1 µSec. With multiple nodes, hardly one frame delay is noted. Data transfer needn't be in any physical order in the network and addressing can be independent. The master device takes care of implementation of broadcast or multicast between slaves. It also supports UDP/IP datagrams for IP routing.

An added advantage of using EtherCAT is the direct use for time synchronization. EtherCAT uses a distribute clock mechanism to synchronize the slaves. Since the jitter is less than 1 µSec, it automatically becomes compatible with PTP (Precision Time Protocol) and IEEE 1588 Protocol Standard. No extra hardware is needed to implement this and is basically done in software on any of the Ethernet MAC, which dos not even require a dedicated communication processor.

As mentioned, the implementation is done using master-slave architecture. The master broadcasts and whichever slaves latch the value of the internal clock twice (once when received and when returned, EtherCAT has a ring topology), becomes the first slave and the reference clock; forcing the other slaves to set their clocks appropriately with the delay calculated. The master from the latched values calculates delay for each slave. The master or the slave keeps broadcasting to take care of the speed difference between the clocks of each of the slaves. Each slave then adjusts their internal clock

through some mechanism.

The PS741/745 I/O boards are used for taking analog inputs. The IO boards perform the task of ADC converter and anti-aliasing if required. The ADC sample inputs every 2 μSec with every sample being 16 bits wide and signed. Possible A/D values ranges from decimal -32768 to 32767. Input voltage ranges from -2.5V to 2.5V. Anti-aliasing filters are implemented as cascaded IIR filters. All the filters support upto 8 parallel channels with identical filter parameters, which can be configured from the DSP application.

The first filter is intended to filter the input before it's passed to the DSP. The second filter is intended to filter the input before it is send over the eTDM. The third filter is a low-pass filter of 4th order, with a cut-off frequency of 500 Hz and runs in the fastest task level, executed every 100 μSec. The input samples are taken from the low-pass filters of the FPGA, sitting in the PS741/745, which samples at 5 KHz. So to avoid aligning the filter is set to run at 10 KHz or faster. The data type used is 32-bit floating point.

3.2.3 | Implementation

After testing the Simulink model (Figure 3.1), an equivalent is implemented in ABB MACH3 controller. The following figure shows the implementation in HiDraw:

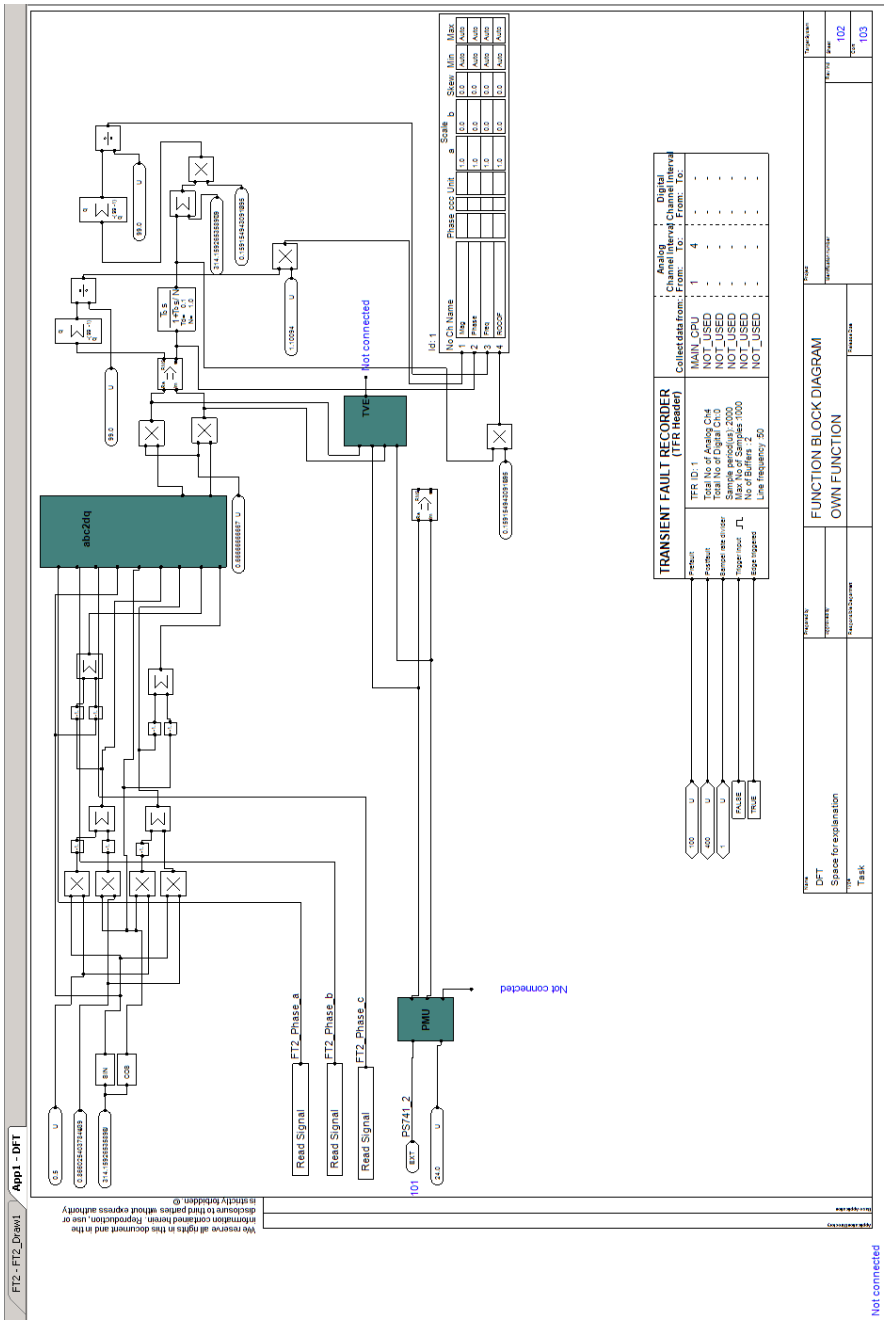

Figure 3.9: Equivalent Model in HiDraw

Filtering and A/D Conversion is done in a separate application, explained next. The filtered input is received and the phasor calculations are made in this application,

which runs at 1KHz.

It is important to note that, that filters are much more efficient at a high sampling rate so the filters had to be run in a "Fast Task" in HiDraw which runs at 4KHz. Also the filters in case of HiDraw are Infinite Impulse Response(Infinite Impulse Response (IIR)) filters contrary to Finite Impulse Response (FIR) filters in the literature survey and original model in MATLAB. This is owing to the fact that for a given cut-off frequency and sampling rate, IIR filters have much lower order compared to FIR filters. A cascaded filter approach is implied. A low pass filters of order 16 is used for any harmonics. A 4^{th} order high-pass filter is used to remove DC bias. The results are presented in the next section. Signals can also be transferred from one application to other in HiDraw.

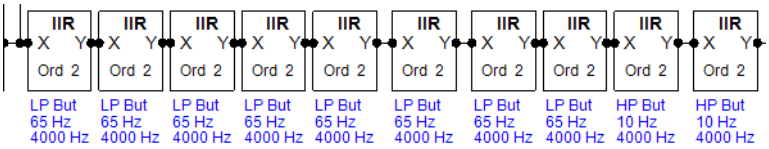

Figure 3.10: Filter Implementation in HiDraw

Chain of IIR filters of order 2 is used to filter out any noise and DC bias present in the input signal presented above.

The blocks in green are the blocks that had to be coded, as they were not available in the library.

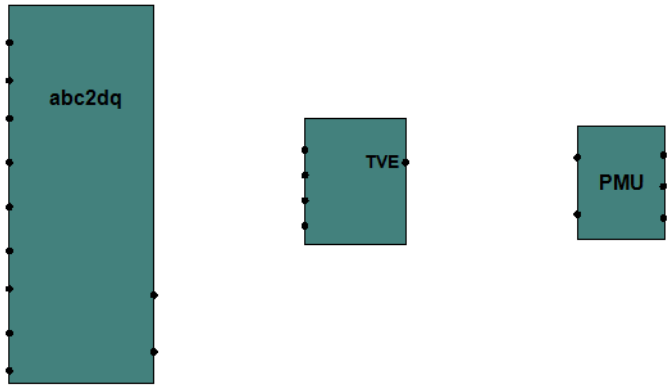

Figure 3.11: Blocks unavailable in HiDraw were coded to get their functionality

With reference to Figure 3.11, the ABC2DQ block performs the park transformation of a three phase waveform. The TVE blocks calculates the Total Vector Error between the actual and observed outputs. The inputs required are the real and imaginary components of the actual and observed signals. The PMU blocks implements the algorithm computing the phasors.

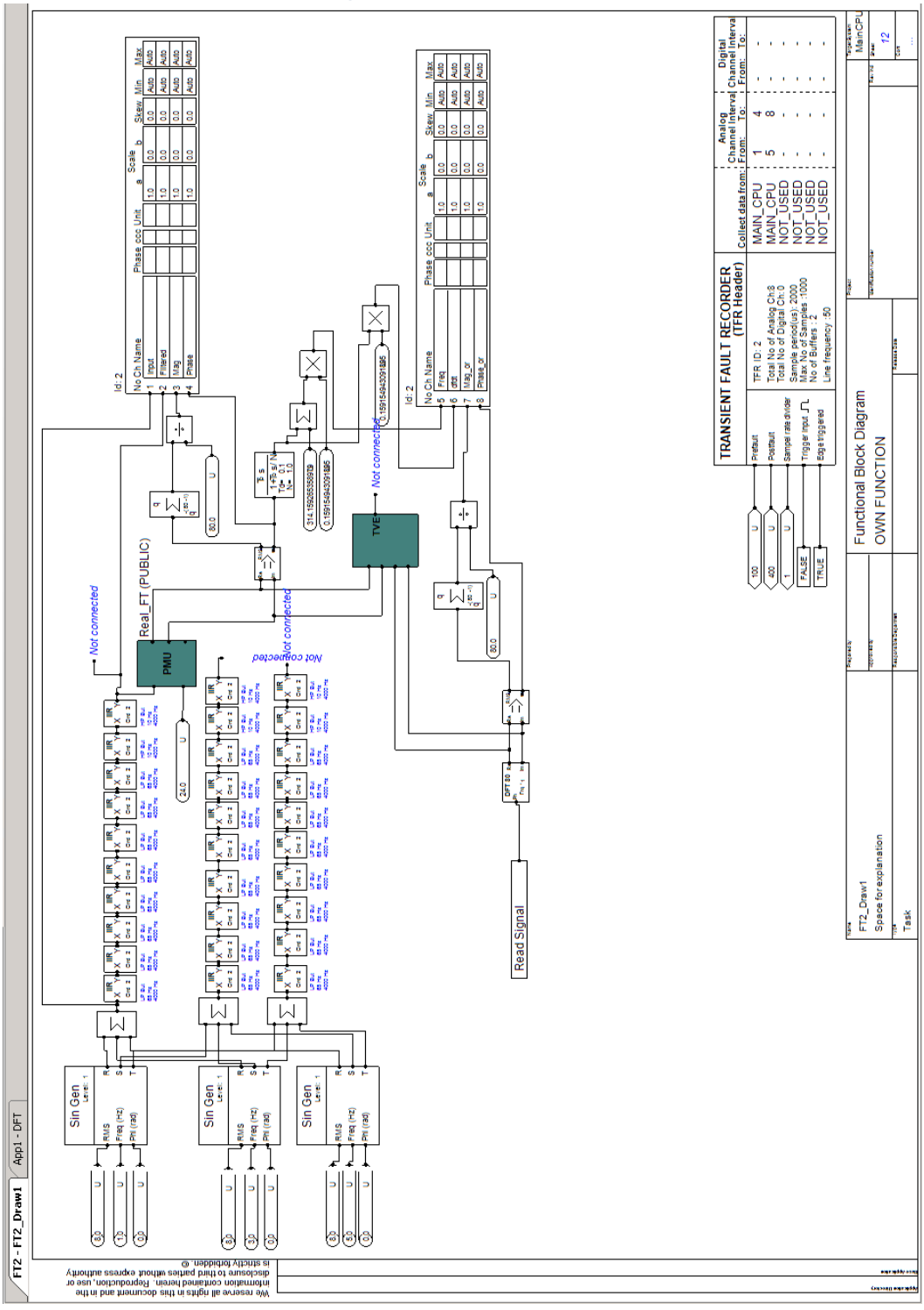

Figure 3.12: All the filtering is done in a "Fast Task" which runs at 4KHz.

Figure 3.12 shows the FastTask in HiDraw which does all the filtering. The coefficients of the filters are calculated from MATLAB and copied into HiDraw.

3.2.4 Time Synchronization

Time synchronization in MACH is built right into the system. The "TSS" block available in the library does this task. The PS700 computer waits for 60 sec on start for the GPS signal. If it receives the GPS pulse, the GPS device becomes the master and the PS700 starts acting as slave. If it is not available, the slave with the latest time assumes itself as master and the remaining PS700 act as slaves.

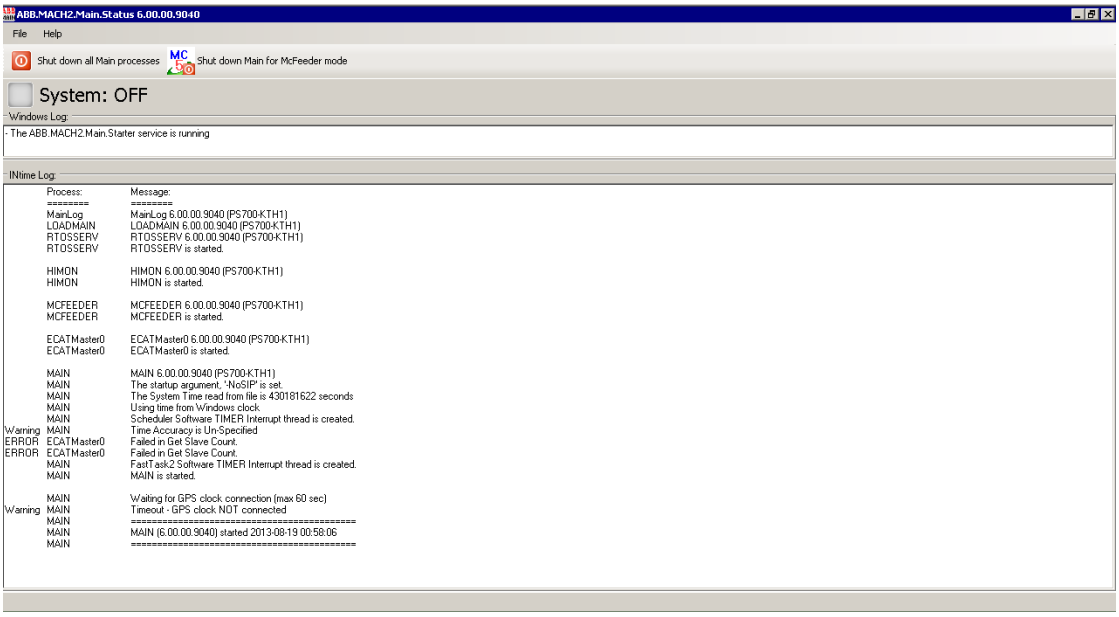

Figure 3.13: GPS Functionality in MACH3.

The above figure shows the start up window of the PS700 running Windows InTime. If everything goes well, a message is displayed in the end: "Main Started" otherwise errors and exceptions are thrown at this point. Debugging (using HiBug) can't be done until the main starts. If GPS is connected, the PS700 starts acting as a slave and the GPS becomes the master. A display message is shown "Time Set by GPS Clock." If GPS is not connected(like in above figure), the device waits for 60 seconds and gives a warning message "Time Accuracy is unspecified."

3.2.5 Results

First a look at the output from the filters, which run at 4 KHz. The input is a simulated waveform with 3rd and 5th harmonics inserted right in the beginning at 50 Hz. and To record the output, Transient Fault Recorder (TFR)(Transient Fault Recorders) are used. TFR is a Matlab based graphical tool to record a number of samples and analyze it in form of waveform. It can also export the data to Matlab as a .mat file. After filtering, Discrete Fourier Transform is applied to the waveform. This also will give us insights into why a better algorithm is needed.

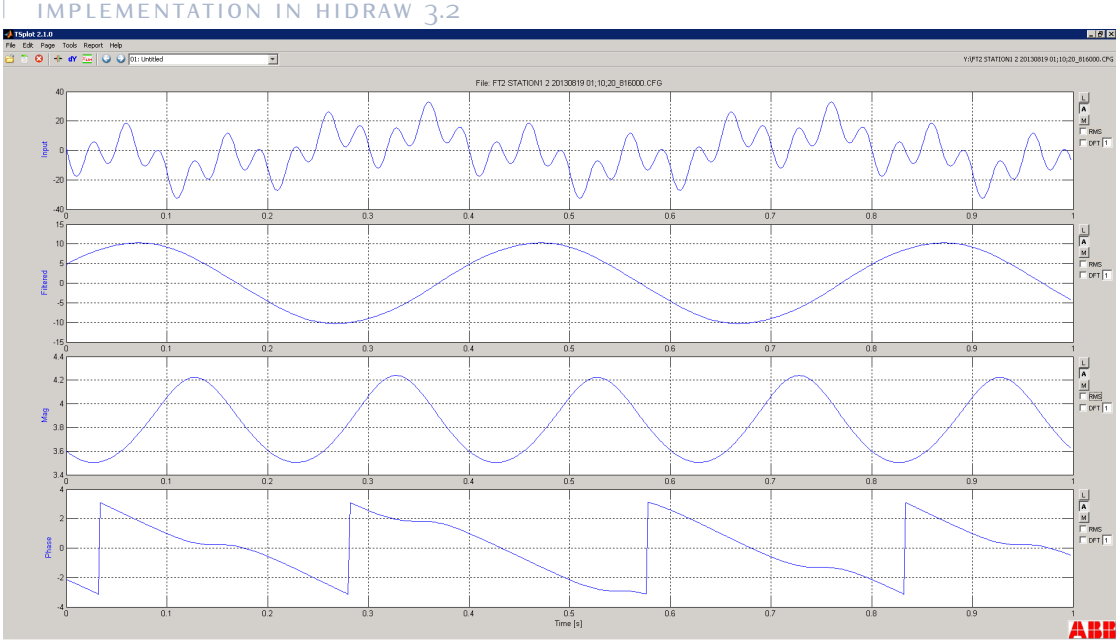

Figure 3.14: Filtered Output

With reference to Figure 3.14, after receiving the output from filters; magnitude and phase is calculated by performing DFT. As seen, in presence of harmonics and bias, DFT can not accurately measure the magnitude and phase. Next the results from the algorithms, which compute the positive sequence fundamental value from the three, phase by performing Park Transformation. The input remains the same as in above case. The input remains the same as in above case.

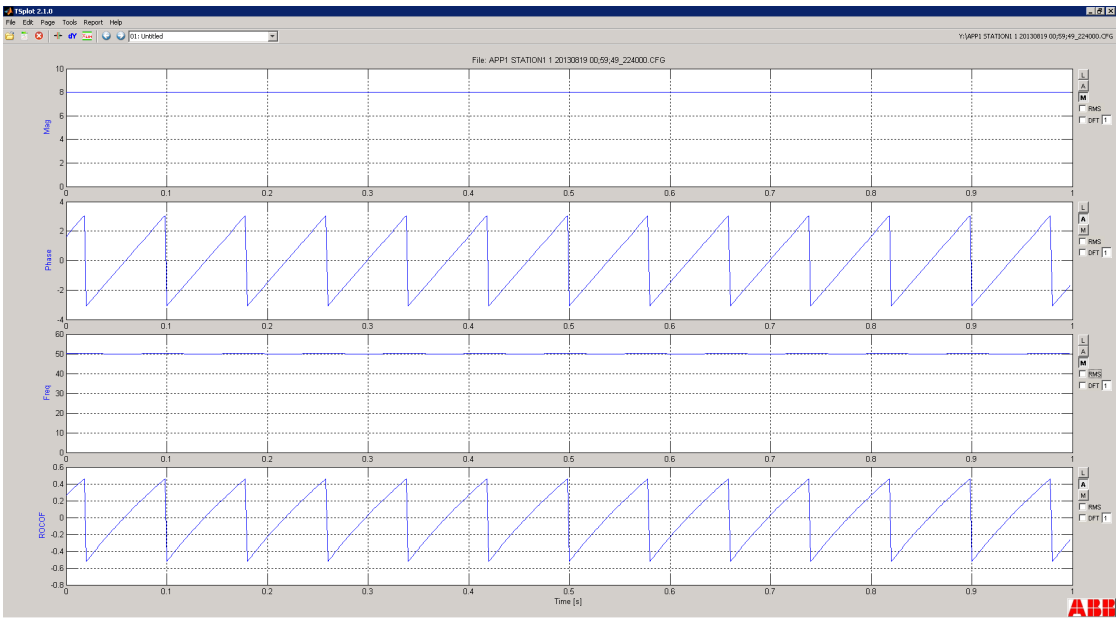

Figure 3.15: Output from an enhanced algorithm

As seen in Figure 3.15, the magnitude measured is measured accurately as 8(peak to peak), the phase is smooth and fluctuating between $\pm\pi$; frequency is accurately estimated at 50Hz and the ROCOF remains relatively very small.

It can be seen that the accuracy of this algorithms is much better than the traditional DFT algorithm. Further, this approach is very low on computation power. The results obtained from this algorithm comply with the M-class of PMU algorithms because the algorithm measures accurately the frequency and phase only for 50 ± 2 Hz. The ROCOF is also very small. The reported frequencies are in Hertz, phase in radians and magnitude measured is peak-to-peak. Unfortunately TVE calculation wasn't possible because a reference PMU is needed to compute TVE from. However, for the sake of completeness, TVE was calculated between the two algorithms described above(Traditional DFT and Park Transformation based) and is plotted below.

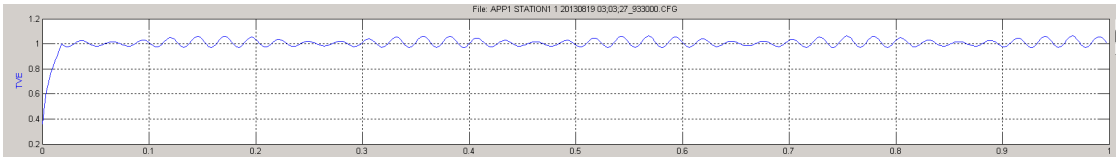

Figure 3.16: TVE Calculations in HiDraw

3.2.6 Comparison and Limitations of Algorithm implemented

As seen in earlier chapters, the literatures discuss the use of FIR filters whereas the implementation has an IIR filters. This is due to the fact that for a given sampling rate and cut-off, FIR needs a much higher order to implement the filter compared to IIR. The figures attached below demonstrate this:

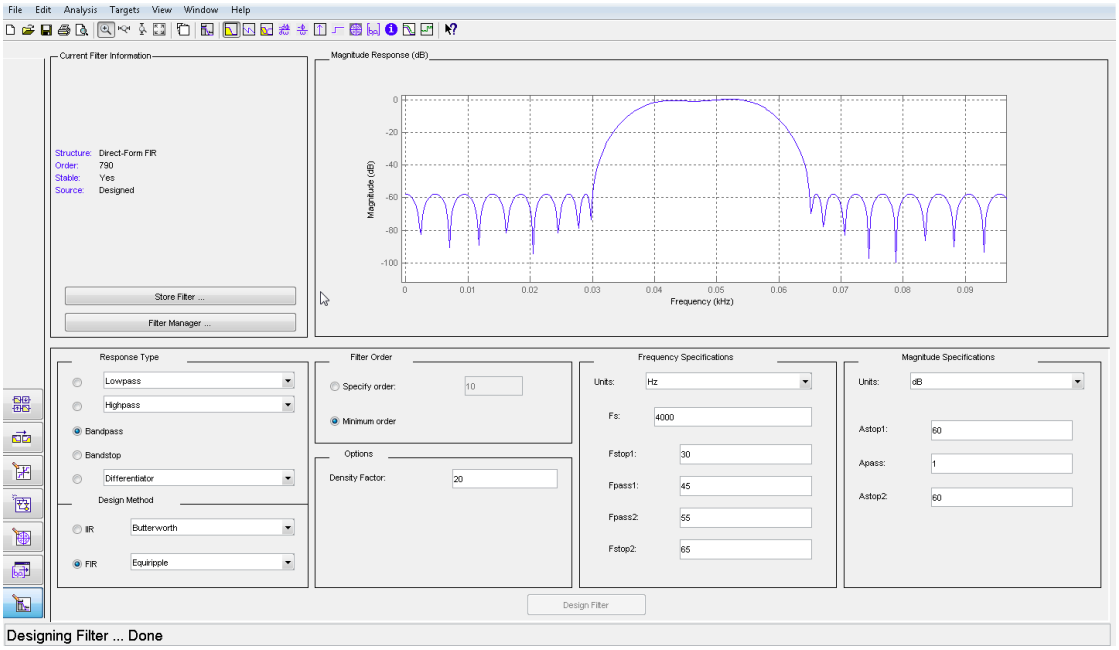

Figure 3.17: FIR Filter Design usign Matlab FDA tool

An FIR filter designed using the Filter Design And Analysis tools in MATLAB generated a 760th order filter with the specifications given. This is impractical while implementing in hardware.

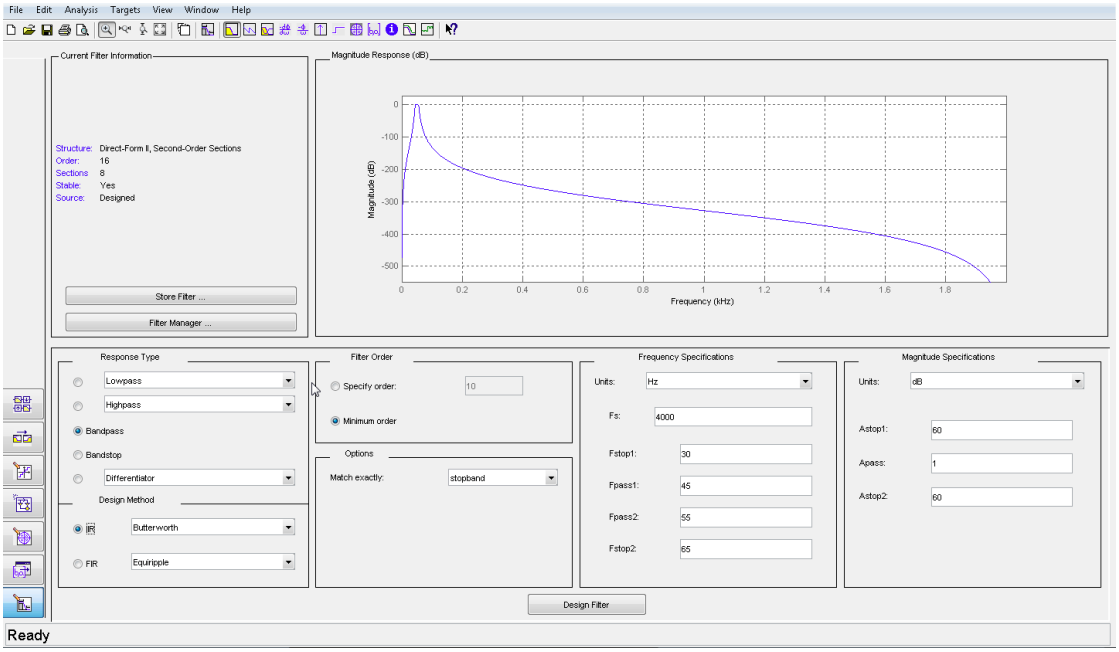

Figure 3.18: IIR Filter Design using Matlab FDA tool

Whereas for an IIR filter like above, the order is just below 20 which is quite practical.

Hence an IIR filter was implemented. This in turn gives rise to "phase delay" and "group delay" but there are approaches to solve this problem. However, it is very hard to implement a filter above order 100.

Further, the second algorithm under study- "Sidhu(link to literature study pending)" implements a very robust algorithm which is a very good implementation of a M-Class PMU. However, it takes significant amount of memory and processing in real time and needs specialized hardware's to run. It in fact implements a complex array multiplication of 64*64 size and applies a pseudo-inverse function of the same matrix, of size 64. On running this, the Real Time Operating System started throwing exceptions. Because all the calculations could not be finished within the interrupt time (which was lowest at 48 mSec).

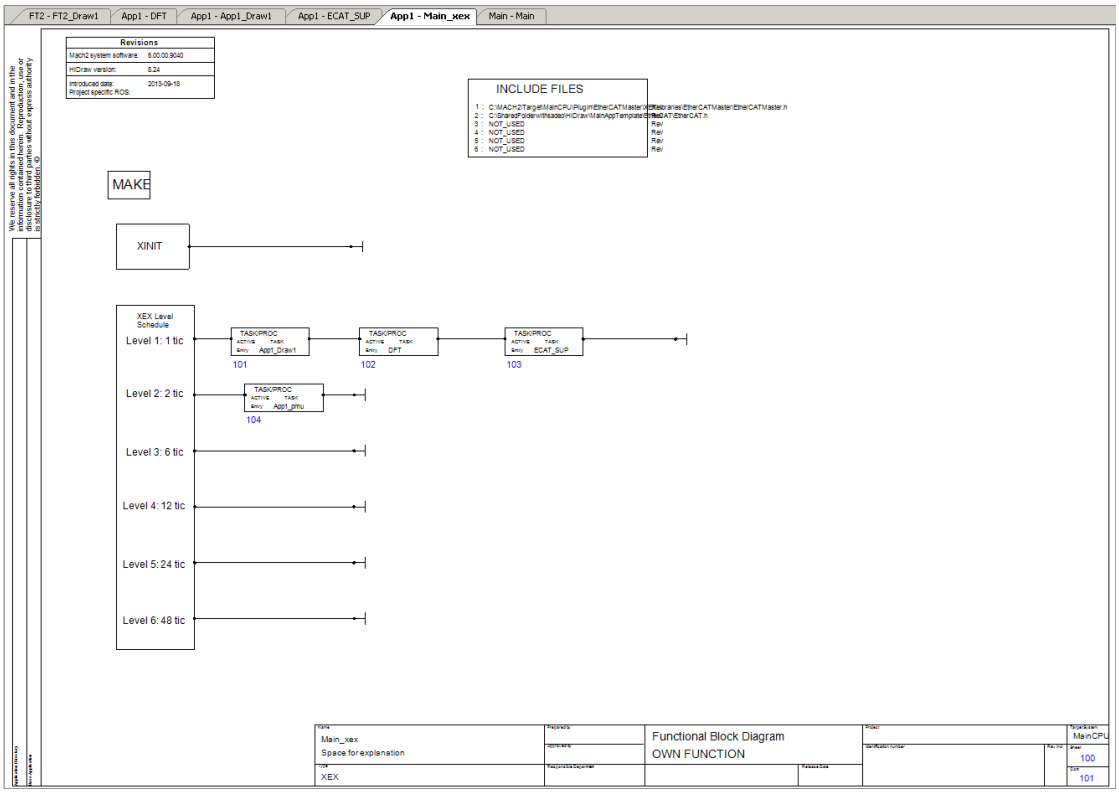

Figure 3.19: Schedulers in Windows InTime RTOS

Figure 3.19 shows how are schedulers implemented in HiDraw.

Running the Algorithms on Matlab, the phasors were streamed on self-made IEEE C37.118 library in Linux. A reporting rate of 30frames/sec was achievable in PMU Connection Tester. (PMU Connection Tester, administered by the Grid Protection Alliance (GPA) verifies that a data stream from synchrophasor measurement device is

being successfully received. The PMU connection tester supports IEEE C37.118.2005/2011 data format.) However this has been left for future implementation in ABB MACH controller due to time and resource constraints.

4 CONCLUSION AND FUTURE WORK

This final chapter concludes the results obtained from the thesis and also the vast scope for future work in this area.

While there is an enormous amount of research and work going on in this field, implementation and application comes to a halt when it comes to software-hardware integration. Many good algorithms take up relatively a very high amount of memory and processing for phasor computation. Making a custom device for such algorithms is one possible solution to this problem but it adds costs to the development.

At the same time, the P-Class and M-Class PMU's are sold separately with two different types of algorithm running on them. The M-Class algorithm focuses on accurate measurements and is not so fast. They can have lower reporting rates. Whereas P-Class PMU's are meant for protection purposes and need to be fast enough with a slight trade off possible with accuracy. An implementation focusing on P cum M-Class PMU would be the best solution, both for measurement and protection.

The delay factors added due to the use of IIR filters also need to be compensated, either in the algorithm design or at the PDC end. At the algorithm end, an anti-filter or an all-pass filter can be used to compensate for the delay. However, this may take additional processing power. All pass filters are cascaded with the normal filters of the same order in the model and they act as compensating filters.

Owing to the fact that there is a lot of communication involved in a PMU and from a PMU, it becomes another important issue to make sure that data being communicated is over a trusted interface and the communication is secure. Research's in this field include implementation of SNORT algorithms, which is an open source network intrusion prevention and detection system. This is yet another area of active research in nascent stage.

BIBLIOGRAPHY

[1] P. Banerjee and S. Srivastava. "A Subspace-Based Dynamic Phasor Estimator for Synchrophasor Application". In: *Instrumentation and Measurement, IEEE Transactions on* 61.9 (Sept. 2012), pp. 2436–2445 (cit. on p. 17).

[2] C. Chen. "A Phasor Estimator for Synchronization between Power Grid and Distributed Generation System". In: *IEEE Transactions on Industrial Electronics* (2011), p. 1 (cit. on p. 1).

[3] *Department of Defense (DoD), US Government*. Available on-line: http://www.gps.gov/ (cit. on p. 13).

[4] Z. Huang et al. "Laboratory Performance Evaluation Report of SEL 421 Phasor Measurement Unit". In: *Pacific Northwest National Laboratory, The U.S. Department of Energy* (Dec. 2007), pp. 41–43 (cit. on p. 20).

[5] "IEEE standard for synchrophasors data transfer for power systems, C37.118.2-2011". In: *IEEE Std C37.118.2-2011 (Revision of IEEE Std 1344-1995)* (2011) (cit. on p. 1).

[6] I. Kamwa, A. Pradhan, and G. Joos. "Adaptive Phasor and Frequency-Tracking Schemes for Wide-Area Protection and Control". In: *IEEE Transactions on Power Delivery* 26.2 (Apr. 2011), pp. 744–753 (cit. on pp. 1, 17).

[7] I. Kamwa, A. Pradhan, and G. Joos. "Compliance Analysis of PMU Algorithms and Devices for Wide-Area Stabilizing Control of Large Power Systems". In: *IEEE TRANSACTIONS ON POWER SYSTEMS*. Sept. 2012 (cit. on p. 18).

[8] Z. Kun et al. "Test platform for synchrophasor based wide-area monitoring and control applications". In: *Power and Energy Society General Meeting (PES)* (July 2013), pp. 1–5 (cit. on p. 26).

[9] K. Martin. "Exploring the IEEE standard C37.118-2005 synchphasors for power systems". In: *IEEE Transactions on Power Delivery* (Oct. 2008) (cit. on p. 1).

[10] M. Paolone, A. Borghetti, and C. A. Nucci. "A Synchrophasor Estimation Algorithm For The Monitoring Of Active Distributions Networks In Steady State And Transient Conditions". In: *17th Power Systems Computation Conference*. Stockholm, Sweden, Aug. 2011, pp. 1–8 (cit. on pp. 1, 15).

[11] A. G. Phadke, J. S. Thorp, and M. G. Adamiak. "A New Measurement Techniques for Tracking Voltage Phasors, Local System Frequency, and Rate of Change Frequency". In: *IEEE Transactions on Power Apparatus and Systems* 102.5 (May 1983), pp. 1025–1038 (cit. on p. 14).

[12] A. Phadke and J. Thorp. *Synchronized Phasor Measurements and Their Applications*. New York: Springer, 2008. ISBN: 978-0-387-76535-8 (cit. on p. 22).

[13] PMU Connection Tester. [Online]. Available: http://pmuconnectiontester.codeplex.com/ (cit. on p. 26).

[14] T. S. Sidhu. "Accurate Measurement of Power System Frequency Using a Digital Signal Processing Technique". In: *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT* 38.1 (Feb. 1999), pp. 75–81 (cit. on p. 19).

[15] B. Sukumar M, L. Phillip De, and K. Rajesh G. "Investigating the Option of Removing the Antialiasing Filter From Digital Relays". In: 24 (Oct. 2009), pp. 1864–1868 (cit. on p. 17).

[16] K. Torbjörn, H. Mats, C. Lars, and B. Hans. *Modern Control And Protection System For HVDC*. [Online]. Available: http://www05.abb.com/ (cit. on p. 31).

[17] M. Wang and Y. Sun. "A practical, precise method for frequency tracking and phasor estimation". In: *Power Delivery, IEEE Transactions on* 19.4 (Oct. 2004), pp. 1547–1552 (cit. on pp. 1, 16).