High Accuracy Real-time GPS Synchronized Frequency Measurement Device for Wide-area Power Grid Monitoring

Chunchun Xu

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
In
Electrical and Computer Engineering

Dr. Yilu Liu (Chair)
Dr. Virgilio Centeno (Co-Chair)
Dr. Richard W. Conners
Dr. Michael S. Hsiao
Dr. Yao Liang
Dr. Tao Lin

February 20th 2006
Blacksburg, Virginia

Keywords: Power System Frequency Tracking, Frequency Disturbance Recorder, FDR, Frequency Monitoring Network, FNET, Real-time Embedded System Design, GPS, Wide-area Measurement System

High Accuracy Real-time GPS Synchronized Frequency Measurement Device for Wide-area Power Grid Monitoring

Chunchun Xu

Abstract

Frequency dynamics is one of the most important signals of a power system, and it is an indicator of imbalance between generation and load in the system. The Internet-based real-time GPS-synchronized wide-area Frequency Monitoring Network (FNET) was proposed to provide imperative frequency dynamics information for a variety of system-wide monitoring, analysis and control applications. The implementation of FNET has for the first time made the synchronized observation of the entire U.S. power network possible with very little cost.

The FNET is comprised of many Frequency Disturbance Recorders (FDR) geographically dispersed throughout the U.S. and an Information Management System (IMS), currently located at Virginia Tech. The FDR works as a sensor, which performs local measurements and transmits calculations of frequency, voltage magnitude and voltage angle to the remote servers via the Internet. Compared with its commercial counterpart Phasor Measurement Unit (PMU), FDR provides less expensive version for networked high-resolution real-time synchronized. The improved single phase algorithm in the FDRs made it possible to measure at 110V level which is much more challenging than PMUs due to the noise involved at this level.

This research work presents the challenges and issues of both software and hardware design for the novel measurement device FDR, which is one of the devices with the highest dynamic precision for power system frequency measurement. The DFT-based Phasor Angle Analysis algorithm has been improved to make sure the high-resolution measuring FDRs are installed at residential voltage outlets, instead of substation high-voltage inputs. An embedded 12-channel timing GPS receiver has been integrated to

provide an accurate timing synchronization signal, UTC time stamp, and unit location. This research work also addresses the harmonics, voltage swing and other noise components' impacts on the measurement results, and the optimized design of filters and a coherent sampling scheme to reduce or eliminate those impacts. The verification test results show that the frequency measurement accuracy of the FDR is within +/-0.0005Hz, and the time synchronization error is within +/-500ns with suitable GPS antenna installation. The preliminary research results show the measurement accuracy and real-time performance of the FDR are satisfactory for a variety of FNET applications, such as disturbance identification and event location triangulation.

Acknowledgement

I would like to express my sincere appreciations to my advisor, Dr. Yilu Liu, for her consistent guidance, encouragement and support throughout my study and life here at Virginia Tech. She is not only a successful researcher with profound knowledge and acute foresight, but also a great person with a warm heart and amazing personalities. I feel so honored to be one of her students, and I enjoy my experience working with her very much.

My sincere appreciation goes to Dr. Richard W. Conners for his invaluable guidance and leadership on the implementation of frequency disturbance recorder, and to my co-advisor Dr. Virgilio Centeno for his selfless sharing and guidance throughout my research activities. I also would like to express sincere appreciation to Dr. Michael S. Hsiao, Dr. Tao Lin, and Dr. Yao Liang for their support and serving as my committee members. The research work would not have been possible without their guidance and assistance.

Special thanks go to Dr. Fred Wang, Dr. Mark Shepard, Rajarshi Sen, and William Collard for their strong recommendation and invaluable guidance during my Co-op with General Electric, which enriched my study and professionally provided me the valuable industrial experience that I cannot get at a university. Many thanks also go to Dr. A.G. Phadke for his invaluable guidance on frequency estimation algorithm development.

It has been a great pleasure to work with the talented, creative, helpful and dedicated colleagues. I would like to express my gratitude and appreciation to all fellow students in my team: Dr. Kevin Zhong, Ryan Zuo, Dr. Henry Zhang, Dr. Steven Tsai, Lei Wang, Dr. Ling Chen, Bruce Billian, Vivian Liang, Matt Gardner, Dr. Li Zhang, Dawei Fan, Will Kook, Mark Baldwin, Jon Burgett, Kevin Khan, Keith McKenzie, Jason Bank, Jingyuan Dong, Il-Yop Chung. Also thanks to Glenda Caldwell for her administrative support.

Many thanks also go to my dear friends for all their understandings and being supportive all the time. They are Yunjun Xu, Yanli Xia, Lingyin Zhao, Ming Xu, Ping Yan, Yu Zheng, Wenduo Liu, Fenghua Chen, Asma Waqar, Kevin Yang, Xiaodan Crouch, Jian Liu, Guiqing Wang, Fei Liu, Xiangfei Ma, Tiger Zhou, Ming Leng, Jinghong Guo, Jia Wei, Jian Yin, Yang Qiu, Juanjuan Sun, Bing Lu, Gary Yao, Rao Shen, Yan Jiang, Jing Xu, Xi Xiong, Ning Zhu, Wei Dong, Dianbo Fu, Julu Sun, Yuancheng Ren, Qian Liu, Chingshan Lev.

Finally, with the deepest love, I would like to express my heartfelt appreciation to my mother Zhongfang Shi, my boy friend Chuanyun Wang, my cousins Feng Xu, Xiaoyong Gu, Xuehua Zhu, and other families for their unconditional love and providing me guidance and discipline needed for achievements. Last year, my dearest father passed away, which leaves me indescribable grief and prevents me picturing this special moment and the promising future with him. He was a great father, and there is nothing in the world could bear comparison with his kindness and love. Without him, it was impossible for me to achieve this accomplishment. Wherever he is, I know I will miss and love him forever.

Dedicated to my parents

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Scope and Background of Research Interest

The electric power network is one of the largest and most important networks in the world, since it affects all aspects of our lives every moment. However, recent large-scale blackouts throughout the world warn us that most of the power grids are heavily loaded and operate at their maximum capacity, and the power grids are very vulnerable to system-wide disturbance and cascade failures. US deregulation, competition and the increase of complexity of today's power networks have only further exacerbated power system stability issues.

In order to ensure system stability in a heavily loaded system, all or most installed components should remain in service, and the right actions must be taken quickly if the system does not recover after a serious event. To cater to those requirements, one of the best solutions is to have wide-area real-time monitoring systems, which could provide critical information for a variety of operation, protection, control and analysis applications.

In recent years, more and more wide-area monitoring systems have been built around the world to meet the growing needs for all kinds of system-stability-related applications. Most of the current wide-area monitoring systems are based on Phasor Measurement Units (PMU). PMU is a measurement device developed and commercialized in the late 1980s. It is capable of recording long-term phasor representations of voltage and current, which could define the status of a power network in phasor format and help improve the power grid's performance. However, in order to obtain the valuable measurement information for observing a power system's dynamics, a number of PMUs are required to be installed in facilities with high-voltage three-phase inputs, which make PMUs costly to manufacture, install and maintain.

In terms of global energy balance, as long as power generation is equal to power consumption and loss of loads, the frequency of the entire power network remains constant. However, in the real world, the balance between the generation and load constantly suffers perturbation due to fluctuations or outage of loads, generations and interconnections. Frequency is a good indicator of imbalance between power generations and loads, and it is one of the most important signals for disturbance identification. GPS timing synchronization techniques, coupled with fast-growing information technology, provides a great opportunity to measure and monitor the frequency wide area in real time. An Internet-based Frequency Monitoring Network (FNET) was proposed in 2001 [1], and has since been implemented .

The FNET consists of two major parts, the Frequency Disturbance Recorder (FDR) and the Information Management System (IMS). The FDR works as a sensor, which performs local frequency data measurements and transmits the data to remote servers through the Internet. IMS works as a central server, which provides data collection, storage, web service, post-disturbance analysis and other information management functions. FNET provides continuous, real-time, wide-area collection of GPS time-stamped frequency data for monitoring purposes; it made the synchronized observation of the entire US power network possible with reasonable cost for the first time. Based on the valuable information provided by FNET, a variety of protection and control applications could be studied in order to access and improve the performance of power systems during steady-state and transient conditions.

1.2 Challenges

Many studies and investigations show that the lack of effective sensing and monitoring information for system reliability, operation, protection, control process and other applications is one of the major causes of 2003 North American and Italian blackouts [2, 3]. The frequency of a power system remains constant regardless of voltage level; however, when a disturbance occurs in the power grids, the frequency varies. Therefore

the proposed FNET should be able to provide a coherent picture of how the entire power grid behaves during both the steady state and disturbances, and help prevent cascade blackouts like the one in the summer of 2003. In order to provide critical support for system monitoring, measurement, protection, operation, control, and reliability assessment, the measurement of FNET should be networked, occur in real-time, be synchronized, and be done with satisfactory accuracy. Those challenges have been the keys to implementation of FNET.

1.3 Frequency Monitoring Network (FNET)

Frequency is one of the most important parameters for accessing and controlling the power system dynamic performance. In other words, frequency information can be used in many applications of power system operation, monitoring, analysis and control [1]. Traditionally, only local frequency measurement with the assumption of a steady state can achieve acceptable estimation accuracy. However, system frequency varies with respect to time and location, and the frequency information during disturbance or faults is much more valuable. Accurate, continuous, time-synchronized, quasi-real-time wide-area dynamic frequency information is the key for accurate monitoring, control and protection applications of a power system, including: disturbance and events analysis, system monitoring and tests, power system models and theories validation, and method development for grid reliability management

In the past several decades, there have been many attempts to develop frequency estimation algorithms and measurement device, in which the Phasor Measurement Unit (PMU) has been a big success. However, due to some issues discussed in the previous section, PMUs have not been widely used for many research and academic purposes.
The Frequency Monitoring Network (FNET) provides a cost-effective way for accurate dynamic frequency monitoring. The FNET implementation includes, but is not limited to the following:

- Development and implementation of highly accurate frequency estimation algorithms and techniques;
- Design and development of measurement device hardware and software;
- Development of an Internet-based information management system;
- Development of real-time and offline post data processing algorithms for a variety of power system analysis and control applications;
- Development of data compression or aggregation methods for data transmission, storage and processing.

1.4 Frequency Estimation Algorithm

In real world power systems, there are a growing number of power electronics devices, arc furnaces and other equipment, which bring harmonics, flickers and noise [4] to the modern power network. It is always a challenge to discover and develop reliable algorithms for precise frequency estimation in presence of harmonics and noise. During the past several decades, a variety of synchronous frequency estimation algorithms have been proposed. These include, but are not necessarily limited to: Zero-crossing Techniques, Least Error Square Approximation, Fast Fourier Transform, leakage effect of DFT, Phasor Angle Analysis, and Kaman Filtering Techniques.

In the zero-crossing technique, a very simple circuit is required, which could minimize the design efforts and computation burden. However, the technique is vulnerable in the presence of noise, and the measurement accuracy heavily depends on the accuracy of the oscillator. It's not suitable for applications requiring heavy measurement data, since only one new frequency measurement result at most could be obtained every cycle.

The Leakage Effect of the Fourier Transform can be easy implemented, but accuracy of this algorithm suffers a great deal from the non-fundamental components of a real power system. Since it only works for sinusoidal waveform, a zero-crossing detector is required for the algorithm implementation, which could bring additional noise issues. The Least

Error Square Technique can adjust the coefficients of pseudo-inverse matrix to improve the noise suppression. However, this technique will introduce a larger error for large frequency deviation. Phasor Angle Analysis provides fast recursive estimation of phasor and frequency by calculating the DFT fundamental frequency component of an input signal. The phasor angle analysis is capable of self-filtering in presence of noise and can provide a highly precise estimation of frequency. It is therefore the best fit for FNET implementation.

1.5 Frequency Disturbance Recorder

The Frequency Disturbance Recorder (FDR) is the key component of FNET. It takes the single phasor voltage input from a 110V residential wall outlet, estimates the synchronous frequency, voltage, and other power system parameters in real time, and transmits the measurement results to a central server via Internet.

The FDR consists of 4 major parts; the computation unit, GPS, the device server, and signal digitizing and conditioning. The computation unit is the place where the frequency estimation algorithm, process scheduler, and data processing are running. In the first generation, the computation unit is a microcontroller, and in the second generation it could be an MCU, DSP or PC depending on different installation scenarios.

The GPS receiver provides the pulse per second (PPS) signal and UTC time stamp. The PPS is the trigger to synchronized pulse train of voltage sampling, and the UTC time stamp provides the common timing tag for the sampling at a certain instant.

A device server is capable of converting serial communication to the Ethernet, which make all the measurement results networked and accessible in real-time. The signal digitizing and conditioning system processes the input voltage signal in order to provide a clean and reasonable measurement signal to algorithm.

Compared with its commercial counterpart, even though the FDR gives up line voltage angles and flow, it is portable, compact in size, cost effective, and accurate, all of which make it a practical and effective way to deploy the wide-area, real-time synchronized frequency measurement and monitoring system. So far, there are around 30 FDRs in place at universities, research institutes, and offices all across the country providing continuous, GPS-synchronized, real-time collection of power system frequency measurement data.

1.6 Dissertation Outline

This dissertation is organized as follows: The first chapter will give some introduction and background of the research work of this dissertation. The second chapter will address investigation and research work on the frequency estimation algorithm and makes a safe conclusion that the DFT-based Phasor Angle Analysis is the best solution so far for FDR and FNET implementation. In the third chapter, the conceptual design of a portable FDR will be proposed, and the software and hardware architecture of the FDR will be discussed fully. The fourth chapter presents some research work on the issues affecting frequency measurement accuracy and reliability. Some preliminary research work of FNET applications will be presented in the fifth chapter, and the last chapter will summarize the dissertation.

Chapter 2 Frequency Estimation Algorithms Review

2.1 Frequency Estimation Algorithm

The phasor representation can be used to consider the ideal voltage and current signal in power system as a rotation vector:

$$v(t) = X(t)e^{j\theta(t)}. \tag{2.1.1}$$

$X(t)$ and $\theta(t)$ are the instantaneous magnitude and phasor angle respectively, and both of them can be considered as functions of time. Whenever the change of $\theta(t)$ is constant or small enough with respect to the synchronous frequency, i.e. ω_0, the frequency is defined as the inverse of the shortest time interval between two instants with the same value for all times [5].

$$(\forall t)\left(v(t) = v(t-T) = v\left(t - f^{-1}\right)\right) \tag{2.1.2}$$

The instantaneous frequency deviation is defined as

$$\Delta f = \frac{\Delta\omega}{2\pi} = \frac{1}{2\pi}\frac{d\theta}{dt}. \tag{2.1.3}$$

Frequency is one of the most important parameters and is the quality index of the power system. Deregulation, expansion of the transmission and distribution network, and the increased application of power electronics make the power system more complex and make the determination of accurate frequency more difficult.

In the past several decades, a variety of synchronous frequency estimation algorithms have been proposed. These include, but are not limited to:

- Zero Crossing Techniques [6-8],
- Leakage Effect of Fourier Transform [9],
- Phasor Angle Analysis [10-14],
- Least Error Square Techniques [15-17],

- Kalman Filter Techniques [18],
- Smart Discrete Fourier Transforms [19-22],
- Wavelet Approach [23], and
- Adaptive Neural Network Approach [24, 25].

In the following sections, some analysis and comparison results of these algorithms will be discussed, based on which we make a safe conclusion that Phasor Angle Analysis is the ideal solution for FNET and FDR implementation.

2.2 Zero Crossing Techniques

The zero crossing technique is the most intuitive method for frequency measurement. It uses a counter to record the time between two successive zero crossing points (where the signal goes from positive to zero) of a voltage waveform. The relationship between the counts and the period of the voltage is as follows:

$$\frac{1}{2}T = \frac{C_m}{C_{f_0}}\frac{1}{f_0} \tag{2.2.1}$$

Where T is the period of the voltage signal,

C_m is the counter number recorded for the voltage signal been measured,

C_{f_0} is the counter number recorded for the signal operating at nominal frequency, and

f_0 is the nominal frequency.

Besides the technique of measuring the frequency at only the zero-crossing points of a signal, many other variations have been developed. Using the slope of linearly interpolated samples, the Level Crossing Technique measures frequency at every incoming sample. Some other variations extend the measurement window more than two cycles to minimize the noise effects.

When the zero crossing technique is used, a very simple circuit is needed, and the minimum computation burden will be applied to microcontrollers. However, this technique is vulnerable in the presence of noise, and at most one new frequency

measurement result can be obtained every cycle. Additionally, for high accuracy, the very expensive oscillator and peripheral circuit are necessary for precise timing.

2.3 Leakage Effect of Fourier Transform

The leakage effect is the discrepancy between the Fourier Transform and the function's Fourier series of a signal with only part of a cycle. The leakage coefficient η could be written:

$$\eta = \frac{\sum_{k=0}^{N-1} |X(k)| - |X(1)|}{|X(1)|} \tag{2.3.1}$$

where $X(k)$ are the Discrete Fourier Transform terms and

$X(1)$ corresponds to the fundamental component.

Assuming a sinusoidal waveform with a small frequency deviation $(\pm 5Hz)$, the relationship between the leakage coefficient η and the frequency deviation Δf is linear [9]. The governing equation of the relationship is:

$$|\Delta f| = \frac{\eta}{0.095584345}. \tag{2.3.2}$$

For a cosinusoidal waveform, this relation will not hold. In this case, a zero-crossing detector must be applied to detect the zero-crossing point (positive going to zero in this case) to obtain a sinusoidal waveform. However, there's no perfect zero-crossing detector in the real world, so a time error will always be introduced. This implies that the sinusoidal waveform poses a phase delay, which will definitely introduce an error in the frequency estimation.

Another thing that must be noted here is that the leakage coefficient defined above is for only the fundamental component. However, a real power voltage waveform contains a lot of waveforms or noise in addition to the fundamental component. Those non-fundamental components will definitely impair the estimation accuracy using this technique.

2.4 Phasor angle analysis

This DFT-based frequency measurement technique is the most widely used algorithm in the commercial Phasor Measurement Units (PMU).

Consider a pure sinusoidal input signal

$$X(t) = \sqrt{2}X\sin(2\pi ft + \theta). \tag{2.4.1}$$

The phasor of the signal $\overline{X}$ can be expressed as

$$\overline{X} = Xe^{j\theta} = X\cos\theta + X\sin\theta. \tag{2.4.2}$$

Assuming that $X(t)$ is sampled N times per cycle at 60 Hz to produce a sample data sequence $\{x_k\}$,

$$x_k = \sqrt{2}X\sin\left(\frac{2\pi k}{N} + \theta\right) = \sqrt{2}X\left(\frac{e^{j\left(\frac{2\pi k}{N}+\theta\right)} - e^{-j\left(\frac{2\pi k}{N}+\theta\right)}}{2j}\right). \tag{2.4.3}$$

Take the N-point Discrete Fourier Transform of $\{x_k\}$, and evaluating the fundamental frequency component,

$$\overline{X_1} = \frac{2}{N}\sum_{k=0}^{N-1} x_k e^{-j\frac{2\pi}{N}k} = \frac{2}{N}\sum_{k=0}^{N-1} x_k\cos\left(\frac{2\pi}{N}k\right) - j\frac{2}{N}\sum_{k=0}^{N-1} x_k\sin\left(\frac{2\pi}{N}k\right). \tag{2.4.4}$$

Substituting for x_k from (2.4.3),

$$\overline{X_1} = X_c - jX_s \tag{2.4.5}$$

$$X_c = \frac{2}{N}\sum_{k=0}^{N-1}\sqrt{2}X\sin(\frac{2\pi k}{N}+\theta)\cos\left(\frac{2\pi}{N}k\right) = \frac{2}{N}\sum_{k=0}^{N-1}\sqrt{2}X\left[\frac{e^{j\left(\frac{2\pi k}{N}+\theta\right)} - e^{-j\left(\frac{2\pi k}{N}+\theta\right)}}{2j}\right]\left[\frac{e^{j\left(\frac{2\pi}{N}k\right)} - e^{-j\left(\frac{2\pi k}{N}\right)}}{2j}\right]. \tag{2.4.6}$$

Using a geometric series,

$$\sum_{n=p}^{N-1} r^n = \frac{r^p - r^N}{1-r} \tag{2.4.7}$$

$$X_c = \sqrt{2}X\sin\theta \tag{2.4.8}$$

while

$$X_s = \sqrt{2} X \cos\theta . \tag{2.4.9}$$

So in general, the phasor of signal (2.4.1) is:

$$\overline{X} = \frac{1}{\sqrt{2}} j\overline{X_1} . \tag{2.4.10}$$

Consider the phasor of X(t) at the rth sampling interval [10],

$$\overline{X}^{(r)} = \frac{1}{\sqrt{2}} j\overline{X_1}^{(r)} = \frac{1}{\sqrt{2}} j \frac{2}{N} \sum_{k=0}^{N-1} x_{r+k} e^{-j\frac{2\pi}{N}k} . \tag{2.4.11}$$

Substituting for x_{r+k} from (2.4.3) results in [26]

$$\overline{X}^{(r)} = \frac{X}{N} \sum_{k=0}^{N-1} \left(e^{jA} - e^{-jB} \right) \tag{2.4.12}$$

with

$$A = \frac{2\pi\left(r+k\right)}{N} + \theta - \frac{2\pi k}{N} = \frac{2\pi r}{N} + \theta \tag{2.4.13}$$

$$B = \frac{2\pi\left(r+k\right)}{N} + \theta + \frac{2\pi k}{N} = \frac{2\pi}{N}\left(r+2k\right) + \theta . \tag{2.4.14}$$

Then the phasor is

$$\overline{X}^{(r)} = \frac{X}{N} \sum_{k=0}^{N-1} \left(e^{j\frac{2\pi}{N}r} e^{j\theta} - e^{-j\frac{2\pi}{N}r} e^{-j\frac{2\pi}{N}2k} e^{-j\theta} \right). \tag{2.4.15}$$

Applying the identity

$$\sum_{n=0}^{N-1} \left(e^{j\theta} \right)^n = \frac{\sin\left(\frac{N\theta}{2}\right)}{\sin\left(\frac{\theta}{2}\right)} e^{j(N-1)\frac{\theta}{2}} \tag{2.4.16}$$

leads to

$$\sum_{k=0}^{N-1} e^{-j\frac{2\pi}{N}2k} = 0 \tag{2.4.17}$$

so

$$\overline{X}^{(r)} = \frac{X}{N} \sum_{k=0}^{N-1} \left(e^{j\frac{2\pi}{N}r} e^{j\theta} \right) = X e^{j\theta} e^{j\frac{2\pi}{N}r} . \tag{2.4.18}$$

Now assuming a signal with the off nominal frequency $f + \Delta f$,

$$X(t)=\sqrt{2}X\sin\left(2\pi\left(f+\Delta f\right)t+\theta\right). \tag{2.4.19}$$

The new phasor at the rth sampling is

$$\overline{X}_{f+\Delta f}^{(r)}=\frac{X}{N}\sum_{k=0}^{N-1}\left(e^{jC}-e^{-jD}\right) \tag{2.4.20}$$

with

$$C=\frac{2\pi}{N}r+\frac{2\pi\Delta f}{N60}r+\theta+\frac{2\pi\Delta f}{N60}k \tag{2.4.21}$$

$$D=\frac{2\pi}{N}r+\frac{2\pi\Delta f}{N60}r+\theta+\left(\frac{4\pi}{N}+\frac{2\pi\Delta f}{N60}\right)k. \tag{2.4.22}$$

Then

$$\overline{X}_{f+\Delta f}^{(r)}=\frac{X}{N}\left(e^{j\frac{2\pi}{N}r}e^{j\frac{2\pi\Delta f}{N60}r}e^{j\theta}\sum_{k=0}^{N-1}e^{j\frac{2\pi\Delta f}{N60}k}-e^{-j\frac{2\pi}{N}r}e^{-j\frac{2\pi\Delta f}{N60}r}e^{-j\theta}\sum_{k=0}^{N-1}e^{-j\left(\frac{4\pi}{N}+\frac{2\pi\Delta f}{N60}\right)k}\right) \tag{2.4.23}$$

for rth data window, the phasor of (2.4.1) is expressed by

$$\overline{X}_{f}^{(r)}=Xe^{j\theta}e^{j\frac{2\pi}{N}r}. \tag{2.4.24}$$

To make the phasor be stationary in the complex plane, a time delay was used to compensate the angular:

$$\overline{Y}^{(r)}=\overline{X}^{(r)}e^{-j\frac{2\pi}{N}r}=Xe^{j\theta}. \tag{2.4.25}$$

Applying the identity (2.4.16) and introducing the phasor expression in (2.4.25) leads to

$$\overline{X}_{f+\Delta f}^{(r)}=\overline{X}_{f}^{(r)}e^{j\frac{2\pi\Delta f}{N60}r}e^{j(N-1)\frac{\pi\Delta f}{N60}}\frac{\sin\frac{\pi\Delta f}{60}}{\sin\frac{2\pi\Delta f}{N60}}-\overline{X}^{*(r)}e^{-j\frac{2\pi\Delta f}{N60}r}e^{j(N-1)\left(\frac{2\pi}{N}+\frac{\pi\Delta f}{N60}\right)}\frac{\sin\frac{\pi\Delta f}{N60}}{\sin\frac{2\pi}{N}+\frac{\pi\Delta f}{N60}}. \tag{2.4.26}$$

For a small value of Δf, the following simplifications hold:

$$\frac{\sin\frac{\pi\Delta f}{60}}{\sin\frac{2\pi\Delta f}{N60}}=1 \tag{2.4.27}$$

$$\frac{\sin\frac{\pi\Delta f}{N60}}{\sin\frac{2\pi}{N}+\frac{\pi\Delta f}{N60}}=0. \tag{2.4.28}$$

So (2.4.26) can be reduced to

$$\overline{X}_{f+\Delta f}^{(r)} = \overline{X}_{f}^{(r)} e^{j\frac{2\pi\Delta f}{N60}r} e^{j(N-1)\frac{\pi\Delta f}{N60}}. \tag{2.4.29}$$

The computed phasor is broken down into its magnitude and angle parts.

By denoting the angle of the new phasor as ψ_r and the angle of the previous phasor as ψ_{r-1}, one can approximate the new phasor as:

$$\psi_r = \frac{\Delta f}{f_0}\frac{2\pi}{N} + \psi_{r-1} \tag{2.4.30}$$

$$\frac{d\psi}{dt} = \lim_{t\to 0}\frac{\psi_r - \psi_{r-1}}{t} \approx \frac{\psi_r - \psi_{r-1}}{1/Nf_0} = 2\pi\Delta f\ . \tag{2.4.31}$$

The system frequency can now be computed with:

$$f = f_0 + \Delta f = f_0 + \frac{1}{2\pi}\frac{d\psi}{dt} \quad , \tag{2.4.32}$$

and the rate of change of frequency is

$$\frac{df}{dt} = \frac{1}{2\pi}\frac{d^2\psi}{dt^2}\ . \tag{2.4.33}$$

The Phasor Analysis Technique is simple in terms of computation burden. Under the assumption that the system frequency is very close to 60Hz in normal operation, the leakage error of DFT is very small. So phasor angle analysis can provide highly accurate frequency estimation with minimum computation complexity and time.

2.5 Least Error Square Technique

The Taylor Series Expansion is the underlying principle of the Least Error Square Technique. A single frequency sinusoidal waveform can be represented as

$$\begin{aligned} v(t) &= A_v \sin(2\pi ft + \theta) \\ &= A_v \cos\theta \sin(2\pi ft) + A_v \sin\theta \cos(2\pi ft) \end{aligned}. \tag{2.5.1}$$

Expanding the Taylor series of $\sin(2\pi ft)$ and $\cos(2\pi ft)$ centered on the nominal frequency f_0, the first three terms are taken,

$$\begin{aligned} v(t_1) &\approx a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + a_{15}x_5 + a_{16}x_6 \\ v(t_2) &\approx a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 + a_{25}x_5 + a_{26}x_6 \\ &\vdots \\ v(t_m) &\approx a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + a_{m4}x_4 + a_{m5}x_5 + a_{m6}x_6 \end{aligned} \tag{2.5.2}$$

where

$$x_1 = A_v \cos\theta, x_2 = (f - f_0) A_v \cos\theta$$
$$x_3 = A_v \sin\theta, x_4 = (f - f_0) A_v \sin\theta$$
$$x_5 = \left(-\frac{(2\pi)^2}{2} f^2 + (2\pi)^2 \, ff_0 - \frac{(2\pi)^2}{2} f_0^2 \right) A_v \cos\theta$$
$$x_6 = \left(-\frac{(2\pi)^2}{2} f^2 + (2\pi)^2 \, ff_0 - \frac{(2\pi)^2}{2} f_0^2 \right) A_v \sin\theta$$

and

$$a_{i1} = \sin(2\pi f_0 t_i), a_{i2} = 2\pi t_i \cos(2\pi f_0 t_i), a_{i3} = \cos(2\pi f_0 t_i)$$
$$a_{i4} = 2\pi t_i \sin(2\pi f_0 t_i), a_{i5} = t_i^2 \sin(2\pi f_0 t_i), a_{i6} = t_i^2 \cos(2\pi f_0 t_i)$$

.

$v(t_1) \ldots v(t_m)$ are measured inputs, and all as can be calculated if the nominal frequency f_0 and sampling instant t are known. The only unknowns are values of xs .

We can rewrite (2.5.2) as

$$\mathbf{AX} = \mathbf{V} \tag{2.5.3}$$

where $\mathbf{A}$ is a $m \times 6$ matrix, and m ought to be greater or equal to 6 to obtain a solution of X . When $m \geq 6$, the least error square estimation of X is:

$$\mathbf{X} = \left[\left[\mathbf{A}^{\mathrm{T}} \mathbf{A} \right]^{-1} \mathbf{A}^{\mathrm{T}} \right] \mathbf{V} . \tag{2.5.4}$$

With the solution of X, the frequency deviation Δf can be found according to equations (2.5.5) and(2.5.6):

$$\frac{x_2}{x_1} = \frac{(f - f_0) A_v \cos\theta}{A_v \cos\theta} = f - f_0 \tag{2.5.5}$$

$$\frac{x_4}{x_3} = \frac{(f - f_0) A_v \sin\theta}{A_v \sin\theta} = f - f_0 . \tag{2.5.6}$$

Another alternative is the combination of (2.5.5) and (2.5.6):

$$(f - f_0)^2 = \frac{x_2^2 + x_4^2}{x_1^2 + x_3^2} . \tag{2.5.7}$$

The pseudo-inverse matrix $\left[\left[\mathbf{A}^{\mathrm{T}}\mathbf{A}\right]^{-1}\mathbf{A}^{\mathrm{T}}\right]$ can be considered a filter, which will amplify or suppress noise depending on its coefficients. Both the sampling rate and the window size play rather important roles in determining the pseudo-inverse matrix. Experimental data shows that increasing the sampling rate improves the noise immunity characteristics of the pseudo-inverse matrix [15]. However, only extending the data window will not reduce the noise effects on the output. The underlying principle of the least error square is the approximation of the voltage waveform using the Taylor Series instead of the least error square. So, for a more accurate result, one needs to increase both the window size and the terms of Taylor Series. However, more Taylor Series terms imply more unknowns, and then a bigger pseudo-inverse matrix, all of which will add up to more of a computational burden. Also, if the waveform frequency is not constant within the longer window, then additional errors will be introduced. Giray and Sachdev found that for large frequency deviation, the error introduced will also be larger. To obtain better accuracy and to meet the speed requirements of relay systems, they proposed a lookup table to compensate the errors [16].

2.6 Kalman Filter Techniques

Girgis and Huang were first to propose the Kalman Filtering Technique for frequency estimation in detail [18]. The Kalman filter addresses the general problem of trying to estimate the state of a discrete-time-controlled process that is governed by the linear stochastic difference equation

$$\mathbf{x}_{k+1} = \phi_k \mathbf{x}_k + \mathbf{w}_k \tag{2.5.8}$$

where $\mathbf{x}_k$ is the state vector at step k,

$\boldsymbol{\phi}_k$ is the state transition matrix, and

$\mathbf{w}_k$ is the process noise vector with a covariance matrix $\mathrm{E}\left[\mathbf{w}_k \mathbf{w}_j^{T}\right] = \mathbf{Q}_k$, in which $\mathbf{Q}_k = 0$ if $k \neq j$.

The measurement equation is

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{2.5.9}$$

where $\mathbf{z}_k$ is the measurement vector at step k,

$\mathbf{H}_k$ is the measurement matrix,

$\mathbf{v}_k$ is the random measurement error vector with covariance

$\mathrm{E}\left[\mathbf{v}_k \mathbf{v}_j^{\,T}\right] = \mathbf{R}_k$ in which $\mathbf{R}_k = 0$ if $k \neq j$.

The random variables $\mathbf{w}_k$ and $\mathbf{v}_k$ represent the process and measurement noise respectively. And they are assumed to be independent, white, and with normal probability distributions.

Defining the estimate error covariance P_k and the Kalman gain matrix K_k, the Kalman filter time update equations can be represented as:

$$\begin{aligned} \hat{\mathbf{x}}_{k+1/k} &= \boldsymbol{\phi}_k \hat{\mathbf{x}}_k \\ \mathbf{P}_{k+1/k} &= \boldsymbol{\phi}_k \mathbf{P}_k \boldsymbol{\phi}_k^{\,T} + \mathbf{Q}_k \end{aligned}, \tag{2.5.10}$$

and the Discrete Kalman filter measurement updates equations are:

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_{k/k-1}\mathbf{H}_k^{\,T}\left(\mathbf{H}_k \mathbf{P}_{k/k-1}\mathbf{H}_k^{\,T} + \mathbf{R}_k\right)^{-1} \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k/k-1} + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k/k-1}\right) \\ \mathbf{P}_k &= \mathrm{E}\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_k\right)\left(\mathbf{x}_k - \hat{\mathbf{x}}_k\right)^T\right] \end{aligned} \quad . \tag{2.5.11}$$

A two-state model and a three-state model can be used in the Kalman Filter Technique. The two-state Kalman filter technique calculates the voltage phasor first, and then computes the frequency using $\frac{d\phi}{dt} \approx 2\pi\Delta f$. In this technique, the equation of a waveform can be represented as:

$$v(t) = A_v \exp\left(j\omega t + \phi\right) = x_1 \cos\omega_0 t - x_2 \sin\omega_0 t \tag{2.5.12}$$

where x_1 and x_2 are real and imaginary parts of phasor respectively.

The state vector is:

$$\mathbf{x}_k = \begin{bmatrix} x_{1_k} \\ x_{2_k} \end{bmatrix} \tag{2.5.13}$$

and the transition matrix can be defined as:

$$\phi_k = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}. \tag{2.5.14}$$

The measurement equation can be defined as:

$$\mathbf{z}_k = \begin{bmatrix} \cos(\omega_0 k\Delta t) & -\sin(\omega_0 k\Delta t) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \mathbf{v}_k . \tag{2.5.15}$$

After knowing the initial estimation of $\mathbf{P}_0$ and $\hat{\mathbf{x}}_0$, (2.5.10) and (2.5.11) can be used to solve the real-time phasor angles recursively. One thing needs to be noted: by using this two-state Kalman Filter technique, a small error in phasor angles computation will introduce a large error in the later frequency estimation.

A three-state Kalman filter model can compute frequency deviation directly. The equation of the waveform can be represented as:

$$v(t) = A_v e^{j\left[(\omega+\Delta\omega)t+\phi\right]} = x_1 \cos(\omega_0 t + 2\pi x_3 t) - x_2 \sin(\omega_0 t + 2\pi x_3 t) \tag{2.5.16}$$

where x_1 and x_2 still represent the real and imaginary components of the phasor respectively, and x_3 represents the frequency deviation.

The state vector is

$$\mathbf{x}_k = \begin{bmatrix} x_{1_k} \\ x_{2_k} \\ x_{3k} \end{bmatrix}. \tag{2.5.17}$$

The transition matrix can be a 3×3 Identity Matrix, and the measurement equation can be

$$\mathbf{z}_k = h_k(x) + \mathbf{v}_k \tag{2.5.18}$$

where $h_k(x) = x_{1_k} \cos(\omega_0 k\Delta t + 2\pi x_{3_k} \Delta t) - x_{2_k} \sin(\omega_0 k\Delta t + 2\pi x_{3_k} \Delta t)$.

Similarly, (2.5.10) and (2.5.11) can be used to solve x_{3k}, if the initial estimation of $\mathbf{P}_0$ and $\hat{\mathbf{x}}_0$ is known.

The accuracy of the result obtained by the three-state model is much better than the accuracy of the result obtained by the two-state model. Kalman has the ability to filter out variable covariance noise, so the Kalman filter is a very accurate technique for frequency measurement, especially in the presence of noise. Also, the Kalman filter technique

allows different model representations for frequency deviation, such as random constant, random walk, or a combination of both.

However, the Kalman filter technique is built based on the assumption that the probability density functions of the state vectors are known, which is not always true. High values for the variance of the measurement noise need to be used in order to minimize the noise effects on the measurement result. The high variance will introduce unnecessary delay, which causes a longer response delay in estimation calculation. The three-state Kalman filter requires on-line calculation of the Kalman gain, which implies that fast microprocessor or Digital Signal Processor is needed for algorithm implementation.

2.7 Smart Discrete Fourier Transforms

In 2000, Yang and Liu [19] proposed a series of DFT-based algorithms to calculate frequency in real time. The fundamental component of a waveform is considered to be:

$$x(t) = X_1 \cos(\omega t + \phi_1) . \tag{2.6.1}$$

This also can be presented as:

$$x(t) = \frac{\hat{x}_1 e^{j\omega t} + \hat{x}_1^{*} e^{-j\omega t}}{2} \tag{2.6.2}$$

where $\hat{x}_1^{*}$ is the complex conjugate of $\hat{x}_1$.

Defining $z_1 = real(a) = \cos\left(\frac{2\pi}{60N}(60 + \Delta f)\right)$, and (60*N) as the sampling rate to generate voltage sample set $\{x(k)\}$, the following expression could be obtained after some algebraic manipulations.

$$\begin{array}{llll} 2\times & & x(r+1) & \times z_1 \\ -1\times & & (x(r)+x(r+2)) & =0 \end{array} \tag{2.6.3}$$

z_1 is the only unknown in (2.6.3), so the frequency could be obtained as

$$f = 60 + \Delta f = \cos^{-1}\left(real\left(z_1\right)\right) \times \frac{60N}{2\pi}. \tag{2.6.4}$$

If considering an input signal with integral harmonic, non-integral harmonic and DC offset at the same time,

$$x(t) = X_l \cos(\omega t + \phi_1) + X_m \cos(m\omega t + \phi_m) + X_n \cos(\omega_n t + \phi_n) + X_d e^{-\alpha t} \tag{2.6.5}$$

where $X_l \cos(\omega t + \phi_1)$ is the fundamental component,

$X_m \cos(m\omega t + \phi_m)$ is the integral harmonics,

$X_n \cos(\omega_n t + \phi_n)$ is non-integral harmonic component,

$X_d e^{-\alpha t}$ is DC offset,

$\frac{1}{\alpha} = \tau$ is the time constant of the signal,

and the fundamental frequency component of DFT of $\{x(k)\}$ is

$$\bar{x}_r = \frac{2}{M} \sum_{k=0}^{M-1} x(k+r) e^{-j\frac{2\pi k}{N}}. \tag{2.6.6}$$

After algebraic operations an expression similar to (2.6.3) can be obtained, which provides the solution of z_n and a_d. So f_n and τ can also be calculated by:

$$f_n = 60 + \Delta f_n = \cos^{-1}\left(real\left(z_n\right)\right) \frac{60N}{2\pi} \tag{2.6.7}$$

$$\tau = \frac{1}{60N \log a_d}. \tag{2.6.8}$$

There's no leakage error in this technique, which means it could estimate the frequency from 0 to half of the sampling frequency. So the smart Discrete Fourier Transforms provide a good way to estimation frequency over a wide range in the presence of harmonics and DC offset.

However, this technique needs tradeoffs between estimation accuracy, computation complexity and time, and observation window length, and this technique is not suitable for real-time processing due to the increased computation complexity.

2.8 Wavelet Approach

Wavelets are mathematical functions that provide representation of a signal corresponding to different frequency components, and then study each component with a resolution matched to its scale. Compared with Fourier methods, wavelets have an advantage for analyzing of signals that contain discontinuities and spikes.

The analyzing wavelet function $\psi(t)$ satisfies the "admissibility condition"

$$C_{\psi} = \int_{-\infty}^{+\infty} \frac{\|\psi(\omega)\|}{|\omega|} d\omega < \infty . \tag{2.6.9}$$

This presents the band pass filtering property of $\psi(t)$, and $\psi(\omega)$ is the frequency response of $\psi(t)$. Then a series of wavelet can be derived from $\psi(t)$:

$$\psi_{a,b}(t) = \frac{1}{a}\psi\left(\frac{t-b}{a}\right) \tag{2.6.10}$$

where a is the scaling (dilation) factor, and b is the time shifting (of translation) factor.

Similar to the Fourier Transform, the Continuous Wavelet Transform (CWT) measures the difference between the analyzed real signal $s(t)$ and sample wavelet function $\psi_{a,b}(t)$, and it is defined as the inner product of $\psi_{a,b}(t)$ and $s(t)$:

$$W_{\psi}(a,b) = < s(t), \psi_{a,b}(t) > = \int_{-\infty}^{+\infty} s(t)\overline{\psi}_{a,b}(t)\,dt = \int_{-\infty}^{+\infty} s(t)\overline{\psi}_{1,a}(b-t)\,dt = s(t) * \psi_{1,a}(t) \tag{2.6.11}$$

of which $\psi_1(t)$ is the mother filter function and satisfies $\psi_1(t) = \overline{\psi(t)}$ or frequency response $\psi_1(\omega) = \overline{\psi}(\omega)$; and sample filter function $\psi_{1,a}(t)$ is derived from mother filter function $\psi_1(t)$ and satisfies $\psi_{1,a}(t) = \frac{1}{\sqrt{a}}\psi_1\left(\frac{t}{a}\right)$ or frequency response $\psi_{1,a}(\omega) = \sqrt{a}\psi_1(a\omega) = \sqrt{a}\overline{\psi}(a\omega)$. The frequency response of $W_{\psi}(a,b)$ can be derived as

$$W_{\psi}(a,\omega) = S(\omega)\psi_{1,a}(\omega) = \sqrt{a}S(\omega)\psi_1(a\omega) = \sqrt{a}S(\omega)\overline{\psi}(a\omega) \tag{2.6.12}$$

in which $S(\omega)$ is the frequency response of $s(t)$.

Similar to the orthogonal filters of the Fourier algorithm, the CWT acts as complex band pass filter, and can used to estimate rotation vector from original current or voltage signal. Assuming $W_R(t)$ and $W_I(t)$ are the real part and imaginary part of CWT, the instantaneous frequency can be derived.

$$\begin{aligned} W_R(\tau) &= \mathrm{Re}\left(W_\psi(a,\tau)\right) = s(t) * \left[\mathrm{Re}\left(\psi_{1,a}(\tau)\right)\right] \\ W_I(\tau) &= \mathrm{Im}\left(W_\psi(a,\tau)\right) = s(t) * \left[\mathrm{Im}\left(\psi_{1,a}(\tau)\right)\right] \\ \theta(\tau) &= \tan^{-1}\left(W_I(\tau)/W_R(\tau)\right) \\ f(\tau) &= \frac{1}{2\pi}\frac{d\theta(\tau)}{d\tau} \end{aligned} \tag{2.6.13}$$

By defining a proper polynomial-type mother wavelet, the recursive CWT can also be implemented to save significant computation time and resources.

The wavelet approach can track the frequency over a wide range under dynamic conditions, and it is immune to harmonics and DC components. However, compared with the Fourier Algorithm, the estimation accuracy of CWT is limited even within the small range around the nominal frequency. Due to its complexity, the CWT could not be easily implemented and requires more computation time and resources than the Fourier algorithm.

2.9 Adaptive Neural Network approach

The Adaptive Neural Network approach uses a single-layer adaptive neural network consisting of adaptive neurons called an adaline. The voltage signal of a power system is modeled as a difference equation, where coefficients are identified using an adaline. The weight vector of the adaline is adaptively changed by using a discrete error equation rather than the conventional back propagation technique.

Considering an input signal containing a decaying DC component and harmonics:

$$y_t = \sum_{m=1}^{M} A_m \sin\left(m\omega_0 t + \phi_m\right) + a_{dc}\exp\left(-\alpha_{dc} t\right) + \varepsilon(t) \tag{2.7.1}$$

where M is the highest frequency component in the signal,

$a_{dc}\exp(-\alpha_{dc}t)$: is the decaying DC component,

$\varepsilon(t)$: is additive noise,

the relationship between three consecutive samples can be represented as follows

$$y_t = \cos\omega_0 T_s y_{t-1} - y_{t-2} + \sum_{m=1}^{M}\{A_m(2\cos m\omega_0 T_s - 2\cos\omega_0 T_s)\cdot[\sin m\omega_0(t-T_s)+\phi_m] - 2\cos m\omega_0 T_s a_{dc}\} - 2(1-\cos\omega_0 T_s)kT_s a_{dc}\alpha_{dc} \quad (2.7.2)$$

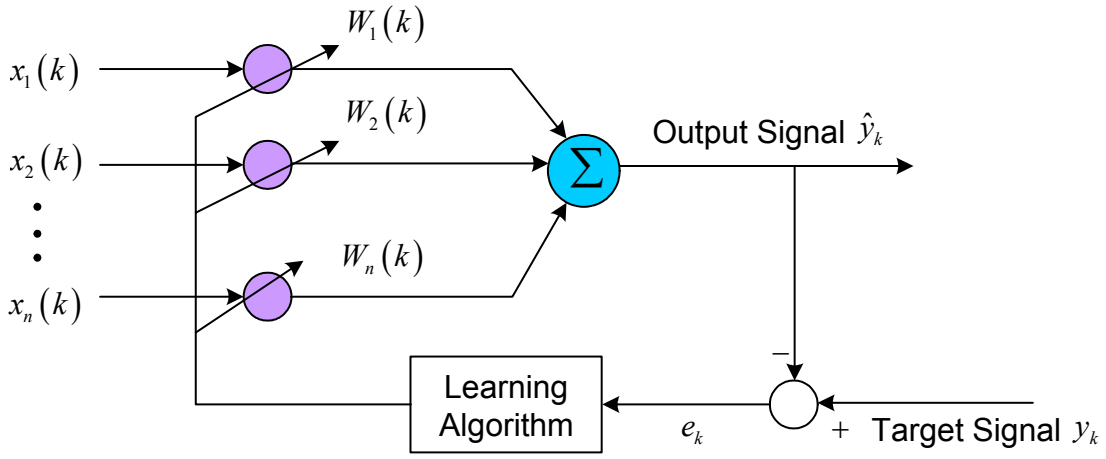


Figure 2.1 Block diagram of ADALINE

Figure 2.1 is the block diagram of an Adaline, and the learning algorithm can be developed as

$$W_{k+1} = W_k + \frac{\alpha e(k)\theta(X)}{X^T\theta(X)} \quad (2.7.3)$$

Where k is the time index,

$W_k = [W_1(k), W_2(k), \cdots, W_n(k)]^T$: is the weight vector at time k,

$X = [x_1, x_2, \cdots, x_n]^T$: is the input vector,

α : is the reduction factor;

$$\theta(X)=\begin{bmatrix} SGN(x_1) \\ SGN(x_2) \\ \vdots \\ SGN(x_n) \end{bmatrix} \text{ with } SGN(x)=\begin{cases} +1 & \text{if } x>0 \\ -1 & \text{if } x<0 \end{cases}.$$

Therefore, if the pure sinusoidal input is considered, the weight vector of the adaline converges to

$$W=\begin{bmatrix} 2\cos\omega_0 T_s \\ -1 \end{bmatrix} \tag{2.7.4}$$

and the frequency estimation can be derived as

$$f_0=\left[\cos^{-1}\left(W(1)/2\right)\right]/\left(2\pi T_s\right). \tag{2.7.5}$$

If an input signal containing the DC component and harmonics is taken into consideration, the frequency is updated as each iteration as

$$\omega_0(k)=\cos^{-1}\left(0.5W_{1,k}\right)/\left(2\pi T_s\right). \tag{2.7.6}$$

The Adaptive Neural Network approach is a straightforward method, and easily implemented. However in a practical environment in the presence of harmonics, noise and a DC component, the estimation accuracy will be severely jeopardized, from 0.001Hz for a pure sinusoidal input to 0.05Hz for a real world input.

2.10 Summary

After exploring different kinds of frequency estimation algorithms, we find that each algorithm has its unique advantages and disadvantages. Precise frequency estimation is not only a choice of a certain algorithm, but also should be considered as an unconstrained optimization problem. Phasor angle analysis provides fast and accurate frequency estimation over a wide range of interested frequency, and the computation requirement is modest for online implementation. From a practical point of view, we consider Phasor Angle Analysis as the best choice for FDR implementation.

Chapter 3 FDR Conceptual Design

3.1 Frequency Monitoring Network Architecture

The Frequency Monitoring Network (FNET) was first proposed in 2001 to meet the fast-growing demands of wide area monitoring [1]. The underlying concept of FNET is that in steady states the power system frequency remains constant regardless of voltage level; however, when a significant disturbance occurs, the frequency varies with respect to time and location. According to this proposal, the FNET should provide continuous real-time wide-area time-synchronized measurement and analysis of power grids' frequency, and feed those values to a variety of power system control and operation applications.

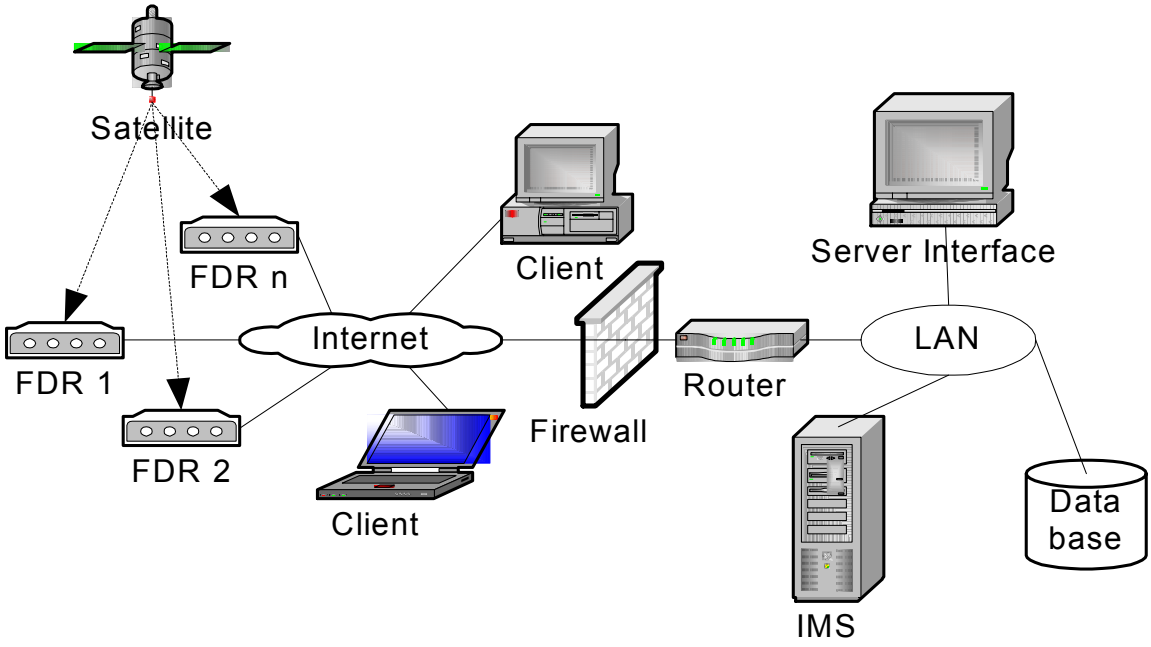


Figure 3.1 Frequency Monitoring Network (FNET) Architecture

Figure 3.1 shows the FNET architecture. The whole FNET system consists of three major components: Frequency Disturbance Recorders (FDR), the Information Management Server (IMS), and the Internet. Frequency Disturbance Recorders or FDRs perform as sensors, which are geographically dispersed at different locations in the US power grid. They take the 110V single-phase voltage input of residential wall outlets, calculate and

analyze the local frequency changing, and communicate with remote central servers via the Internet for data transmission and system configuration.

The Information Management Server (IMS) provides data collection, storage, communication, database operations, and web service. The IMS design is based on the multi-tier client/server architecture, which coordinates, integrates and models the data collection, processing, display and other functions. The IMS performs four major functions:

- Communication with FDRs
- Database operation and management
- GUI and display
- Provides web service

The communication between FDRs and the IMS server is based on the TCP/IP protocol, which ensures a reliable connection and data transmission. The ongoing and future research of IMS functions includes historical data retrieval, automatic event trigger and reconstruction, and event prediction.

One of the most crucial innovations of FNET is that the FDR can perform the frequency measurement based on the 110V voltage input from 110V wall outlets; thus the overall cost of FNET can be significantly reduced, which enables the build up, expansion and reinforcement of FNET within a short period of time. With the efforts of the research team at Virginia Tech and associate members in different universities and industrial facilities, the FNET system started operation in summer of 2004, and there are almost 30 FDR units installed in critical locations as of January 2006. The FNET has made observations of the entire power grid of US possible for the first time.

3.2 Frequency Disturbance Recorder Design Overview

The Frequency Disturbance Recorder (FDR) is a measurement device that is capable of taking the single phase voltage from a wall plug, calculating and analyzing the frequency

dynamics, and sending measurement results to central servers. In FNET, FDRs perform as remote sensors, and are distributed at many critical observation locations all over the US. The measurement accuracy is one of the most important concerns for FDR design, as all FNET applications are based on the measurement data. In other words, the more accurate the measurements are the more reliable and valuable the FNET applications are. According to the review and comparison results of the frequency estimation algorithms discussed in Section 2, the Phasor Angle Analysis technique can provide acceptable frequency estimation accuracy as well as many other advantages for algorithm implementation.

In order to capture the whole picture of power grid performance throughout the US at any given instant, all the measurement data should have a common time reference for synchronization, which is the most important feature for wide-area monitoring. The use of different time references in monitoring systems of different companies is partially responsible for the delay in determining the causes of the 2003 North America Blackout. In the past several years, recognizing the importance of frequency at different locations for system performance monitoring and analysis, many attempts have been made to synchronize the frequency measurement. However, none of these attempts could be called a success, as those technologies put severe limitations on the achievable accuracy of synchronization. It is only in recent years, as GPS has become more available and economical for civil applications, that we can synchronize the measurement in a practical and affordable way with a synchronization error of less than 1 microsecond.

According to an EPRI study, the noise of a real world power system is a combination of harmonic noise, white noise and random noise[27]. Harmonic noise comes from the harmonic components in the input signal. White noise is the frequency spectrum over a wide range of frequencies, and all frequency components in equal amounts. A random spike is generated by external random changes, such as sudden load change, lighting, and capacitor switching. The noise in the power system has severe impacts on the accuracy of frequency measurement, so analog and digital filter design is an important part of FDR implementation. In recent years, significant improvement has been made on MATLAB

Signal Processing Toolbox, which provides a convenient and effective Filter Design and Analysis Tool (FDA Tool). With the FDA Tool, it is easy to make a primary design, analyze and adjust digital filters to meet system requirements. Coupled with advanced digital processing techniques of a Microcontroller DSP or FPGA, digital filters can be effectively implemented with expected performance.

Traditionally, most power system measurements are performed locally, so geographic distance and time delay prevent those measurements from being easily employed for different power system monitoring, protection and control purposes. In recent years, with information technologies that have developed so quickly, the internet has become a safe and reliable media for data transmission, which makes wide-area synchronized measurement more realistic and desirable.

Taking full advantage of the state-of-the-art technologies and concepts in different areas, the compact portable frequency disturbance recorder provides the GPS synchronized, close to real-time frequency measurement and analysis data from flexible locations in US to one or multiple central servers.

Conceptually, the FDR comprises four main functional components:

- Main computation unit: it can be a microcontroller, DSP, FPGA or a PC,
- GPS receiver,
- Digital signal conditioning part,
- Serial to Ethernet converter.

Figure 3.2 is the system block diagram of the second generation MCU-based FDR.

As shown in Figure 3.2, the main computation unit is based on MPC555 from Freescale, and CME-555 is the off-the-shelf evaluation board of MPC555 from Axiom Manufacturers. The CME 555 performs voltage input digitizing and filtering, and it also provides precise and clean voltage data for frequency estimation. The CME 555 is where the frequency estimation algorithm is located, and it also performs the GPS time synchronization, data communication, signal conditioning, interrupt handling, process scheduling and other functions.

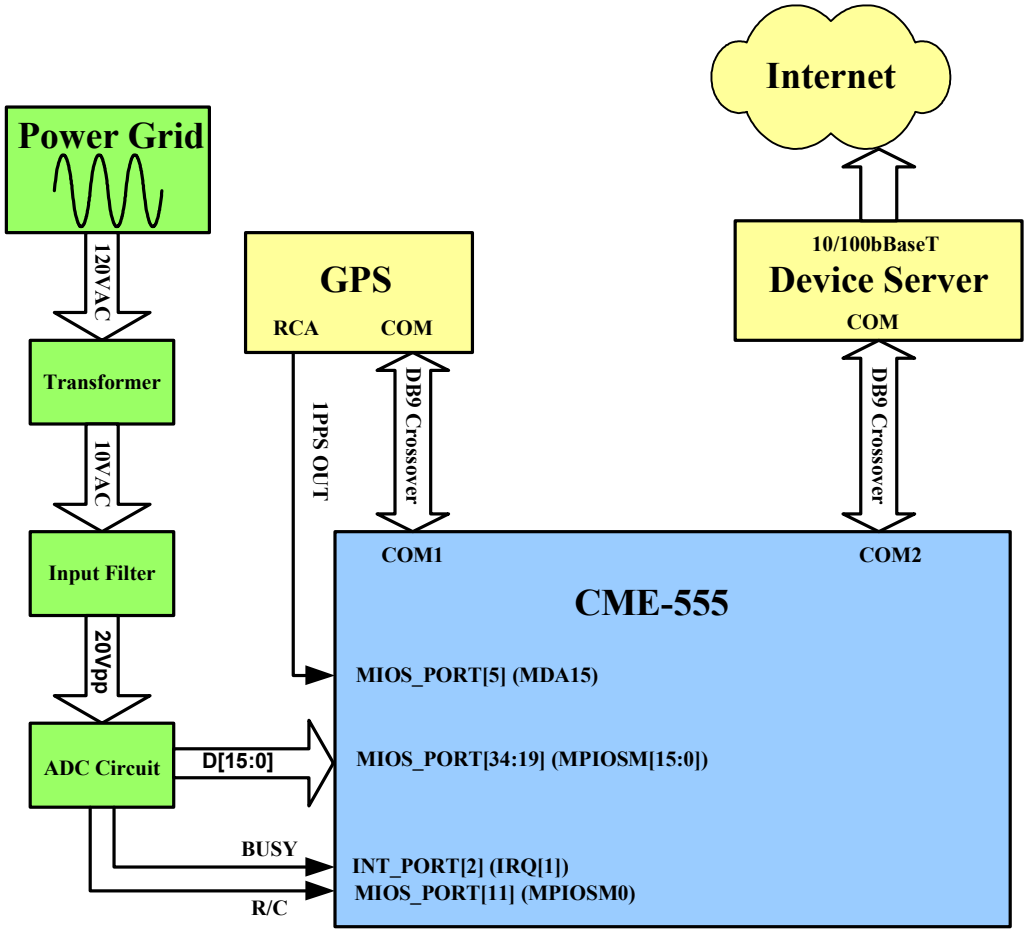


Figure 3.2 System Block Diagram

In the second generation of the FDR, the M12+ timing GPS receiver from Motorola is adopted (On June 2nd, 2005, SiRF Technology Acquires Motorola's GPS Chip Set Product Lines), which provides precise Pulse Per Second signal, and the common timing stamp for different units dispersed in different locations. The timing accuracy of M12+ timing GPS receiver is less than 500ns with SA on, and the Motorola binary I/O commands can be used to initialize, configure, control, and monitor the GPS receiver. With GPS, the frequency estimation of different units is on a common reference if they are sampled at precisely the same instant.

The serial-to-Ethernet converter, the Device Server, makes the device networked. Compared with serial communication, Ethernet communication needs tremendous time and cost to make it reliable. The Device Server saves those efforts, and provides an easy and practical way to make serial communication data be Ethernet data, which can be

transmitted to remote access points via TCP/IP. The Device Server can be configured remotely online; therefore it gives great flexibility for different data collection relocation and analysis applications interface updates.

Figure 3.3 Appearance of Frequency Disturbance Recorder

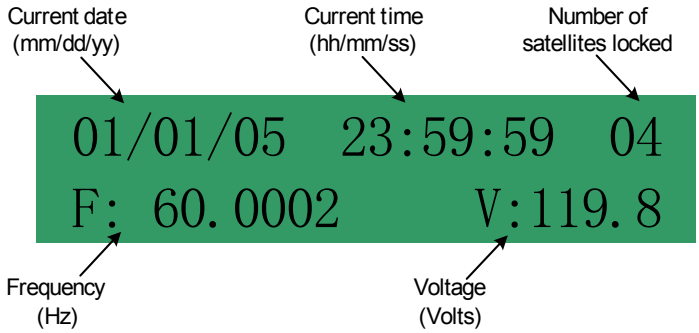


Figure 3.4 LCD panel display

The FDR is lightweight, has a compact size and is free-standing, and Figure 3.3 shows the first commercial FDR. In addition to the data transmitted via internet, on the front of FDR an 20×2 LCD provides the current frequency measurement result updated at a rate of 10 times per second. Figure 3.4 is a demo of the LCD message, which includes current time, frequency, RMS voltage, and the number of satellites locked by GPS. The frequency estimation precision of the second generation FDR is 0.0005 Hz in a lab environment, which can meet most needs of power system monitoring, control and protection applications.

3.3 Phasor Angle Analysis Algorithm Implementation

From (2.4.4) and(2.4.5), one can derive the successive phasor expression:

$$\begin{aligned} X_c^{(k+1)} &= X_c^{(k)} + \frac{2}{N}\left(x_{k+1} - x_{k+1-N}\right)\cos\left(\frac{2\pi}{N}k\right) \\ X_s^{(k+1)} &= X_s^{(k)} - \frac{2}{N}\left(x_{k+1} - x_{k+1-N}\right)\sin\left(\frac{2\pi}{N}k\right) \end{aligned} \tag{3.3.1}$$

$$k = Nf_0 t \tag{3.3.2}$$

where N is the number of phasor angles per cycle and

f_0 is the nominal frequency.

The angle of the kth phasor is given by:

$$\psi(k) = \tan^{-1}\frac{-X_s^{(k)}}{X_c^{(k)}}. \tag{3.3.3}$$

Assuming a quadratic function could express the variation of phasor angles with respect to the sample numbers:

$$\psi(k) = a_0 + a_1 k + a_2 k^2, \tag{3.3.4}$$

a computation window with M phasor angles are used to estimate the value of coefficientsa_0, a_1 anda_2.

$$\begin{aligned} \psi(1) &= a_0 + a_1 1 + a_2 1^2 \\ \psi(2) &= a_0 + a_1 2 + a_2 2^2 \\ &\vdots \quad \vdots \quad \vdots \quad \vdots \\ \psi(M) &= a_0 + a_1 M + a_2 M^2 \end{aligned} \tag{3.3.5}$$

Rewriting equation (3.3.5) in matrix format yields:

$$\begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_M \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2^2 \\ \vdots & \vdots & \vdots \\ 1 & M & M^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \quad \text{or} \quad \psi = \mathrm{Xa}. \tag{3.3.6}$$

Using the least error square solution the matrix a can be solved:

$$\mathbf{a} = \left[\mathbf{X}^T\mathbf{X}\right]^{-1}\mathbf{X}^T\phi. \tag{3.3.7}$$

Taking the derivate of (3.3.4) with respect to k,

$$\frac{d\psi}{dk} = a_1 + 2a_2 k \,. \tag{3.3.8}$$

Taking the derivate of (3.3.2) with respect to t,

$$\frac{dk}{dt} = Nf_0 \,. \tag{3.3.9}$$

So

$$\frac{d\psi}{dt} = \frac{d\psi}{dk}\frac{dk}{dt} = Nf_0 \frac{d\psi}{dk} = Nf_0 \left(a_1 + 2a_2 k\right) = Nf_0 \left(a_1 + 2a_2 Nf_0 t\right) \tag{3.3.10}$$

with (2.4.31) and (2.4.33)

$$\Delta f \approx \frac{1}{2\pi} Nf_0 \left(a_1 + 2a_2 Nf_0 t\right) \tag{3.3.11}$$

$$\frac{df}{dt} \approx \frac{1}{2\pi} 2\left(Nf_0\right)^2 a_2 \,. \tag{3.3.12}$$

The actual power system frequency varies from the DFT fundamental frequency by an amount of Δf . Equations (3.3.11) and (3.3.12) are the main formulas used for frequency estimation in FDRs. It must be remembered that the frequency estimation obtained at the end of the computation window is actually the frequency at the half cycle point before the end of the window. This is because the phasor computed at the end of one cycle actually represents the phasor at the center. Figure 3.5 shows the flow chart of algorithm initialization and the flow chart of main function body.

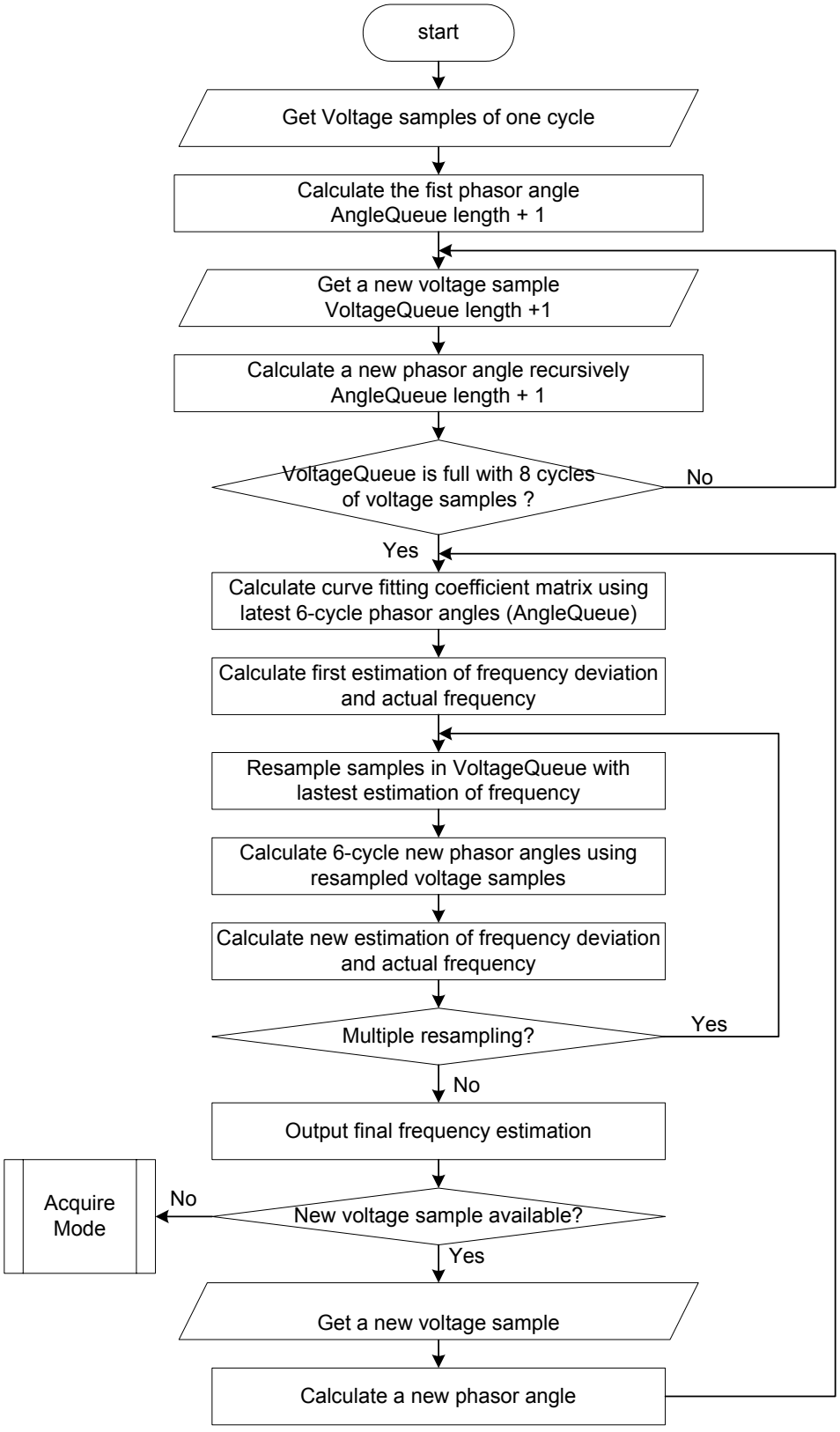


Figure 3.5 Flow chart of frequency estimation algorithm

3.4 Hardware Architecture

As shown in Figure 3.2, the second generation of FDR consists of six major hardware subsystems: a microcontroller system (single board computer), an A/D Converter, a GPS receiver, a serial-to-Ethernet converter (device server), an analog filter, and a power supply. In order to meet the performance goals for the whole system, the selection and integration of these subsystems have consistently been a huge challenge.

3.4.1 Microcontroller

Do we need fixed- or floating-point-based microcontroller (or DSP for 3rd generation) for this system? This is the first question we came up with when we selected the microcontroller. As the terms fixed-point and floating-point indicate, the fundamental difference between the two types of computation is in their respective numeric representations of data. While fixed-point hardware performs strictly integer arithmetic, floating-point hardware supports either integer or real arithmetic. Therefore the fixed-point data has less bit-width, so it is more hardware efficient than the floating-point hardware. In addition, real arithmetic could be coded directly into hardware operations with the floating-point format, while fixed-point devices must implement real arithmetic indirectly through software routines, which adds development time and extra instructions to the algorithm.

The much greater computational power offered by the floating-point unit is normally the critical element in the fixed- or floating-point design decision. However, the floating-point representation has longer bit-width, and thus requires more internal circuitry and more hardware resources. These factors, in addition to the greater number of pins required by the wider data bus, meant a larger die and larger package that resulted in a significant cost premium for the new floating-point devices. Offsetting the cost issue is the fact that the floating-point microcontroller is easier to program; usually they have been adopted for low-volume applications where the time and cost of software development were of greater concern than unit manufacturing costs. A fixed-point

microcontroller is therefore favored for high-volume applications like digitized voice and telecom concentration cards, where unit manufacturing costs have to be kept low.

Microcontroller performance is another important concern for microcontroller selection. It is a common trap to compare microcontroller performance only by crystal frequency. The fact of the matter is that many microcontroller vendors have included bonus architectural features to the microcontroller. Those features enable micros and peripherals run directly off the crystal oscillatory circuitry, which make designs easily meet RT and EMI specifications when the crystal speed is ideally matched to performance.

There are a variety of benchmarks to evaluate microcontroller performance. The Embedded Microprocessor Benchmark Consortium (EEMBC) is one of the most widely used benchmarks to help the designer to decide the best for their application. EEMBC provides an easy way to objectively evaluate both microcontroller performance and compiler efficiency, and offers system designers a valuable tool for selecting the right embedded processors for their systems. EEMBC was founded by a group of semiconductor vendors seeking a method for determining relative levels of processor performance with more representative workloads than Dhrystones. Today EEMBC's members include over fifty of the world's leading microcontroller and microprocessor vendors, such as Freescale, NEC, and AMD.

There are two common classifications for defining the kind of instruction set the processor has: CISC (Complex Instruction Set Computer) means that the processor has a large set of instructions that perform complex tasks, and RISC (Reduced Instruction Set Computer) means that that the processor has quite a small set of instructions so that they do less per command (complicated operations are done by combining many simple instructions to perform a complicated task). Some processors are CISC-based (like x86, Z80etc.), while others are RISC-based (like ARM, MIPS, PowerPC).

In today's embedded microcontroller market, ARM, MIPS and PowerPC are the three most popular RISC computer architectures. The ARM architecture, originally the Acorn

RISC Machine, was started in 1983 as a development project at Acorn Computer Ltd. Due to its power-saving features, ARM variants are in a variety of embedded and low-power applications; ARM7TDMI has been the most successful implementation with hundreds of millions sold in mobile phones and handheld video game systems. ARM's main business has always been to sell IP cores, and many licensees, such as Freescale, IBM, TI, and Intel, have licensed the basic ARM core for various uses.

MIPS (Microprocessor without interlocked pipeline stages), is another popular RICS computer architecture developed by MIPS Computer Systems Inc. The latest of these, MIPS 32/64 Release 2, defines a control register set as well as the instruction set. Several "add-on" extensions are also available, including MIPS-3D, which is a simple set of floating-point SIMD instructions dedicated to common 3D tasks; MDMX, which is a more extensive integer SIMD instruction set using the 64-bit floating-point registers; and MIPS16, which adds compression to the instruction stream to make programs smaller. The use of MIPS designs is widespread in SGI's computer product line, Windows CE devices, and Cisco routers.

PowerPC is a RISC microprocessor architecture created by the 1991 Apple-IBM-Motorola alliance, known as AIM. PowerPC processors bring the processor's local bus to the chip's surface, and connect to a bridge chip that translate this into other on-board device buses that attach to RAM, PCI, and other devices. Thirty-two-bit PowerPC processors have been a favorite of embedded computer designers. Originally intended for personal computers, PowerPC CPUs have since become popular embedded and high-performance processors as well. Besides Apple's Macintosh lines from 1994-2005, the greatest success of PowerPC architecture, PowerPC processors have been used in many embedded products and will be contained in the sixth-generation consoles of the three most popular game consoles: Sony's PlayStation 3, Microsoft's Xbox 360, and the Nintendo Revolution console.

The MPC555 is a high-speed 32-bit central processing unit that contains a floating-point unit designed to accelerate the advanced algorithms necessary to support complex

applications. The MPC555's 32-bit core—compliant with the PowerPC instruction set architecture—supports a wide range of on-board peripherals and offers excellent functionality and performance. Its high level of flexibility is combined with low development costs and quick time-to-market cycles. In addition, code compatibility and scalability among the MPC500 Family members eases the migration process and offers software reuse between family members.

The automotive, industrial control, avionics and robotics market have been major drivers of the MPC555, so it is highly reliable and durable, and can withstand harsh environmental conditions. The combination of its features makes the MPC555 well suited for automotive applications such as engine and transmission control as well as robotics and avionics control. Nonetheless, MPC555s, like their counterparts, are very inexpensive and are able to deliver powerful features that would be too costly to implement otherwise. High-performance data manipulation capabilities and a large on-chip FLASH memory with powerful peripheral subsystems allow for the flexibility to change or upgrade the product late in the production cycle.

As shown in Figure 3.6 of the MPC555 block diagram, the MPC555 has following features:

- 26 Kbytes of Static RAM,
- 448 Kbytes Flash EEPROM Memory with 5-V programming (CMF),
- Flexible Memory Protection Unit,
- General-Purpose I/O Support,
- Dual Time Processor Units (TPU3),
- 18-Channel Modular I/O System (MIOS1),
- Dual Queued Analog-to-Digital Converter Modules (QADC),
- Dual CAN 2.0B Controller Modules (TouCANs),
- Queued Serial Multi-Channel Module (QSMCM), and
- U-Bus System Interface Unit (USIU).

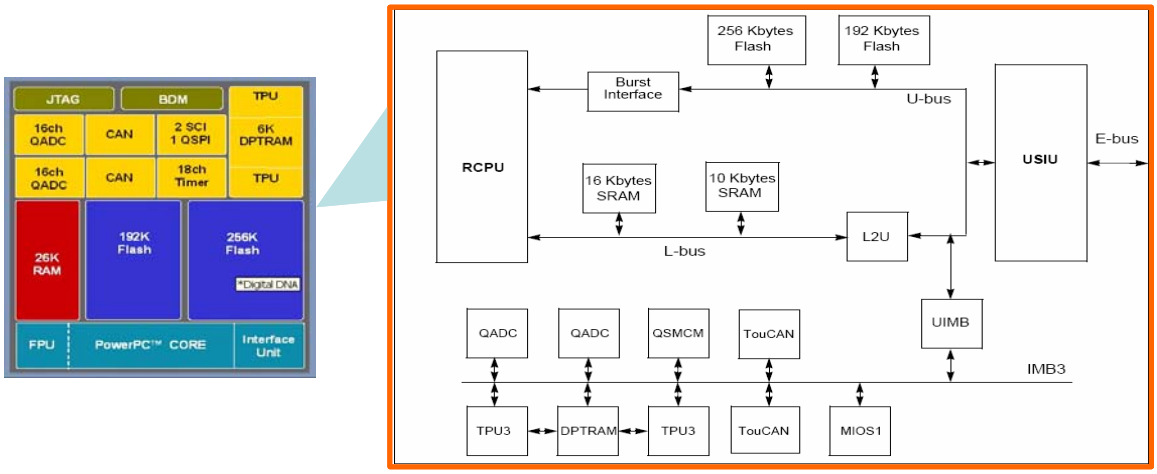


Figure 3.6 MPC555 block diagram

The Time Processor Units (TPU) module consists of 16-bit time bases, sixteen independent timer channels, a task scheduler, a microengine, a host interface, and other pre-programmed functions microcoded in ROM. Two 16-bit counters provide reference time bases for all output comparison and input capture events. Each channel can use either the same time base for match and capture, or use one time base for match and the other for capture. The host interface allows the host CPU to control the operation of the TPU. The scheduler determines the order in which channels are serviced based on channel number and assigned priority. The microengine is composed of a control store and an execution unit.

The modular I/O system (MIOS) consists of a library of flexible I/O and timer functions including an I/O port, counters, input capture, output compare, pulse and period measurement, and PWM. Because the MIOS is composed of submodules, it is easily configurable for different kinds of applications. The MIOS1 is composed of the following submodules: one MIOS bus interface submodule (MBISM), one MIOS counter prescaler submodule (MCPSM), two MIOS modulus counter submodules (MMCSM), ten MIOS double action submodules (MDASM), eight MIOS pulse width modulation submodules (MPWMSM), one MIOS 16-bit parallel port I/O submodule (MPIOSM), and two MIOS interrupt request submodules (MIRSM).

The Serial Multi-Channel Module (QSMCM) provides three serial communication interfaces: the queued serial peripheral interface (QSPI) and two serial communications interfaces (SCI1 and SCI2). These submodules communicate with the CPU via a common slave bus interface unit (SBIU).

The unified system interface unit (USIU) of the MPC555 controls system start-up, system initialization and operation, system protection, and the external system bus. The MPC555 USIU functions include system configuration and protection, an interrupt controller, system reset monitoring and generation, a clock synthesizer, power management, external bus interface (EBI) control, a memory controller, and debug support.

Embedded programming implies programming a system in which resources are limited, and the cost should be kept low on high-volume competitive products. The MPC555 is bundled into a system-on-chip (SOC) integrated circuit, and contains the processor core, cache and the processor's local data on-chip, along with clocking, timers, memory (SDRAM, RAM, FLASH), peripheral (serial, I/O), and bus (PCI, PCI-X, ROM/Flash bus, I2C, CAN) controllers. So the MPC555 has been selected as the computation unit, which calculates the frequency; and the timing message from the GPS, along with the sample number at the beginning of a window, is assigned to the frequency measurement as its identifying tag.

3.4.2 Analog to Digital Converter

The Analog-to-Digital Converter is a device that processes the conversion of analog signals to a digital representation. The Integrating ADC (Dual Slope), FLASH ADC (Parallel ADC), Pipelined ADC, Sigma Delta ADC, and SAR (Successive Approximation Register) ADC are the five most widelyused ADC architectures on the market.

Integrating ADC, also known as Dual-Slope ADC, integrates and compares input voltage against a known reference value to implement the analog to digital conversion. Integrating ADCs are extremely slow devices with low input bandwidths, but their ability to reject high-frequency noise and fixed low frequencies such as 50Hz or 60Hz makes

them useful in noisy industrial environments and applications. Integrating ADCs provide 10 to 18-bit resolution, and the conversion time for a medium speed of a 12-bit integrating ADC is about 20mS. This type of ADC can be used in many portable instrument applications, including digital panel meters and digital multi-meters.

Flash ADCs are implemented by cascading high-speed comparators. Each comparator represents one LSB, and the output code can be determined in one compare cycle. Flash ADCs, also known as parallel ADCs, are the fastest way to convert an analog signal with large bandwidths to a digital signal. However, this type of ADC has relatively low resolution, consumes more power, and can be quite expensive. This limits them to high-frequency applications that typically cannot be addressed any other way. Examples include data acquisition, satellite communication, radar processing, sampling oscilloscopes, and high-density disk drives.

The Pipelined ADC consists of numerous consecutive stages, each containing a track/hold (T/H), a low-resolution ADC and DAC, and a summing circuit that includes an inter-stage amplifier to provide gain. Pipeline ADCs provide high speeds, few Msps to 100+ Msps, 8-bit to 16-bit resolution, and lower power consumption than flash ADCs. The Pipeline ADC is an optimum balance of size, speed, resolution, power dissipation, and analog design effort, and has become increasingly attractive to major data-converter manufacturers and their designers.

The Sigma Delta ADC, also known as the oversampling ADC, consists of two major blocks: a modulator and a digital filter. The architecture of the modulator is similar to that of a dual-slope ADC, and the output filter converts the bit stream to a sequence of parallel digital words at the sampling rate. The Sigma Delta ADC is good for applications with a bandwidth up to 1MHz, and it provides high resolution, high accuracy, low noise, and low cost. However, sigma-delta has inherent disadvantages; the filter does not provide attenuation at integer multiples of the modulator sampling frequency, the complexity of the digital circuitry limits the speed of conversion, the digital filtering results in long latency between the start of the sampling cycle and the first valid digital

output, and there is a lag between digital outputs and their corresponding sampling instants.

The SAR ADC, also known as bit-weighing ADC, employs a comparator to weigh the applied input voltage against the output of an N-bit digital-to-analog converter (DAC). Using the DAC output as a reference, this process approaches the final result as a sum of N weighting steps, in which each step is a single-bit conversion. Initially all bits of SAR are set to 0. Then, beginning with the most significant bit, each bit is set to 1 sequentially if the DAC output does not exceed the input signal voltage; otherwise it is set back to 0. It is kind of a binary search, and for an N-bit ADC, N steps are required. The SAR ADC has medium to high resolution (8 to 16-bit), draws low power, and provides a small size package and the lowest production cost. However, conversion speed is limited to less than 5Msps, and an anti-aliasing filter is required for some applications.

The selection of ADC is a tradeoff between resolution, channel count, power consumption, size, conversion time, static performance, dynamic performance, and price. In order to implement and verify the best performance of a frequency estimation algorithm, the throughput rate of the ideal ADC in FDR should be higher than 100kSPS; the minimum resolution should be 12 bits; the conversion time, power consumption, and size of the ADC should be minimized to meet portable real-time system requirements; and the price should be within a reasonable range. The AD976A from Analog Device is 16-bit SAR ADC with 200kSPS throughput, 100mW Max Power Dissipation and has a high-speed parallel interface; it was therefore selected as the best candidate for FDR implementation.

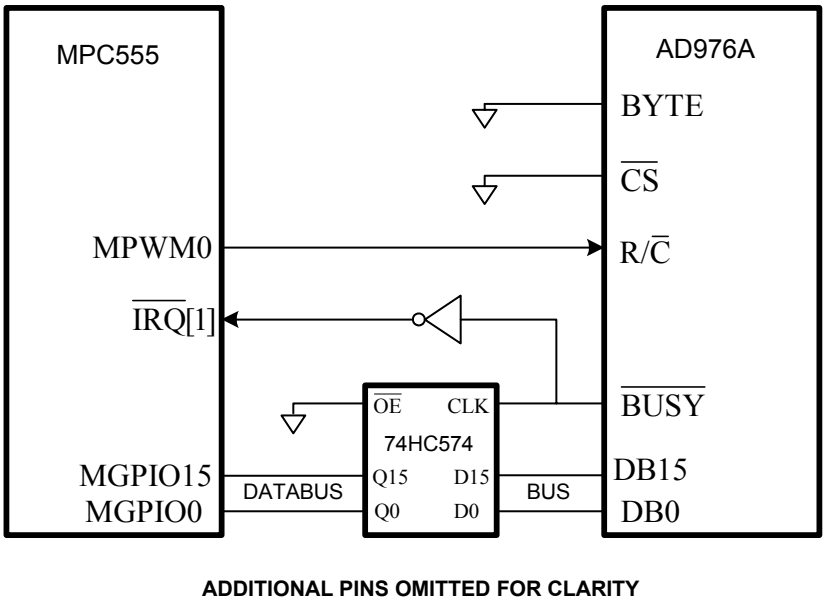


Figure 3.7 AD976A and MPC555 interface

Since the input range of AD976A is from –10V to +10VDC, a transformer is necessary to step-down the 110-120VAC raw input from residential wall outlets. A signal transformer deployed here can be helpful to clean up the polluted input directly from the power grid. In order to minimize the impact of high-frequency noise and harmonics, an optimized analog filter, as discussed in a later section, is adopted to work with the AD976A to obtain the best input for further frequency estimation.

To control the conversion sequence of AD976A, a PWM (Pulse Width Modulation) signal from the MIOS1 (Modular Input Output System) subsystem of the MPC555 triggers the R/$\overline{C}$ (Read/Conversion) line. The PWM module can be programmed online to obtain the ideal sampling rate of AD976A. Any inaccuracy or drift in the MPC555's oscillator is removed by measuring the one pulse-per-second (PPS) signal from the GPS receiver using the MIOS Double Action Subsystem Module (MDASM). The $\overline{BUSY}$ signal is inverted and then connected to pin $\overline{INT}$ of the microcontroller. Finally, the data bus pins are connected to the GPIO (General Purpose IO) pins on the MPC555.

3.4.3 GPS receiver

In the past several years, with the recognition of the importance of knowing frequency at different locations for system performance monitoring and analysis, many attempts have been made to synchronize frequency measurement [28-30]. Radio broadcasts are probably the least expensive way to share this information, but are the most susceptible to interference and have the lowest accuracy. For an instance, AM broadcasts WWV, WWVB and WWVH provided by the US government are typically around 1ms, which is not accurate enough for power system applications. Fiber optic and microwave systems can deliver synchronization accuracy at 1us and better. However, this solution is expensive for implementation and maintenance due to the specialized interface or separate communication path. Satellite broadcast has significant advantages over radio broadcast, fiber and microwave timing systems. Satellite has a wide area of coverage with continuous reference to the national standard, and the satellite signal is little affected by environmental variations. Because most satellite systems are sponsored for their primary functions other than timing synchronization, the cost is low. The satellite systems that have been widely used include GOES, GPS, and GLONASS; INMARSAT and GALILEO are two possible alternatives of GPS in the near future.

The GOES (Geostationary Operational Environmental Satellite) is a satellite system used primarily for weather monitoring. It is operated by NOAA's National Environmental Satellite, Data, and Information Service in Suitland, Maryland. GOES is composed of two types of satellites: geostationary operational environmental satellites (GOES) for national, regional, short-range warning and "now-casting," and polar-orbiting environmental satellites (POES) for global, long-term forecasting and environmental monitoring. Usually UHF receivers are tuned to GOES' down link (468MHz), which suffers interference problems with mobile communications and outages due to solar eclipses. The time synchronization accuracy of GOES referenced to UTC is with a base of 25us, although a more realistic operating accuracy is 100us.

GPS is a satellite-based radio navigation system developed and operated by the US Department of Defense (DOD). Completed in 1994, the GPS system consists of 24

Navstar satellites, which are arranged into six orbit planes at a height of 20,200 Km and circle the earth twice a day. With no obstruction, there are typically 8-12 satellites visible at any one time from anywhere on earth. Navigation information is broadcast on three L-band frequencies: L1 at 1575.42MHz, L2 at 1227.6MHz and L5 at 1,176.45MHz. The signals from the satellites available for commercial users are called C/A (Coarse Acquisition) codes, and the encrypted signals only for military applications are referred to as the P-Code signal (Private Code). In 2000 the US government removed Selective Access (SA) from the C/A code. SA was a jittering of the satellite timing clocks as represented in the satellite ephemeris. Without SA, GPS accuracy is around 10m over a long period and has a repeatability of around 1m over short periods. Access to CA code signals is available to anyone and will be provided free of charge for a minimum of ten years.

From the information transmitted by the Ephemeris and Almanac, a GPS receiver can determine how long it took the transmitted signal to reach it. That time is proportional to the distance the signal traveled from the satellite (its range) so it can be used to determine an arc on which the receiver must lie. Calculating the intersection point of a number of such arcs derived from different satellites provides a solution to the receiver's position on the surface of the earth. With a GPS receiver that costs less than a few hundred dollars you can instantly learn your location on the planet--your latitude, longitude, and even altitude--to within a few hundred feet.

Each satellite contains a Rubidium atomic clock, which is precise to within a billionth of a second, so between them they represent an extremely accurate time standard available for synchronization at any point on the earth. Furthermore, with the Cesium clock referenced to Coordinated Universal Time (UTC), the ground-based GPS control station monitors the satellite clocks and transmits corrections for these and other parameters necessary to maintain the accuracy of the system. With this continuous adjustment, the timing accuracy of GPS is limited only by short-term signal reception, whose basic accuracy is 0.2 μs. It is this accurate timing that leads to an application of the GPS satellites separate from their function for navigation. The world's cellular and fiber

communications use the time information derived from the GPS satellites for clock synchronization. The business world also uses this reference to ensure that large global financial contracts are executed at precisely the same time at both ends of the transaction.

The gross inaccuracies in the system come primarily from the changes of the speed of radio waves traveling through the different parts of the ionosphere traversed by signals from each of the satellites. This means GPS accuracy is determined by the amount of ionosphere variation. This is most noticeable at dawn and evening when the temperature in the ionosphere changes as the sun crosses the horizon. In the near future, the addition of the second GPS signal L5 is intended to allow these changes to be removed in the calculation of position, because the change in the velocity of radio waves through the ionosphere varies in a more or less predictable way with frequency.

Various means, such as differential correction and WAAS (Wide Area Augmentation Signal), have been devised to improve these accuracies, but they have been made redundant in many systems because of the improvement in accuracy coming from the removal of SA. In order to meet the fast-growing civilian and military demands of GPS, the United States will upgrade the current GPS system. A variety of other enhancements were under consideration, including increased power, the addition of a new military code at the L1 and L2 frequencies, additional ground stations, more frequent uploads, and an increase in the number of satellites. Adding a third carrier signal designated L5 at 1,176.45MHz in the block IIF constellation is one of the most noticeable enhancements, which will be intended for civilian applications in air traffic control.

GLONASS (Global Orbiting Navigation Satellite System) is the Russian satellite-based radio navigation system. GLONASS is a counterpart to the GPS, and both systems share the same principles in the data transmission and positioning methods. The accuracy of time synchronization is as good as un-degraded GPS, and the commercial GLONASS receivers are also available for various applications. Hence GLONASS could be an alternative of GPS for some critical applications. However, the GLONASS constellation

is currently operating in a degraded mode, and the long-term performance of GLONASS is uncertain.

INMARSAT (International Maritime Satellite System) consists of a fleet of eleven geosynchronous telecommunications satellites, and provides telephony and data services to users world-wide via special GPS-like digital radios called "terminals". INMARSAT is a commercial enterprise, so it could be a little risky to develop applications based on INMARSAT for the long term. A real drawback of INMARSAT is that geostationary signals are very low on the horizon in the northern latitudes, making reception difficult at some sites.

Europeans are planning a new system called GALILEO that is scheduled to roll out in 2008. GALILEO is designed to integrate both GPS and GLONASS to become a truly GNSS solution or to operate independently, and aims to provide similar or better levels of accuracy. The GALILEO GPS system, while appearing to be controversial, is being proposed for technical, economic and sovereignty reasons, which in the long run can impact European scientists and civilian users as well as those interested in high-quality positioning and navigation applications.

Overall, neither GOES, GPS, GLONASS, INMARSAT, nor GALILEO could be called a good choice, as those technologies put severe limitations on availability and accuracy of synchronization. GPS can provide sufficient time accuracy, availability, and reliability to meet the power system applications requirements, and GPS also provides the indication of loss of synchronization. It is the most ideal synchronization source, having continuous uninterrupted availability, and can be accessible to any site on the earth, which means large geographic areas of the entire interconnected system can be spanned in the measurements. GPS is the most promising candidate for synchronization frequency measurement with a synchronization error of less than 1 microsecond.

The basic synchronized signal of GPS is the 1PPS pulse train, where the rising edge of the pulse marks the second change of UTC time, and the pulses that trigger data sampling

should be locked in phase with the 1PPS. The measured frequency should maintain a timing accuracy of 1μs between the synchronized pulses. The 1μs synchronization accuracy corresponds to an angular accuracy of 0.022 degrees for a 60Hz system, and 0.018 degrees for a 50Hz system. For most general applications, a 0.5 microsecond accuracy is a safer figure on which to depend [29].

In order to minimize the size and cost of FDR, the OEM GPS receiver turns out to be the best solution in our case. The basic hardware requirements for the system's GPS receiver are as follows:

- The GPS unit shall provide the pulse-per-second signal to the MIOS module of the MPC555 to synchronize the PWM pulse train, which triggers each analog-to-digital conversion of ADC. Each A/D conversion result shall provide subsecond information in order to synchronize the frequency measurement.
- The GPS unit shall provide a UTC timestamp to the CPU for each second of time. The UTC timestamp also shall also be formatted into second-of-century (SOC), which is referenced to the midnight (00:00) of January 1, 1970. The timestamp must be provided, as an unsigned long integer, to the TCP/IP communications subsystem, where it will be attached to each outgoing sample set.
- The GPS shall poll its own locations: latitude, longitude, and altitude, which is essential for FNET applications, such as the event location triangulation.

Several OEM GPS receivers from different vendors are evaluated; including the Garmin GPS 25 series and GPS 15 series, the Trimble Lassen serials, and Motorola Oncore serials (SiRF acquired Motorola's GPS chipset on Jun 2, 2005). Finally, we chose the Motorola M12+ Timing GPS receiver, which has following advantages:

- Has small dimensions of 40.0 x 60.0 x 10.0mm (1.57 x 2.36 x 0.39in.) with low power consumption (< 185mW @ 3V without antenna).
- Uses an evaluation board (for FDR Generation 2), the unit is much less costly (<$200 versus about $800 for the Trimble).
- Provides consistently good GPS reception, even using a similar antenna and antenna placement. The M12+ timing receiver can track 12 simultaneous satellites

theoretically, and in our real applications it can track 4-6 satellites consistently for accurate time and position information.

- Has several advanced timing features, such as Motorola's Time RAIM (Receiver Autonomous Integrity Monitoring) algorithm, that allows us to determine the reliability of our measurements.
- Allows us to configure the PPS output to different modes in order to meet different demands for testing purpose or for maximum timing accuracy. Effectively, it provides accurate, error-corrected time with a minimum of one satellite after a position hold is achieved.
- Provides time information every second without a query, after it is set up as such. During testing, this timing data was unerringly consistent.

It provides 5V active high PPS output through an RCA connector and provides an RS232 connector for communication with the microcontroller.

Since the first commercial FDR was installed in November 2003, the Motorola M12+ Timing GPS receiver has consistently provided the accuracy and performance required by FNET applications. On Jun 2, 2005 SiRF acquired Motorola's GPS chipset, and the possible alternatives to Motorola M12+ Timing GPS receiver have been narrowed down.

3.4.4 Signal Conditioning System

Besides the errors introduced by data sampling, some other factors may have great effects on the accuracy of frequency measurement. Anti-aliasing filters, harmonics, phase shifts and magnitude shifts can significantly effect the measurement. For instance, without anti-aliasing filters, harmonics are aliased directly into the measurement. Another example is that a 2-pole R-C filter with a cutoff frequency of 300Hz will have a phase shift of about 22 degrees and a 2% magnitude roll-off at 60Hz.

The signal conditioning system is required to alter the transformer output for use with the external A/D converter. This combines a voltage divider, a passive low-pass filter, and an output bias circuit. First, the system must take a 12Vrms signal from the output of the transformer and attenuate it to 20V peak-to-peak. Also, according to the Nyquist

Theorem, to obtain a faithful representation of a signal the highest frequency of the signal should be less than the half of the sampling rate. Otherwise the signal with a frequency higher than half of the sampling rate will be aliased down to look like lower frequency components. Hence the signal conditioning system must include a second-order anti-aliasing filter. Generally, anti-aliasing filters are low-pass filters with a cut-off frequency equal to one-half the sampling rate. This provides the low-pass filtering function, which can block the signal components with frequencies higher than those that can be properly reproduced by a given sampling rate.

An anti-aliasing filter could be passive, which consists of resistors and capacitors exclusively, or active, utilizing op-amps. As some buffering between the filters and the ADC is generally necessary, an op-amp is needed in any case, and one could use the active filter design, which leads to a smaller circuit size.

The design of the filter is in any case determined by the sharpness of cut-off in the stop band and the transient response of the filter. In general, if filters with very sharp cut-off frequencies are employed, they produce a longer time delay in their step response [31]. In most computer applications, two-stage RC filters are found to provide an acceptable compromise between sharpness of the cut-off characteristics in the stop band and the time delay in their step response [32]. Two-stage RC filters are quite popular because of their simplicity, passive components, and reasonable frequency response, though they suffer from the disadvantage that they produce a rounded characteristic at the beginning of the stop band.

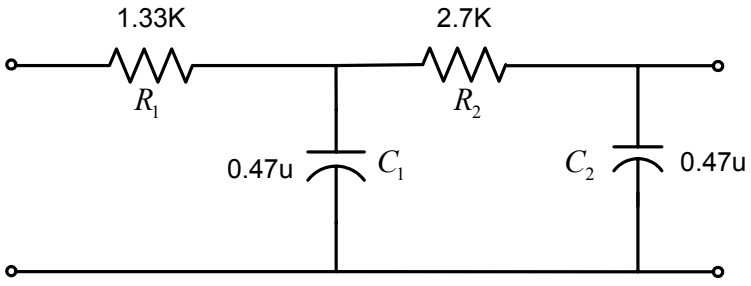


Figure 3.8 Anti-aliasing filter

A two-stage RC filter suitable for a sampling process using a sampling rate of 1440Hz is shown in Figure 3.8. With a specified DC gain of unity, this filter has a cutoff frequency of 720Hz, and the transfer function

$$H(s) = \frac{1}{R_1C_1R_2C_2s^2 + (R_1C_1 + R_1C_2 + R_2C_2)s + 1}. \quad (3.4.1)$$

R_1, R_2, C_1, C_2 are the components of filter as shown in Figure 3.8. The frequency response and step response of this two-stage filter are shown in Figure 3.9 and Figure 3.10. As can be seen, the step response produces an essentially correct output in about 0.8 milliseconds after application of the step function. The phase lag at nominal frequency 60Hz is around 5.4131 degrees, which corresponds to a time delay of about 0.25 milliseconds.

Another consideration in filter design is the stability of its transfer function in the presence of a variation of component value due to aging and temperature variations. The gain magnitude and phase angle of an active filter are more sensitive to variations in component values as compared to those of a passive filter. In order to keep the small variations of components, a high-precision metal film resistor and polycarbonate capacitors should be selected.

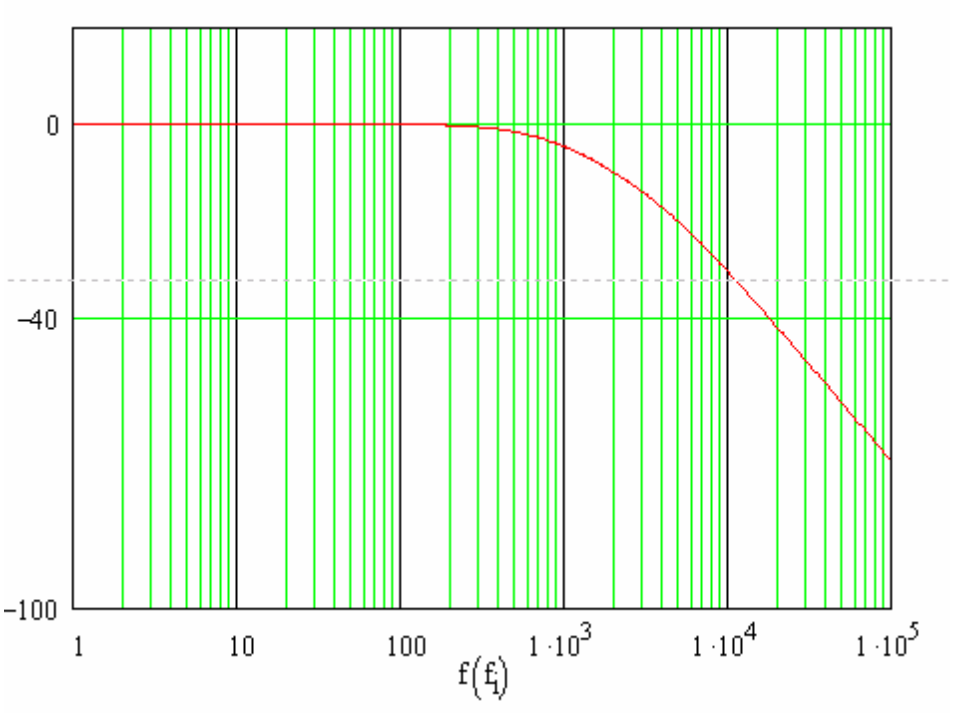


Figure 3.9 Frequency response

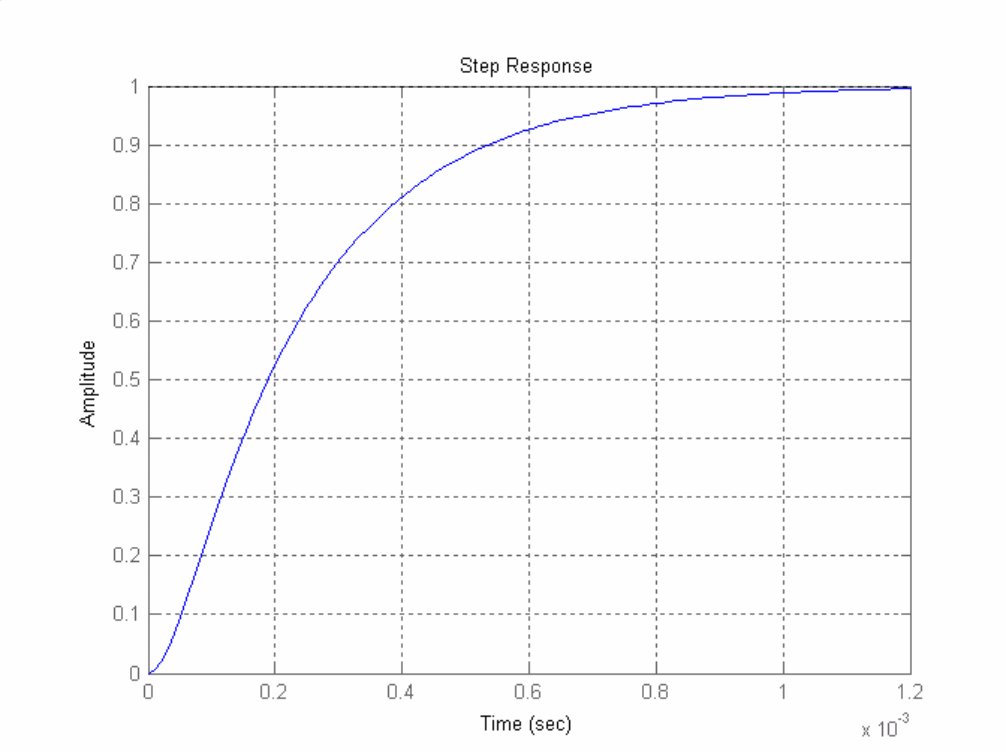


Figure 3.10 Step response

3.4.5 Serial to Ethernet Converter

Internet connectivity is one of the most important design goals of FDR. Running the TCP/IP stack software on a microcontroller for the Internet connection is an intuitive solution. However, most reliable TCP/IP stacks are not easy obtained due to intellectual property or cost issues, and the adoption and development of TCP/IP stack software is full of challenges and time-consuming. Another consideration is that the MPC555 has two built-in serial communication modules, which provide an easy and fast-to-develop serial communication functionality of FDR. So in order to accomplish a realistic and reliable implementation of FDR within a limited time, we determined an alternative that is also a more effective solution by using a device server.

The serial device server, also known as a serial-to-Ethernet converter, allows companies to connection of legacy serial devices to an Ethernet LAN/WAN, providing many more options for data acquisition, device management, and industrial control than would otherwise be available. Generally a serial device server provides network features, such

as a Windows driver, TCP/UDP Client/Server, Ethernet modem, and pair connection, and takes into account agency approvals.

As shown in Figure 3.11, with minimized development efforts, by using the serial device server the frequency measurement data is assembled in a message stream to be communicated to the central server via TCP/IP protocol in a secure and reliable way. However, it needs to be noted that because the algorithm communicates binary data to the server, no special character sequences could be reserved for flow control. Therefore, the TCP/IP unit must support CTS/RTS handshaking.

The measurement data of the FDR are combined into a character array, and transmitted from the FDR to the device server. The character array has the format shown in Table 1. The size of UnitID varies, since the number of FDRs could be from 1 to 255. And the size could be even larger if more FDRs are added into FNET. The size of time_Date and currentTime also varies, since the conversion from integer number to character ignores zeros to save the storage space and data transmission time.

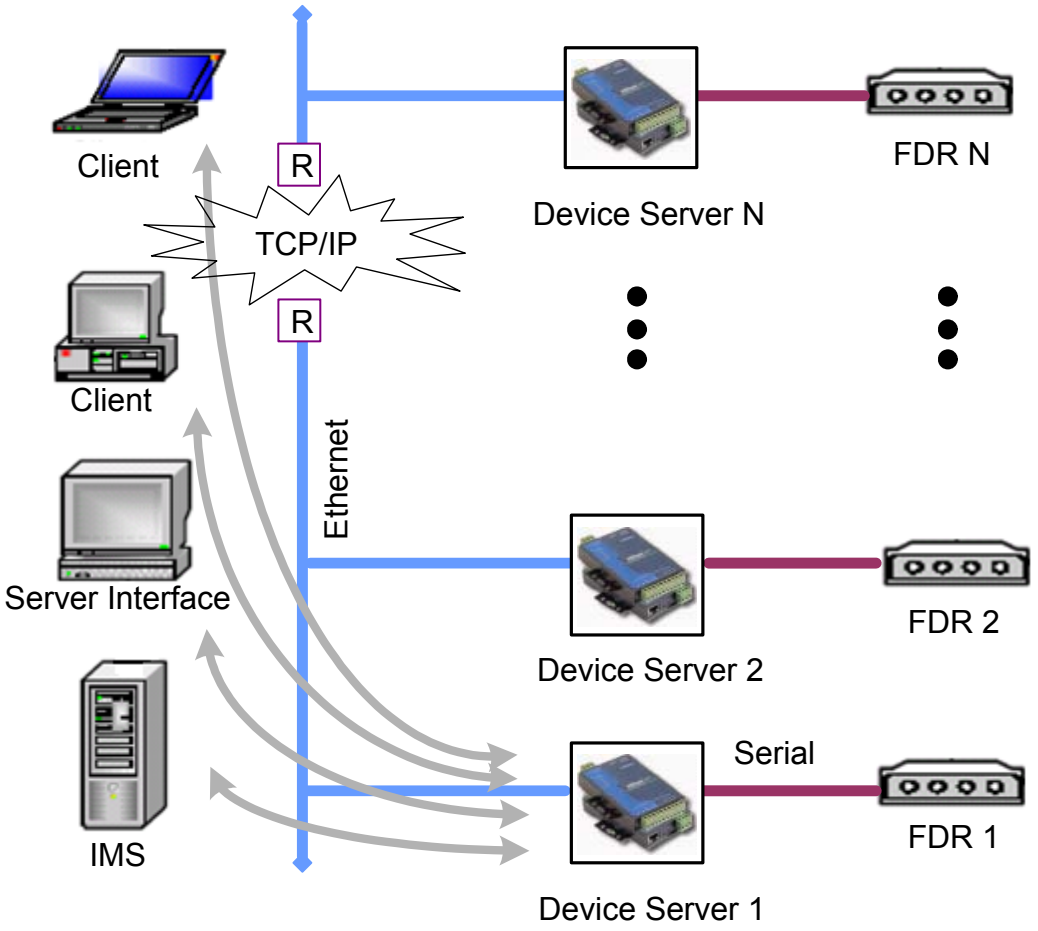


Figure 3.11 Device server connection

Table 1 Format of character array from FDR

Name	Description	Size(Bytes)
UnitID	Unit identification number	1~3 *
time_Date	Current date	3~6 **
currentTime	Current time	1~6 ***
convNum	Number of frequency result	1
firstFreq	Frequency result before resampling	7
finalFreq	Frequency result after resampling	7
voltageMag	Voltage magnitude	8
voltageAng	Voltage angle	6

* The size depends on the number of the UnitID, which in a range from 1 to 255

** Zeros in time_Date variable are ignored when it's converted to a character array

*** Zeros in currentTime variable are ignored when it's converted to a character array

Serial device servers have a fairly short history, and were first developed by Moxa in 1994 [33]. With the prevalence of Ethernet LAN/WANs in both business and industrial settings, the serial device server products have begun to take off and become more popular throughout the years, and will be a major player in communications and industrial automation markets. Moxa, Lantronix, and Digi are three big names in the serial device server market, and each of them provide suitable products for all kinds of wireless, security, commercial and industrial applications.

The Moxa NPort 5110 is a good choice for FDR so far, and it provides a DB9 serial interface to the CPU and an RJ-45 Ethernet interface to a TCP/IP network. The main advantages of using the NPort 5110 are:

- Up to 4 connections can be established between NPort 5110 and hosts,
- Destination IP address parameters can use both IP address and domain name,
- Auto-detecting 10/100 Mbps Ethernet,
- Built-in 15 KV ESD protections for all serial signals,
- Configuration via web/Telnet/serial console,
- Very large and flexible feature set, and
- Large internal buffer to accommodate fluctuation in network throughput.

Overall, the NPort 5110 provides satisfied performance for the FDR, although some improvements could make it more reliable. For instance, although the NPort 5110 is very configurable, its interfaces are designed for human interaction. This is not ideal for an embedded application, and causes some problems, such as relocations. The connection speed or throughput of the NPort 5110 may be low if one of the four connections is slow, since the one slow connection will slow down the other three connections. With a fast-growing user group and new application demands, serial device server vendors keep upgrading their current products and releasing new products fairly quickly. We believe

that in the near future the serial device server can work more efficiently, reliably and be more cost-effective.

Aside from the initial configuration, the NPort 5110's use and operation is quite simple. If the unit has been set up correctly, any data sent to the unit's serial port is passed through to the network. The NPort 5110 automatically wraps data in the appropriate TCP/IP headers, pulling information such as the remote address from its configuration as needed. Although the NPort 5110 is not programmed per se, it must still be configured to work properly with the FDR for each re-location. For Generation 2, the NPort 5110 unit is separate from the rest of the FDR, which will make such configuration easier. Before shipping a complete unit, configuration of the NPort 5110 is required by the user. However, if any changes need to be made, the instructions are very easy to follow and the configuration can be done via web, Telnet or serial console.

Aside from the difference between the products from different vendors, there are several common parameters to be configured for serial device server. Below is a short description and explanations of each parameter and the chosen value:

Baud rate:

Baud rate is the maximum number of bits that are transmitted per second. Usually it could be between 110bps and 230.4Kbps. For optimal speed of communication, the baud rate in our application is set to 57600bps, which is the maximum common speed between the NPort5110 and the CPU.

Interface Mode:

Interface mode refers to the serial communications characteristics. The interface mode specifies line levels, frame size, parity, and stop bits. We specify RS-232C, 8 data bits, no parity, and 1 stop bit for our FDR application.

Flow Control:

Flow control refers to the handshake method for stopping serial communications. As discussed above, our application requires hardware flow control. Therefore, the flow control parameter is set to RTS/CTS handshake.

Connect Mode:

The connect mode defines how the unit makes a connection and how it reacts to incoming connections over the network. TCP Client Mode is chosen to indicate that the unit should establish a TCP connection and transfer data automatically to a pre-defined host computer when serial data arrives.

Destination IP Address:

The FDR always connects to one or multiple single central servers. The IP addresses of those destination central servers are entered here. The destination IP address parameters can use both the IP address and domain name.

Port Number:

Port number refers to the local TCP/IP port used for network identification. According to the Internet Assigned Numbers Authority [34], there are enough unassigned or available port numbers that could be used in FNET, such as 8687-8698, 8700-8732 and 8734-8762.

Force Transmit:

Force Transmit allows us to specify how often the NPort 5110 should fetch the serial data from its internal buffer, and send the packets to the Ethernet. The NPort 5110 stores the data coming in through the serial port in the internal buffer, and transmits the data via TCP/IP, but only if the internal buffer is full or if the force transmit time interval reaches the time specified under force transmit. The force transmit timeout must be at least larger than one character interval within the specified baud rate, and it can be optimized depending on applications. In our case, the serial port is set to 57600bps, 1 start bit, 8 data bits, 1 stop bit, and no parity. The total number of bits needed to send a character is 10 bits, and the time required to transfer one character is

$$(10 \text{ (bits)} / 57600 \text{ (bits/s)}) \times 1000 \text{ (ms/s)} = 0.174 \text{ ms.} \quad (4.2)$$

So the force transmit can be set to 1 millisecond.

3.5 Software Architecture

3.5.1 States of operation

In terms of the system, the FDR has three states of operation: acquisition, initialization, and collection. Figure 3.12 is the top-level state machine of the FDR, and each state is explained in detail below.

Acquisition

The acquisition state refers to the acquisition of adequate GPS satellite signal and receipt of a valid 1PPS signal with UTC time stamp from the GPS receiver. Because all voltage samples and frequency calculations are synchronized with UTC time, the FDR should not proceed until a valid GPS synchronization signal is established. An internal timer is used to verify the accuracy of the 1PPS, to make sure that the time interval between two consecutive 1PPSs is one second plus or minus a pre-set acceptable error. A threshold is set up to determine whether a GPS signal is established. If the FDR has received four consecutive 1PPS signal, and all those 1PPSs are verified by the internal timer, then it considers the valid time synchronization signal has been established. If one 1PPS signal is exceedingly delayed according to the internal timer, then four additional pulses are measured before any further proceeding.

Initialization

Once the FDR establishes the valid time synchronization signal, the unit begins collecting voltage input to seed the frequency algorithm. This initialization process is allocated one second so that synchronized frequency measurements can be made at the start of the next second. If the GPS signal is still valid verified as in acquisition state, the FDR will switch to collection state for further process. If the GPS signal is lost during the initialization process or the loss of synchronization could cause the estimation error out of the present

range, a flag in the measurement data output will be asserted to indicate the loss of time synchronization. The FDR will then reset to the acquisition state and discard all measurements collected. This ensures the FDR is using only current data with known synchronization time information.

Collection

Once the frequency algorithm is initialized, the FDR is capable of generating frequency estimates continuously until either a loss of the voltage source or loss of a GPS signal. If the GPS signal is lost, the FDR will reset to the acquisition state and discard current phasor and frequency calculations. If no valid voltage input is received, the FDR will generate an error code, and switch to acquisition state.

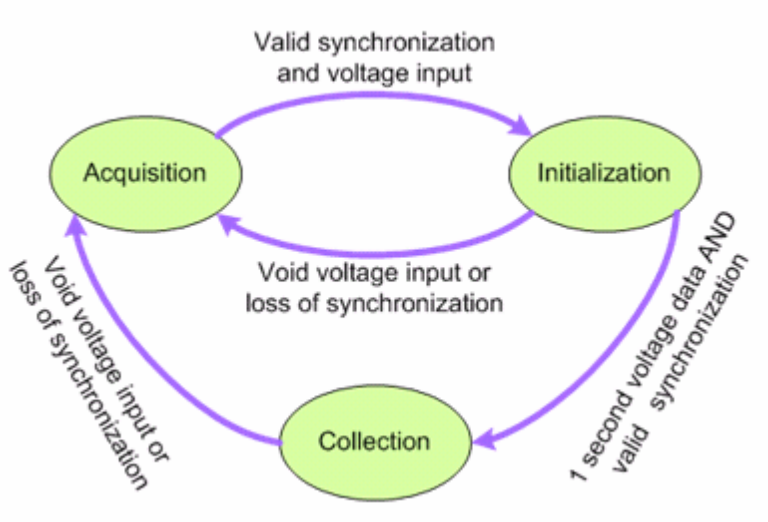


Figure 3.12 Top level state machine of the FDR

Digitizing the voltage input is the first step for frequency measurement. In order to implement a synchronized measurement, the pulses triggering the analog-to-digital conversion should be rigidly locked in phase with the 1PPS signal. In other words, the number of voltage samples per second should be an integer, and those samples should be evenly spaced throughout a second. According to IEEE Std C37.111-1999, there are recommended sampling rates for power system applications [35]. With a higher sampling rate, the FDR can generate more voltage samples for calculation and sophisticated compensation schemes, which may improve the measurement accuracy. However, the more data processing there is, the more computation resources are needed, and this may

raise stringent requirements for real-time performance of FDR. So a tradeoff of 24 Samples/Cycle, 1440 samples per second for a 60Hz system and 1200 samples per second for a 50Hz system is adopted for the FDR. With this sample rate, the voltage input digitizing procedure requires reasonable computational resources, and the accuracy of measurement is still within a satisfied range.

The frequency estimation algorithm operates in a data window of a number of input samples. In order to minimize the DFT spectral leakage error, it is required to sample the input voltage coherently. Coherent sampling means the algorithm operation data window is of an integer number M of cycles, each cycle includes an integer number N of samples. It guarantees a complete, periodic waveform representation exists in the operation data window, and allows most information to be collected about the input signal in the least amount of time. As discussed in the previous paragraph, there are 24 samples per cycle, or N is 24 in the FDR implementation. Some simulation results show that the more cycles there are in the operation data window, the better the estimation accuracy is, although a long operation data window increases the FDR unit computation efforts. So a tradeoff of $M = 6$ is currently adopted in the FDR.

Generally, there are two phases of operation to the estimation algorithm: initialization and continuous collection. In the initialization phase, the algorithm is seeded with voltage data of at least eight cycles, which exceeds the six cycles of operation data window but is needed in order to deploy another improvement discussed later. The first cycle of data is used to calculate the first phasor of the voltage signal. Each point after that is then used to calculate another phasor recursively, where the previous phasor value is used in the calculation. Once eight cycles of voltage has been acquired, a frequency estimate can be calculated based on the latest six cycles of phasor.

Once the algorithm has been initialized with phasor data, it can continuously generate frequency estimates using the phasor generated for each new voltage sample. A frequency calculation, however, involves many floating-point calculations that burden the processor from performing other tasks. In one test, using a 40MHz 32-bit processor,

generating a frequency calculation for every new data point for one second worth of data took 23 seconds to complete. This performance is unacceptable for the requirements of the FDR. Therefore it was decided to make only ten frequency calculations and outputs a second, easily within the computational abilities of most processors. Ten calculations a second equates to a frequency calculation for every 144 phasors, meaning a frequency calculation is made every 144/24 = 6 cycles.

3.5.2 Interrupt Handling

In the PowerPC literature, the definition of "exception" and "interrupt" are not always consistent [36]. Exceptions are events that change normal program flow and machine state, such as reset, system call instruction, and various bus access errors. Interrupts are one type of exception. They are caused by interrupt requests from input pins or devices, such as internal peripherals, and these interrupt are called "external interrupts". The interrupt sources used in the FDR include:

- Input pin $\overline{\mathrm{IRQ}}[1]$
- Peripheral modules on the intermodule bus: TPU, QSMCM, and MIOS.

The exception vector is an address where the processor begins execution after an exception is recognized and the immediate state of the machine saved. In a PowerPC, exception vectors have fixed locations, and each exception has its own exception vector, which is the sum of a base address and a vector offset. In PowerPC architecture, all interrupts share one exception vector offset, normally at 0x500. If no exception vector relocation is applied, the exception vector table is simply the exception vector offset, since the exception base address is 0x0.

An interrupt level is a number assigned by software to all interrupt sources except the input pins $\overline{\mathrm{IRQ}}[0{:}7]$, which have a fixed interrupt level and priorities. The interrupt level provides a mapping mechanism for software to identify which interrupt source is causing an interrupt request, and determine a priority if two or more interrupt requests occur at the same time. The Power PC core has only a single interrupt input, which is from the

interrupt controller. The main interrupt controller is in the USIU, and there are interrupt controller functions in other areas, such as the level mapping of peripherals in the UIMB module. The USIU interrupt controller has sixteen inputs: eight external interrupt request pins $\overline{\mathrm{IRQ}}[0:7]$ and eight internal interrupt "levels". Interrupt sources inside the USIU are assigned a level $0{:}7$. Interrupt sources from peripherals on the IMB3 bus have a level $0{:}31$, and the interrupt controller function in the UIMB reduces the 32 possible interrupt levels to eight levels by mapping interrupt levels 7:31 to level 7 of the USIU level. A common rule it to let each peripheral on the IMB bus use a different interrupt level to minimize interrupt service routine time in determining the source of the interrupt. The low number levels have priority over higher numbers if two interrupts occur at the same time, so the more important interrupt sources must reside in the lower levels. To decrease the response times of interrupt service routines (ISRs), only the high interrupt levels from 0 to 7 should be used.

The sixteen USIU interrupt controller inputs are fed to the SIPEND (USIU interrupt pending register). Software can read this register to see which of the sixteen interrupts are pending. The SIMASK (USIU mask register) contains corresponding mask bits for each SIPEND interrupt bit. Besides SIPEND and SIMASK, a priority and encoder gives a number called interrupt code to the highest priority unmasked interrupt. If two or more unmasked interrupt requests occur at the same time, the one with the lowest numbered interrupt code will have priority. The interrupt code is located in a field of the SIVEC (USIU interrupt vector register), which is a 32-bit read-only register that contains an 8-bit code representing the unmasked interrupt source of the highest priority level. Each interrupt code is separated by four bytes; the width of one instruction. During the interrupt service routine, the interrupt code can be used as index into a branch table for branching to the appropriate interrupt sources service routine.

All of the I/O functions in the FDR are interrupt-driven to minimize the CPU cost on I/O function and optimize the system efficiency. Blocking I/O operations prevents timely data measurements that are necessary for accurate frequency estimates. There are four

interrupt sources used in the FDR: external interrupt pin $\overline{\text{IRQ}}[1]$ for data sampling, MIOS interrupt for PWM pulse and period measurement, TPU for data transmission, and QSMCM for GPS time stamp collection. The interrupts generated by the subsystems of the MPC555 (TPU, MIOS, QSMCM) are considered to be external interrupts, as are the external interrupt pin $\overline{\text{IRQ}}[1]$ on the chip. This requires programmers to test each subsystem they are interested in when an external exception has occurred. Table 2 lists the priority and codes of interrupts used in the FDR.

Table 2 FDR interrupt priority and codes

Priority	Interrupt Source	Interrupt Level	Interrupt Code (Hex)	Exception Vector Table
2	Input Pin	$\overline{\text{IRQ}}[1]$	0x8	0x500
5	MIOS	Level 2	0x14	0x500
7	TPU	Level 3	0x1C	0x500
9	QSMCM	Level 4	0x24	0x500

In order to use interrupts, each subsystem must be configured by setting the appropriate values in its status/control registers. An interrupt level must also be assigned to that subsystem. By assigning a unique interrupt level for each subsystem, the developer can more easily determine the subsystem causing the exception by using the SIVEC register located in the USIU subsystem. Enabling or disabling a particular level of interrupts in the SIMASK register enables the developer to ignore interrupts during critical sections of the frequency algorithm, especially during calculation intensive resampling.

3.5.3 External A/D Conversion

The voltage signal is sampled by using the Analog Device AD976A 16-bit parallel 200ksps (kilo-samples per second). The interface between the external A/D converter and MPC555 simplifies the software configuration quite a bit. As shown in Figure 3.7, the $\overline{\text{BUSY}}$ signal of the A/D converter AD976A is inverted and connected to pin $\overline{\text{IRQ}}[1]$ of

the MPC555. The data bus pins of A/D converter are connected to the GPIO (General Purpose Input/Output) pins of the MPC555. Once an A/D conversion is completed and the digital conversion results are ready on the data bus, the $\overline{\text{BUSY}}$ signal of the A/D converter goes up to inform the MPC555 to read the data from GPIO. After an inverter, the $\overline{\text{BUSY}}$ signal generates a falling edge at $\overline{\text{IRQ}}[1]$, which asserts interrupt request and start the external A/D conversion interrupt subroutine.

These interrupts are activated when voltage A/D conversions are desired, and ignored after 1440 measurements have been taken within a single second. As shown in Figure 3.7, the MPWM0 signal of MPC555 is connected to the R/$\overline{\text{C}}$ input of the ADC, since the PWM (Pulse Width Modulation) submodule of the MIOS is used to generate the 1440Hz triggering pulse for the A/D conversion. In order to generate the appropriate 1440Hz for synchronized voltage sampling, the clock prescaler of the MIOS1's counters needs to be configured and enabled, which provides a stable and reliable internal counter for PWM pulse generation. The period and duty cycle of the PWM pulse can be specified by configuring the PWM Submodule Period Register and the PWM Submodule Pulse Register respectively. The R/C' signal into the AD976A converts on the low signal edge. On the high signal edge, it begins to drive the bus with the output data, which is guaranteed valid after 5us. In order to ensure that we do not try to latch data when the ADC is not driving the signals, a fairly short active low pulse is needed in this case.

In order to assure that our PWM generated pulses are accurate, we use the MIOS Double Action Submodule (MDASM) in concert with the PPS from the GPS unit to measure the period of the pulse. This allows us to get an accurate measurement of one second in "counts" from the MIOS subsystem to remove the inaccuracy or drift in the MPC555's oscillator. With this period measurement, a new PWM period and width can be adjusted in order to better meet the coherent sampling requirements. When the GPS receiver assets a new 1PPS, the PWM pulse train is restarted with the new PWM period and width being effective.

3.5.4 MIOS

The MIOS (Modular I/O system) consists of a library of flexible I/O and timer functions, such as the I/O port, counters, PWM, pulse and period measurement, input capture, and output compare. MIOS1 is the implementation of the MIOS architecture used in the MPC555. MIOS1 is easily configurable for different kinds of applications. The MIOS1 is composed of the following submodules:

- One MIOS bus interface submodule (MBISM),
- One MIOS counter prescaler submodule (MCPSM),
- Two MIOS modulus counter submodules (MMCSM),
- 10 MIOS double action submodules (MDASM),
- Eight MIOS pulse width modulation submodules (MPWMSM),
- One MIOS 16-bit parallel port I/O submodule (MPIOSM), and
- Two MIOS interrupt request submodules (MIRSM).

In FDR, the MIOS serves four specific functions:

- Parallel I/O port,
- PWM pulse triggering A/D conversion,
- Internal timer, and
- Period measurement.

The 16-bit parallel port I/O submodule MPIOSM is used as the parallel I/O port working with the external A/D converter. Once the A/D conversion is completed, the digital data is latched on the data bus, and the MPC555 can read the conversion results through the parallel port.

The MIOS PWM submodule generates a pulse train for the triggering of the external A/D converter; each trigger requests an interrupt. The A/D converter makes 1440 conversions each second, and then disables the interrupt request. The period and width of the PWM pulse train for triggering can be set by configuring the registers of the MIOS Module Counter Submodule, which also allows the user to configure the edge sensitivity and frequency of the MIOS built-in counter system. The PWM will begin new pulse generation with the new PWM period and width setting at the beginning of each second.

The MIOS Module Counter Submodule allows the user to configure the edge sensitivity and frequency of the MIOS built-in counter system. The rollover of an up-counter MMCSM6 can generate an interrupt, and the number of the rollover can be recorded. By comparing the counting cycle number within a single second, the time difference between two consecutive PPS can be determined, and the validation of the PPS from the GPS is judged. This validation is checked in each PPS interrupt to see if there is any abnormality with the PPS signal from the GPS. If the difference between the period measurement result and one second is within a satisfied range, this means the PPS from the GPS is good enough for synchronization application. If the period measurement result is far away from our expected time, then no frequency calculation will be processed and the program will switch to acquisition mode.

The MIOS double action submodule (MDASM) provides two consecutive 16-bit inputs captures or two consecutive 16-bit output compare functions that can occur automatically without software intervention. The input period measurement mode of the MDASM15 is used for period measurement in the FDR. The input edge detector is programmable to trigger the capture function to occur on the rising edge. Thus MDASM15 can check and capture a rising edge of the 1PPS signal from the GPS, and generate an interrupt when a PPS is coming.

3.5.5 QSMCM

The queued serial multi-channel module (QSMCM) provides three serial communication interfaces: the queued serial peripheral interface (QSPI) and two serial communications interfaces (SCI1 and SCI2). The dual, independent SCIs are used to communicate with external peripherals and devices via an asynchronous serial bus. Each SCI is a full-duplex universal asynchronous receiver transmitter (UART) serial interface. In the FDR, the SCI1 is used to receive the GPS timestamp data, and the SCI2 is used to send frequency measurement data to the serial device server. SCI1 and SCI2 have different baud rate and queue management parameters in order to meet different serial communication requirements.

In the FDR, the SCI1 operates in interrupt-driven mode. Once the RIE (Receive Interrupt Enable) bit in the control register of SCI1 is set, an interrupt results whenever the RDRF (Receive Data Register Full) is set. The Receive Data Register (RDRx) is a read-only register that contains data received by the SCI serial interface. The RDRF is set when the contents of the receiving serial are transferred to register RDRx, and the RDRF is cleared before the next transfer from the shifter takes place. So when s time stamp message from the GPS arrives and is transferred to the receive data register, the RDRF is set and an interrupt request is asserted, and the MPC555 can read the message from the RDRx.

In SCI2, because the algorithm communicates binary data to the server, no special character sequences could be reserved for flow control. Therefore, the TPU (Time process unit) of the MPC555 is used to support RTS/CTS hardware handshaking of the serial communication of SCI2. The transmit data register (TDRx) is a write-only register that contains data to be transmitted. Data is first written to TDRx, and then transferred to the transmit serial shifter, where additional format bits are added before transmission. Once the byte in TDRx is transferred to the transmit serial shifter, Transmit Data Register Empty (TDRE) is set. The TCIE (Transmit complete interrupt enable) bit in the control register is set to enable the data transmit complete interrupt. With TCIE set, the TC (Transmit complete) bit is set when the transmitter finishes shifting out all data.

3.6 TPU Timing processing unit

The time processor unit 3 (TPU3), an enhanced version of the original TPU, is an intelligent, semi-autonomous microcontroller designed for timing control. Operating simultaneously with the CPU, the two TPU3 modules process micro-instructions, schedules and processes real-time hardware events; performs input and output; and accesses shared data without CPU intervention. Consequently, for each timer event, the CPU setup and service times are minimized or eliminated. The TPU3 can be viewed as a special-purpose microcomputer that performs a programmable series of two operations; match and capture. Each occurrence of either operation is called an event. A programmed

series of events is called a function. TPU functions replace software functions that would require CPU interrupt service.

The microengine is composed of a control store and an execution unit. The control-store ROM holds the microcode for each factory-masked time function. When assigned to a channel by the scheduler, the execution unit executes microcode for a function assigned to that channel by the CPU. A new feature of the TPU3 microcode ROM is the existence of two entry tables in the four Kilobytes of internal ROM. Each entry table has a set of sixteen functions that define which of the two tables the TPU3 will be able to access.

DIO, Discrete Input/Output, is one of the sixteen functions that defined in microcode. The DIO function allows a TPU channel to be used as a digital I/O pin. As an input, the channel can be read either on command or at a periodic rate. As an output, the channel can be driven high or low upon request by the CPU. The users can choose one of the three following conditions to update the parameter:

- when a transition (positive, negative, or either) occurs,
- when the CPU makes a request to read the logical value driving the pin,
- when the CPU makes a request to drive the pin to a specified logical value or
- when a periodic rate specified in the MATCH_RATE register.

The TPU3 has sixteen independent channels, each connected to an MCU pin. All channels have identical hardware and are functionally equivalent in operation. Each channel consists of an event register and pin control logic. The event register contains a 16-bit capture register, a 16-bit compare/match register, and a 16-bit greater-than-or-equal-to comparator. The direction of each pin, either output or input, is determined by the TPU microengine. Each channel can either use the same time base for match and capture, or can use one time base for match and the other for capture.

In order to properly transmit frequency measurement data on SCI2, RTS/CTS hardware handshaking is required. Request to Send (RTS) is the outgoing flow control signal, and Clear to Send (CTS) is the incoming flow control signal. By using the Channel Function

Select Register, the function assignment and function specific parameters of Channel 14 and Channel 15 of TPU3B can be specified. Channel 15 is used as an output for RTS, and it indicates that Net _buffer is not empty and ready to send out a byte. Channel 14 is an input for CTS, and it is valid when the serial device server completes the previous transition and requests the next byte transition. Each of the TPU3 channels can generate an interrupt service request, so a CTS interrupt service routine is designed for the frequency data transmitter on SCI2.

3.6.1 GPS timing synchronization

GPS is designed primarily for navigational purposes, but it also provides a common-access timing pulse, which is accurate to within one microsecond at any location on earth. The GPS receiver provides the one Pulse Per Second (PPS) signal and a time tag, which consists of the year, day, hour, minute, second and sub-second. The time could be UTC (Universal Time Coordinated) or GPS time. The PPS is usually divided by a phase-locked oscillator in the required number of pulses per second for sampling of the analog signals. In FDR, this is 24 samples per cycle of the fundamental frequency. For accurate acquisition of the timing pulse, only one of the satellites need be visible to the antenna.

Frequency measurement is tagged with the UTC time corresponding to the time of measurement, which consists of two parts, a second-of-century (SOC) count and fraction-of-second count. If there is an interrupt on SCI1, the GPS time message will be saved. The SOC count is in seconds from midnight (00:00) of January 1, 1970 to the current second. In SOC, the leap second is added or deleted as necessary in order to keep the SOC count synchronized with UTC. The second is divided into ten by the integer number of subdivision 0.1 in FDR. In other words, the fraction-of-second number is any number from the set $\{0,1,2,3,4,5,6,7,8,9\}$; and 0 when the measurement coincides with the 1PPS signal, so on and so forth. The fraction-of-second can be changed by changing the subdivision of the second.

The design requirements state that the synchronizing signal shall have a 1PPS with a stability of at least 1E-07. The synchronizing signal must be available without

interruption at all measurement locations throughout the interconnected grid. Reliability should exceed 99.87% (1h of outage per month). The signal should be accurate enough allow the phasor measurement equipment to maintain synchronism within 1us of UTC time including both synchronizing source and local receiving equipment error.

The Global Positioning System (GPS) is a satellite-based radionavigation system, which is developed and operated by United States Department of Defense. Completed in 1994, it consists of 24 satellite arrays to provide a minimum worldwide visibility of four satellites at all times. The time of the GPS system is adjusted with respect to a ground-based Cesium clock ensemble, which is referenced to the world standard Coordinated Universal Time (UTC). Each satellite provides a correction to the UTC time the GPS receiver automatically applied to the outputs. With this continuous adjustment, timing accuracy is limited only by short-term signal reception, whose basic accuracy is 0.2us. For general applications, the 0.5us accuracy is a safer figure to depend on. The inherent availability, redundancy, reliability, and accuracy make GPS a system well suited for the synchronized phasor measurement.

The Motorola M12+ evaluation board uses one very small rechargeable Manganese-Lithium cell as backup battery. Assuming the typical keep-alive current is 5uA, the backup battery can support the M12+ for a little more than two weeks. Without back-up power, the receiver's RAM will lose all the satellite and configuration information. This means that the GPS receiver will restart from default mode once it receives power again. To make sure the normal message traffic between the M12+ and CME-555 board is restored, we need to include an initialization sequence for the GPS.
Fortunately, the M12+ GPS receiver (time version) offers binary commands, which can be used to initialize, configure, control, and monitor the receiver. Motorola binary commands are supported on the serial communication port, and work at a default of 9600 baud, 8 bits, no parity, and one stop bit. The serial port configuration is never changed to simplify operation.

For GPS initialization, we reset the unit. This guarantees the GPS receiver will be in the default mode after reset. To reset the GPS, we just need to input the Cf message in hex format:

0x40 0x40 0x43 0x66 0x25 0x0d 0x0a.

Next we must configure the 1PPS signal, since it is the key timing for other parts of the software. There are three modes for the 1PPS output: 1PPS on continuously, 1PPS active only when tracking at least one satellite, and 1PPS active only when the T-RAIM algorithm confirms the time solution error is within user-defined limits. During the development period, we use the continuous output mode, since this makes testing and debugging for the whole system much easier. To set the 1PPS to output continuously we can input the Gc command in hex format:

0x40 0x40 0x47 0x63 0x01 0x25 0x0d 0x0a.

For embedded software, which needs to assure the timing accuracy, we set 1PPS active only when the T-RAIM algorithm confirms the time solution error is within a limit, such as 1400ns. To set this T-RAIM limitation, the input Gc command in hex format is:

0x40 0x40 0x47 0x63 0x03 0x27 0x0d 0x0a.

Note that before setting 1PPS to be active only when T-RAIM algorithm confirms a time solution error is within a specified limit, we need to enable T-RAIM and set the desired T-RAIM Alarm limit first. The input Ge command to enable T-RAIM in hex format is:

0x40 0x40 0x47 0x65 0x00 0x22 0x0d 0x0a

and the input Gf command to set the T-RAIM alarm limit to 1400ns is:

0x40 0x40 0x47 0x66 0x00 0x0e 0x2f 0x0d 0x0a.

For our application the timing solution is the only concern, so we set the satellite mask angle at 3°. This allows the GPS receiver attempt to track the satellites for which the elevation angle is greater than 3°. The input Ag command to change the mask angle to 3° (in hex format) is:

0x40 0x40 0x41 0x67 0x03 0x25 0x0d 0x0a.

At the end of GPS initialization, we set the receiver to output the position message once per second. The Eq command to set the GPS receiver output Eq message continuously (in hex format) is:

0x40 0x40 0x45 0x71 0x01 0x35 0x0d 0x0a.

Each second, the CPU can get and parse the response message to the above command. The Eq response message has the following format:

@@Eq, mm, dd, yy, hh, mm, ss…

UTC Date:	mm	month	01…12
	dd	day	01…31
	yy	year	99…19
UTC Time:	hh	hours	00…23
	mm	minutes	00…59
	ss	seconds	00…60

Actually, the Eq message includes a much longer list of information than that shown above. In our application we only need the information mentioned here. So far, we have completed the initialization of M12+ GPS, and configured the desired traffic between the GPS receiver and the CME-555 board.

Chapter 4 Practical Implementation Issues

In practical power systems, the ideal sinusoidal wave rarely exists. Meanwhile, the A/D converter, microcomputer and other components in the real FDR implementation will also introduce some estimation errors. In order to obtain high-precision, time-synchronized, real-time, networked data, there have been many challenges and practical issues that have emerged during the system design and implementation. In this section, some description and analysis is dedicated to such challenges and practical issues, and solutions are proposed with supportive simulation results.

4.1 Coherent Sampling Issue

Voltage sampling is the first step in frequency measurement, which is the base for FNET power system analysis applications. The voltage samples should represent the original analog signal faithfully, and the sampling should occur at a precise instant. In addition to these requirements, because of the characteristics of the phase angle analysis technique and the 60Hz power system, some practical and special efforts should be given to implement the synchronized voltage sampling.

4.1.1 Coherent Sampling

In general, the DFT can be used to convert time domain signals to the frequency domain. Mathematically, DFT is a series summation of the product of each time domain sample multiplied by a complex number. All the components in the time domain signal for DFT are assumed to be periodic, and the DFT treats sample data set as though it were one period of a periodic sequence. In other words the time domain samples for the DFT should contain a precise integer number of cycles. If the data for a waveform is not periodic, then distortion may occur because the periodic waveform created by the DFT may have sharp discontinuities at the boundaries of the sample data set. If a longer sample set is created by appending a second set to the first set, these two do not form a continuous waveform. A discontinuity occurs where the two sample sets join, as shown

in Figure 4.1, causing a sharp edge within a sinusoid, which in time has high-frequency components.

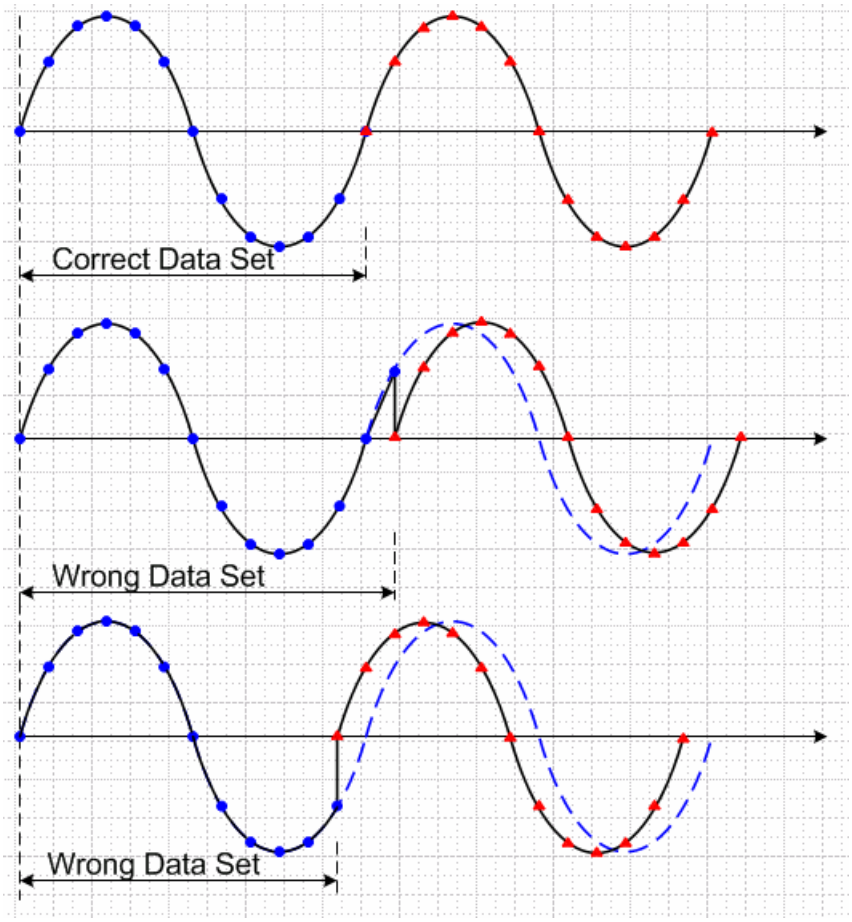


Figure 4.1 Sample Set with Leakage

With the sampling frequency F_s, an N point DFT only contains the information about frequencies that are integer multiples of the fundamental frequency F_s/N, and those frequency components that are not integer multiples of F_s/N leak into the frequency points which are returned by the DFT. The artificial discontinuities in Figure 4.1 turn up as very high frequencies in the spectrum of the signal; frequencies that were not present in the original signal. These frequencies could be much higher than the Nyquist frequency, and will be aliased somewhere between 0 and fs/2. The DFT coefficients obtained by using a DFT will therefore not be the actual coefficients of the original signal, but will be smeared versions. As discussed in Chapter 2 and Chapter 3, the phasor angle

analysis technique is based on DFT, so those discontinuities jeopardize the accuracy of phasor angle, and consequently the frequency measurement.

In the general case of sampling, these discontinuities are unavoidable, and there are two widely used solutions: time window functions and coherent sampling. Time window functions can minimize the effects of leakage. However, unless special care is taken, windowing reduces spectral leakage to only about -80dB, and for high-resolution applications (>12 or 13 bits) windowing may not help [37]. Coherent sampling basically synchronizes the sampling frequency with the inherent frequency of the sampled signal, and by doing this the discontinuities can be minimized or eliminated.

In order to implement coherent sampling, there are two aspects that must be considered. First, the data window should cover an integer number N of signal cycles where each cycle has a period T_0; this means the length of the data window is NT_0. Secondly, the length of the data window should be an integer M multiple of the sampling period T_s. So coherency means that a sample set contains an integer number of samples of an integer number of signal cycles, and the following condition should be satisfied in order to ensure coherent sampling:

$$\frac{F_0}{F_s} = \frac{N}{M} \tag{4.1.1}$$

Where F_s is the sample frequency and $F_s = 1/T_s$,

F_0 is the signal frequency and $F_0 = 1/T_0$,

N is the number of cycles in the data window, and must be an integer, and

M is the number of samples in the data window, and must be an integer.

Coherent sampling guarantees that the number of sets of samples of signal cycles will be an exact integer, and the maximum amount of information about a particular waveform exists in the data window. If such a condition is not satisfied, spectral leakage occurs when using DFT, and the accuracy of measurement may suffer from the truncation error. The only complication is that timing synchronization of the signal and sampling

frequencies is required, increasing development efforts for the sampling scheme. However, in practical situations it is often difficult for the sampling procedure to be exactly synchronized with the input signal. This is primarily because both the sampling and signal frequency may vary with time because of many factors, such as oscillator instability and the signal frequency varying.

4.1.2 PWM Precision

In the FDR, the triggering pulse for coherent sampling is generated by the Pulse Width Modulation (PWM) output of the MPC555, and the frequency of PWM pulse is determined as the following:

$$F_s = \frac{F_{sys} / PSL}{PERR} \tag{4.2.1}$$

in which F_s is the desired sampling frequency,

F_{sys} is the system frequency 40MHz,

PSL is the Clock pre-scalar, which is set to 2 in this case, and

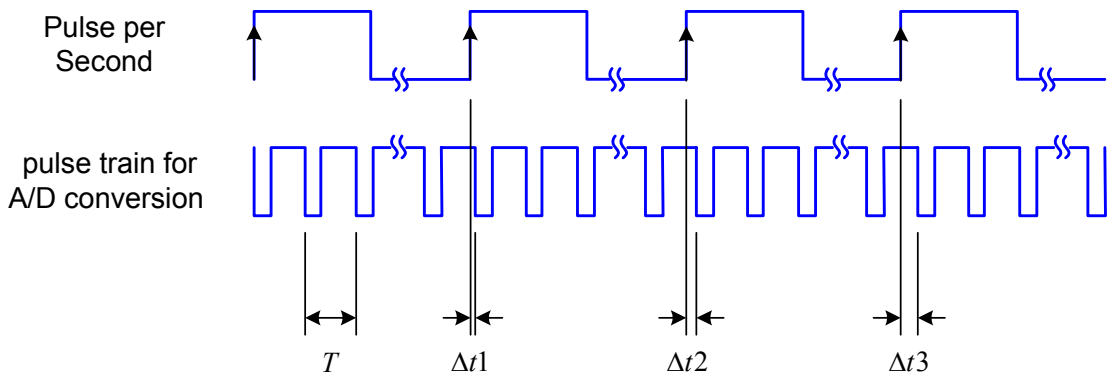
PERR is the PWM period register value, and it must be an integer.

For a 60Hz system, 1440 samples per second or 24 samples per cycle is the designed sampling rate for frequency estimation; however, according to (4.2.1) the closest sample rate could only be 1440.092 samples/second or 14399.885 samples/second, due to the integer requirement of PERR. In other words, PERR can only take 13888 or 13889 samples per second in this case. This means that the period and duty cycle of the A/D conversion triggering pulses are controlled by computer programs, and they are close to but not exactly the ideal value. In other words the trigger signals for sampling can only get within a satisfactory range.

When digitized information is examined with a Fourier transform, the DFT/FFT algorithm assumes that the data was sampled at precise intervals with no time jitter. Any time deviation in the digitizer clock from its expected value causes an error. Jitter in the clock will distort the data the same way it would have if the signal had jittered. When a signal is digitized, only its amplitude information is preserved. The data in the time sample array appears to the math algorithm as amplitude error, distorting the time

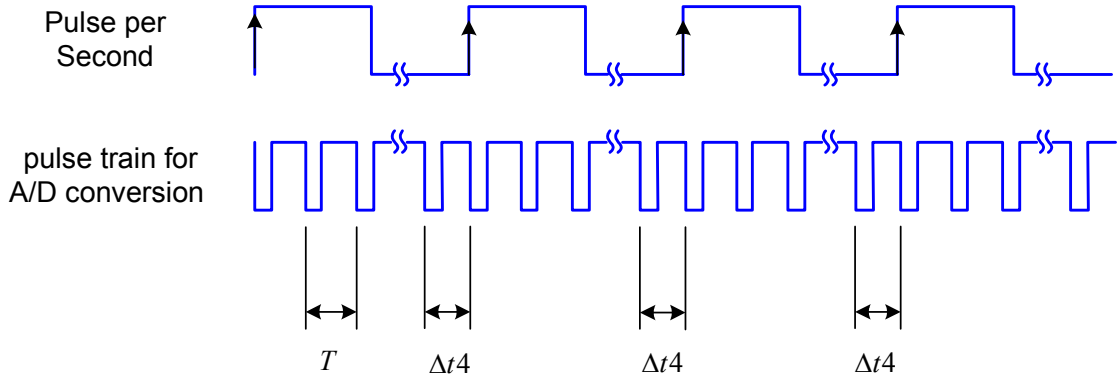
waveform. Time jitter in a sampler distorts the frequency analysis results, which may appear as noise. If the sample timing were perfect and the input sine wave had no amplitude distortion but jitters in time, the result would be the same. In summary, jitter in either the sample timing or the analog signal timing can result in frequency distortion. If there are sample and signal jitters that are statistically independent, the problem is compounded even further. If there are sample and signal jitters that always move exactly the same, the problem is hidden; this is what occurs when both the analog signal and the sample timing are created via a master clock.

There are two straightforward methods for generating the triggering pulse train for A/D conversion. In the first method, shown in Figure 4.2, the pulse train is fixed right at the beginning of measurement, and it will not be changed until the program ends. In the second method, as shown in Figure 4.3, the pulse train is reset at the instant of 1PPS being detected. In both Figure 4.2 and Figure 4.3, the PERR is set to be 13889 according to (4.2.1). In Method One, the time shift between the PPS edge and the first PWM pulse of each second is an uncertain value for each second. Hence it is almost impossible for two units that start at different time to be synchronized to a specified time reference, which is UTC time in this case. The basic meaning of wide-area frequency measurement is to obtain the frequency measurement results at different locations at the same timing base. As mentioned in Chapter 3, the synchronization is accomplished by forcing all frequency data to the synchronized 1PPS, and the required level of precision for the synchronization sampling method is 1us. At this point Method One cannot be the proper choice for the FDR.



T pulse period = 694.45*us*
$\Delta t1$ time shift at the 2nd second = 8*us*
$\Delta t2$ time shift at the 3rd second = 16*us*
$\Delta t3$ time shift at the 4th second = 24*us*

Figure 4.2 Method One: continuous pulses from PWM



T pulse period = 694.45*us*
$\Delta t4$ the period of last pulse of each second = 694.45 - 8 = 686.45*us*

Figure 4.3 Method Two: reset PWM each second

In method two, the parameters of the PWM pulse train restarts at the beginning of each second coincident with the 1PPS signal from the GPS. The first 1439 voltage sampling data of each second occur at an expected sampling rate, i.e. the A/D conversion occurs at every $13889\times(1/20\text{M})=694.45\,\mu\text{s}$. However the time deviation between the last voltage sample of one second and the first voltage sample of the next second is $1\text{-}1439\times694.45=686.45\,\mu\text{s}$, as shown in Figure 4.3. This uneven sampling interval introduces phase shift to the data window, as shown in Figure 4.4, and consequently the phase shift causes errors in the frequency estimation results as shown in Figure 4.5.

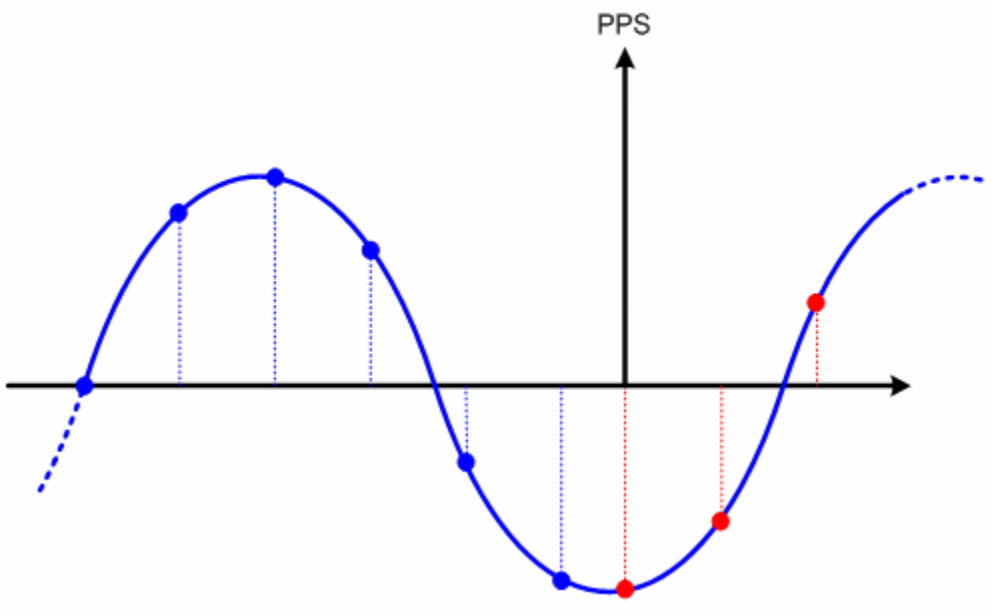


Figure 4.4 Phase shift phenomena

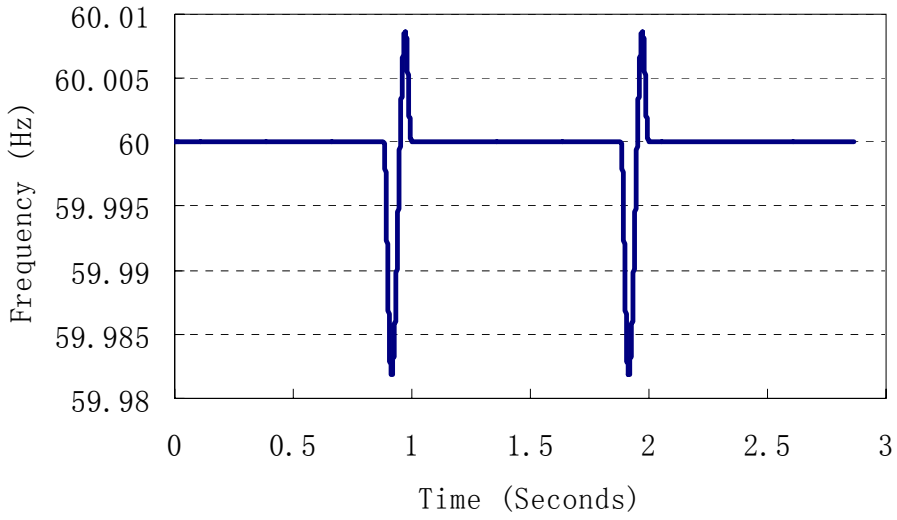


Figure 4.5 Phase shift introduce errors in frequency measurement

4.1.3 External oscillator

As discussed in Section 4.1.1, the undividable internal oscillator of the PWM submodule causes phase shift. In order to eliminate or reduce the phase shift effect, it is intuitive to think about using an external oscillator, which can be dividable by 1440Hz or other future $60 \times N$ sampling rate for a 60Hz system. There are a variety of choices of 1440-

dividable crystal or clock oscillators; for example the 29.4912MHZ clock oscillator ASV-29.4912MHZ-EJ-T from ABRACON, and the 14.7456MHz crystal XT49M-2014.7456M from VISHAY DALE.

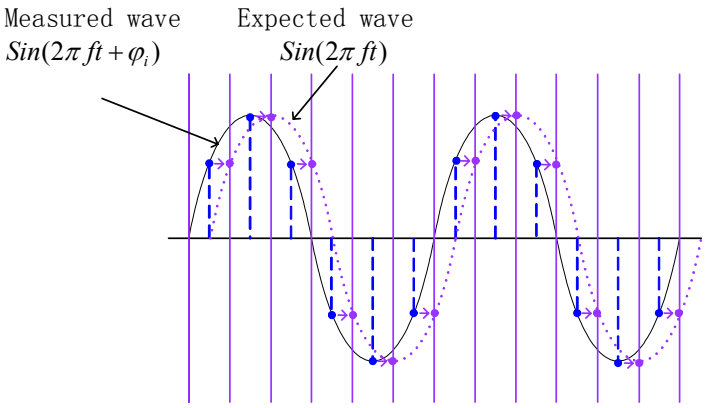


Figure 4.6 Step phase shift

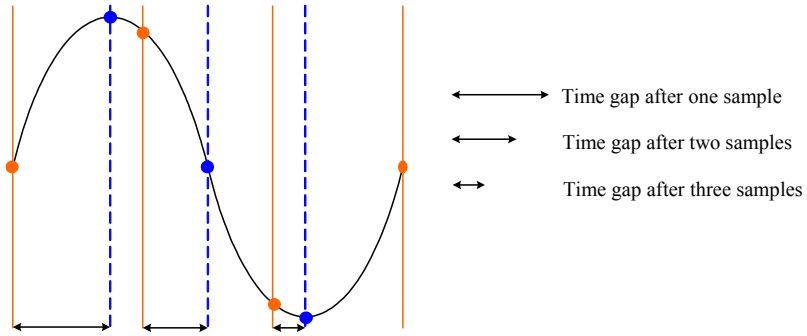


Figure 4.7 Accumulated phase shift

The benefits of using an external oscillator are obvious, since it solves the uneven sampling issue directly. However, external oscillator implementation increases the system complexity and introduces additional possible errors to frequency measurement. Besides the concerns of this particular system, the accuracy of the external oscillator suffers greatly from temperature and aging, which can cause long-term synchronization failure between units in different locations. These errors degrade system performance. Noting that a time error of 1μs corresponds to a phase error of 0.022 degrees for a 60Hz system and 0.018 degrees for a 50Hz system, the accuracy of frequency estimation is very sensitive to variations of the system's oscillator or clock.

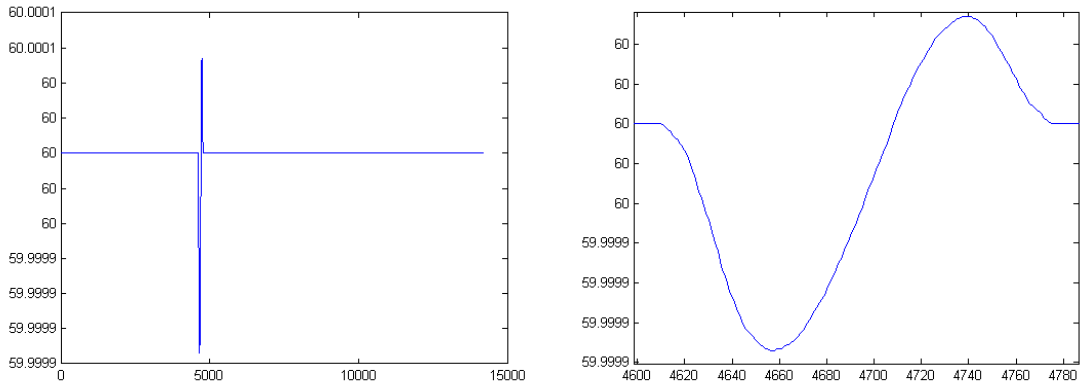


Figure 4.8 Estimation with Phase Shift of 0.001-degree lead

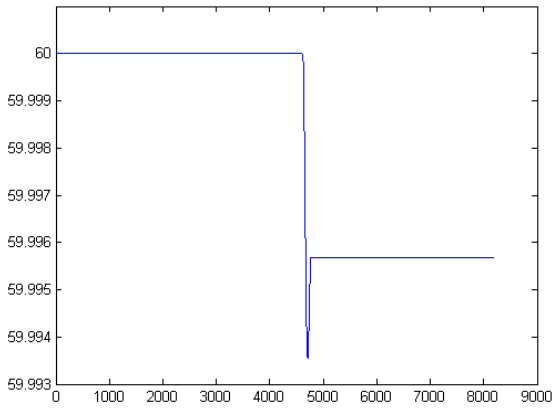


Figure 4.9 Estimation with Accumulated Phase Shift of 0.001-degree lead per sample

As shown in Figure 4.6 and Figure 4.7, the instability of oscillator can easily introduce step phase shift or accumulation phase shift to the frequency measurement. These phase shifts introduce errors to frequency estimation. Considering one clock of a 20MHz oscillator or 0.001-degree equivalent leading phase shift, a spike with peak of 0.0001Hz is introduced by the single step phase shift, and an offset of 0.004Hz from the nominal frequency is caused by the accumulated phase shifts in frequency estimation, as shown in Figure 4.8 and Figure 4.9. Thus a very small change or difference between two external oscillators will cause a big error in the frequency measurement results of different units. At this point, the external oscillator makes two FDRs use different sampling rate at different time, which make the synchronization data measurement for all FDRs meaningless. So at this point it is wise to turn to solutions other than external oscillators.

4.1.4 Alternating over-sampling

According to Figure 4.3, the value of $\Delta t4$ is known for each second, so boosting the sampling rate to a high number, say $1/\Delta t4$, makes more voltage samples available for data manipulation. By doing so, one can avoid the $\Delta t4$ time gap, and can pick up the right data points for frequency computation, as illustrated in Figure 4.10.

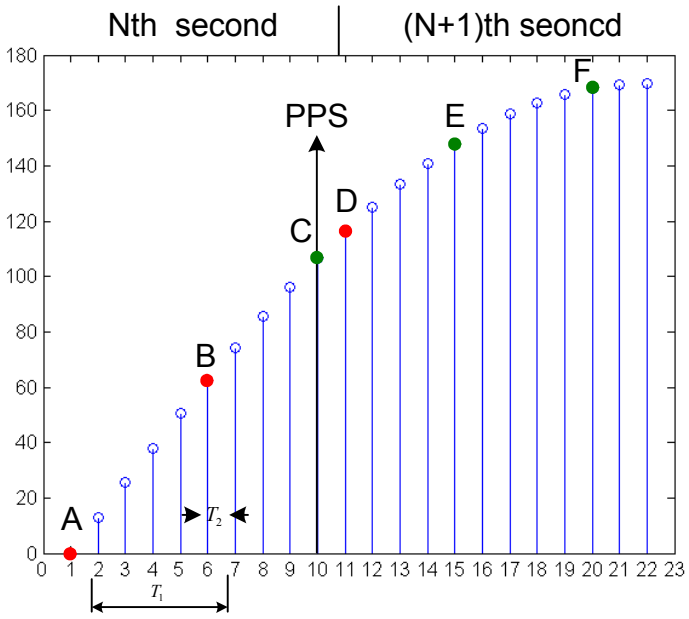


Figure 4.10 Improve over-sampling mechanism

In order not to lose any generality, we should assume the expected sampling rate is f_1 and the sampling time interval is T_1 the over-sampling rate is f_2, and the sampling time interval is T_2. We should also assume that $T_2 = \Delta t$, and Δt is defined as the sampling interval shift between the last sample and the other regular samples of each second, similar to Figure 4.3, and that $f_2 = f_1$. This time shift between sampling intervals will cause the phase shift discussed in Section 4.1.3. Without over-sampling, voltage samples A, B, and C are used for frequency estimation of the Nth second. However, the sampling interval between A and B is different from the sampling interval between B and C, so the error due to the T_2 equivalent phase shift will be introduced into the calculation. By doing over-sampling, one can use A, B, and D points for the frequency calculation of the Nth second, and use points C, E, F for the calculation of successive (N+1)th second. Now

the sampling intervals between samples are all the same, and the phase shift effects can be eliminated.

In the particular case of the FDR, the $f_1 = 1440Hz$, and $T_2 = \Delta t4 = 8.0006us$, so to avoid the phase shift due to uneven sampling, the over-sampling rate ought to be higher than or equal to $1/8.0006us = 125kHz$. One can notice that this is a fairly high sampling rate, and therefore demands more computational and data memory capability. In the current FDR design, the highest throughput of the A/D converter, AD976A, is 200K, so it can handle the higher over-sampling data. However, the computation and memory capabilities of the microcontroller are fairly limited. To implement the alternating over-sampling method, code optimization or another microcontroller or DSP are highly desirable.

4.1.5 Proportional reconstruction

From the above discussion, one can note that the phase shift is mainly caused by uneven sampling. If there is a way that can reconstruct the real signal and resample it with a unique sample interval, then the phase shift effects can be eliminated. For each second, the time shift value is already known, so the following proportional algorithm can be applied for data reconstruction.

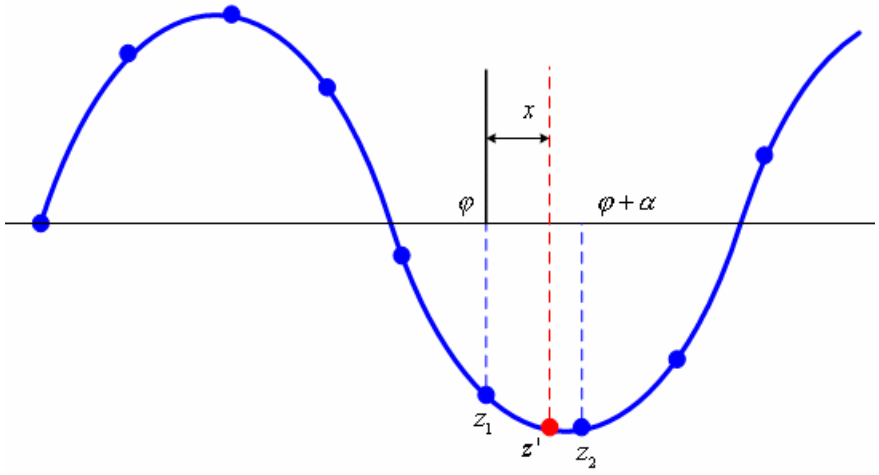


Figure 4.11 Signal reconstruction

Referring to Figure 4.11 for the resampling algorithm, the values of z_1 and z_2 can be represented by the following two equations.

$$z_1 = Z_m \sin(\varphi) \tag{4.2.2}$$

$$z_2 = Z_m \sin(\varphi + \alpha) = Z_m \sin\varphi\cos\alpha + Z_m \cos\varphi \sin\alpha \tag{4.2.3}$$

where Z_m is the amplitude of the waveform,

φ is a sample instant and is an arbitrary known value, and

α is the interval between two samples for specified signal.

Combining (4.2.2) and (4.2.3) results in

$$Z_m \cos\varphi = \frac{(z_2 - z_1\cos\alpha)}{\sin\alpha} \tag{4.2.4}$$

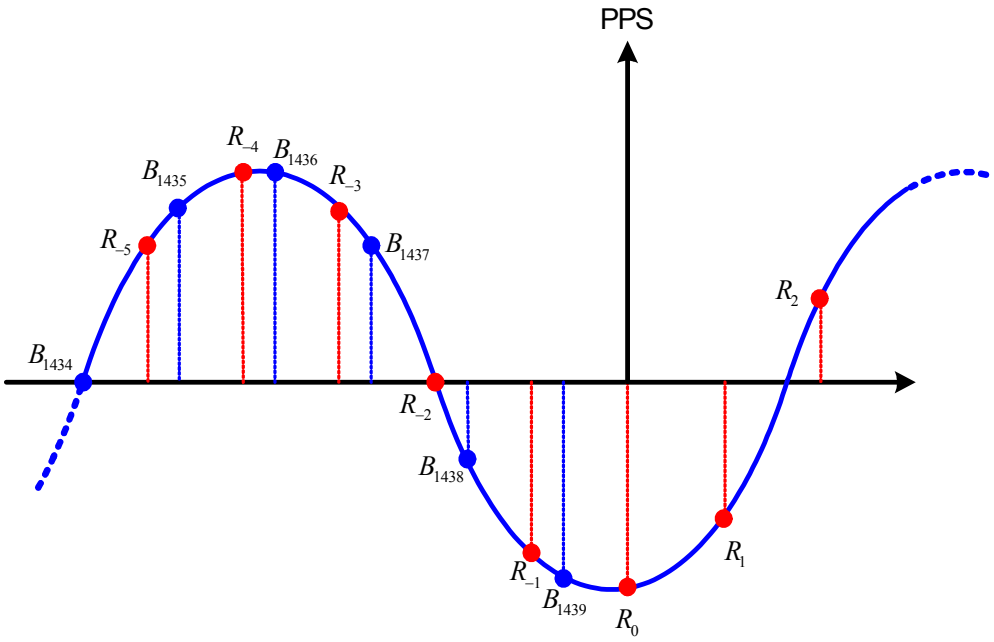


Figure 4.12 Proportional reconstruction algorithm

Letting x be the fractional distance between z_1 and z_2, the resample point z' is given by

$$\begin{aligned} z' &= Z_m \sin(\varphi + x\alpha) = Z_m \sin\varphi\cos x\alpha + Z_m \cos\varphi \sin x\alpha \\ &= z_1 \cos x\alpha + (z_2 - z_1\cos\alpha)\frac{\sin x\alpha}{\sin\alpha} \end{aligned} \tag{4.2.5}$$

We assume that the number of reconstruction data is the frequency computation window of length M, and when the PPS signal is detected, the voltage samples in the latest computation window will be reconstructed by using the proportional algorithm shown above. After reconstruction the data set should contain $M-1$ elements, and a new voltage data sampled at the right moment of PPS coming will be added to the

reconstructed data set, as shown in Figure 4.12. Therefore the new sample data with the $M-1$ reconstructed samples make up a new computation window for a new frequency computation. The computation windows before and after the reconstruction are shown in Figure 4.13.

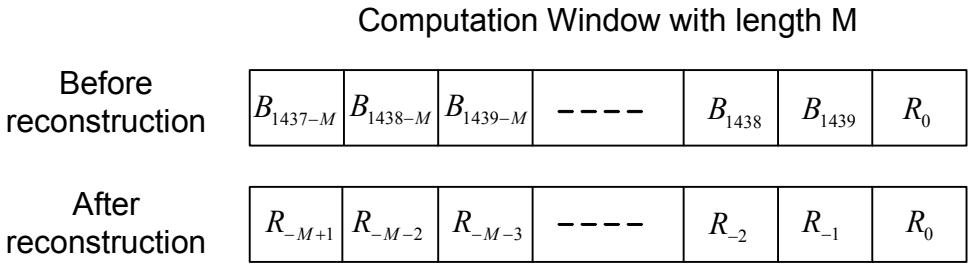
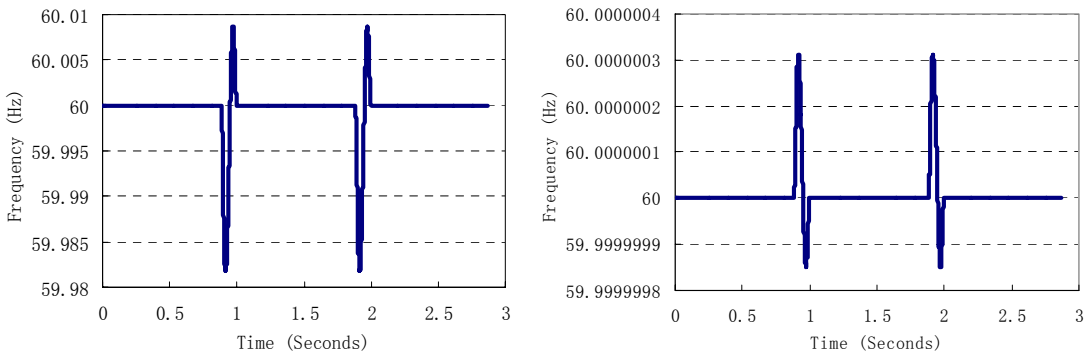


Figure 4.13 Computation window before and after signal reconstruction

To evaluate the effectiveness of the reconstruction method, the following simulation was performed. The simulation settings are the following:

- The original signal is a pure sinusoidal waveform $X(t)=169.7\sin(2\pi 60t)$.
- The sampling interval is 694.45us for the first 1439 voltage samples of each second.
- The sampling interval is 686.45us for the last voltage sample of each second, and the equivalent phase shift is 0.1728 degree.
- The simulation time is 3 seconds.



(a)No signal reconstruction (b)With signal reconstruction

Figure 4.14 Frequency measurement results for a 60 pure sinusoidal waveform

The simulation results show that the by applying the signal reconstruction algorithm, the phase shift effects can be eliminated, and the accuracy of frequency measurement results can be greatly improved. This signal reconstruction algorithm is fairly simple for software implementation, and little computation burden will be added to the microcontroller. Most importantly, the software implementation will not introduce much difference of errors from unit to unit, which means this algorithm will not impair the synchronization performance of units.

However, the basic assumption underlying the reconstruction algorithm is that input is a pure sinusoidal waveform. In a real power system, the voltage waveform is always contaminated by high-frequency harmonics and other noise. The following simulation tests the performance of the reconstruction algorithm on a harmonics-polluted voltage wave. The simulation settings are as following:

The original signal is a pure 60Hz sinusoidal waveform with 5% 5th harmonics

- $X(t) = 169.7\sin(2\pi 60t) + 169.7 \times 5\% \times \sin(2\pi \times 5 \times 60t)$.
- The sampling interval is 694.45us for the first 1439 voltage samples of each second.
- The sampling interval is 686.45us for the last voltage sample of each second, and the equivalent phase shift is 0.1728 degree.
- The simulation time is 3 seconds.

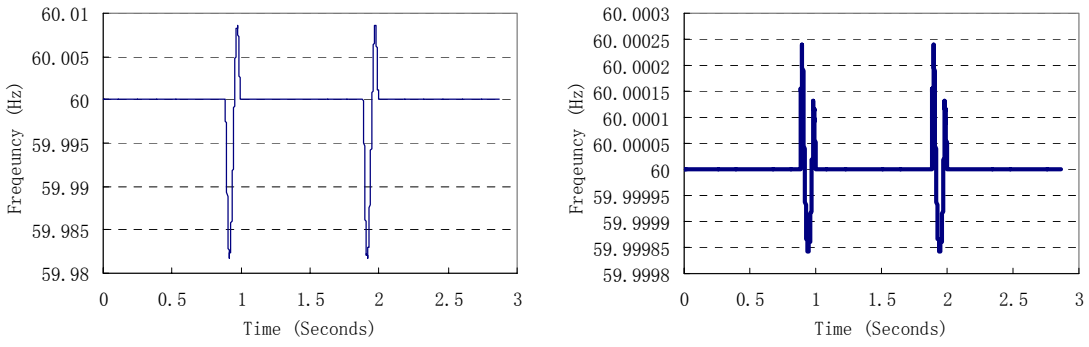


Figure 4.15 Frequency measurement results for a 60 sinusoidal waveform with 5% 5th harmonics

From the simulation results, a safe conclusion can be made that the reconstruction algorithm could not overcome the harmonics issue. Some other methods or another algorithm needs to be explored to provide more practical solutions for real systems. Earlier it was stated that only six cycles of phasors were needed to make a frequency calculation. However, that is only an initial frequency calculation. The frequency algorithm makes an initial frequency estimate using six cycles of phasors under the assumption that the signal is at a nominal 60Hz. In the real world, however, this will probably not be the case. So the proportional reconstruction algorithm addresses this problem by resampling the last eight cycles of phasors at intervals centered at the initial frequency estimate instead of at the nominal 60Hz. If the initial frequency estimate were lower than the nominal 60Hz, then resampling would require the algorithm to extrapolate the extra points beyond the original six-cycle window. Because interpolation is more accurate the extrapolation, the two additional cycle of phasors allows the algorithm calculations to be more accurate.

4.1.6 Backup sampling

Section 4.1.5 proposed the reconstruction algorithm, which reconstructs and resamples the signal at the rising edge of each PPS. In this algorithm, the accuracy of the original sampling data will affect the reconstructed data's accuracy; i.e. the reconstruction algorithm will transfer the errors in original sampling to the reconstructed data. The presence of harmonics in a real system will also cause big errors in data reconstruction processes. Backup sampling is proposed to bypass the reconstruction to minimize or eliminate the error that could be introduced by the reconstruction data processing.

In Figure 4.16, there are two A/D converters used for voltage data sampling. One is the main ADC, and the other is the backup ADC; and the two ADCs have the same sampling rate, which is 1440Hz in the case of the FDR. At the beginning of each second, which is the rising edge of the PPS signal, the main ADC will be reset and begin the new sampling sequence for the new second. After 686.45us of the PPS, the backup ADC will be reset for sampling. By doing this, the sampling data of the main ADC will be synchronized

with the PPS exactly for each second, and the backup ADC can provide the no-phase-shift voltage data for the calculation of the first frequency result of each second.

To better understand the backup sampling, one can refer to Figure 4.16, At the beginning of the Nth second, the main ADC (ADC1) will be reset and restart the sampling, and after 686.45us of the PPS signal occurring, the backup ADC (ADC2) will be reset and restart the sampling. So the backup ADC can provide the no-phase shift sampling voltages, which otherwise would need to be obtained by reconstruction, as proposed in Section 4.1.5. To reduce the cost or complications of the ADC circuit, two ADCs can be replaced by one ADC with multiple channels.

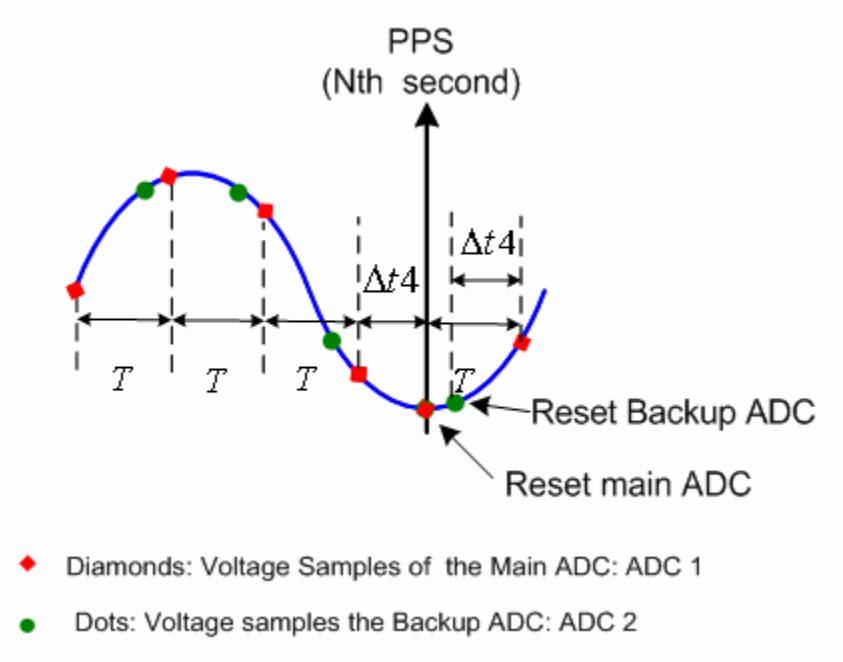


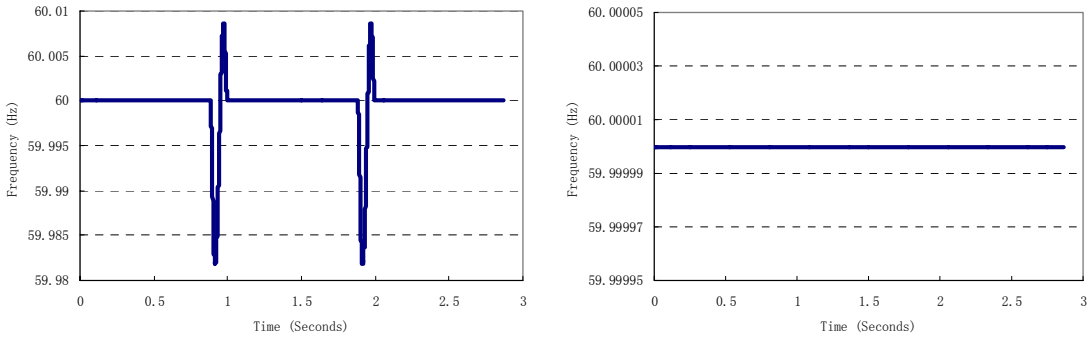
Figure 4.16 Alternating sampling

By doing this alternating sampling, one just needs to pick up the right evenly sampled voltage for frequency computation, instead of applying the reconstruction of signal proposed in Section 4.1.5.

For example, refer to Figure 4.16, when the PPS of the Nth second is coming. The ADC1 is reset and restarted, and at the beginning of Nth second, the ADC1 and ADC2 sample the same voltage value as their first sample of Nth second. Assuming the length of the

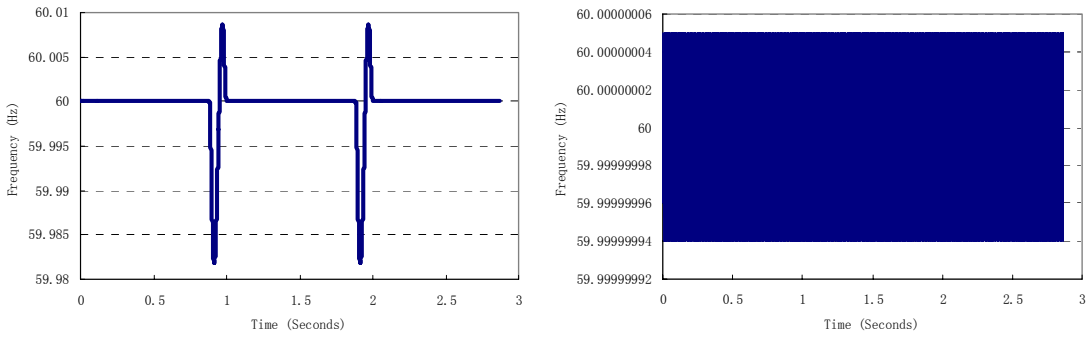
computation window of frequency estimation is M, the first sample of ADC1 during the Nth second and the last (M-1) samples of ADC2 during the (N-1)th second will be used to calculate the first frequency of Nth second. By only moving the computation window on the last (M-1) samples of ADC2 during the (N-1)th second and the new samples of ADC1 during the Nth second, there will be no phase shift in the sampling data set used for the frequency computation for the whole Nth second.

The backup A/D converter ADC2 needs to be reset and restarted after 686.45us of the PPS signal of the Nth second. Additionally the last (M-1) sampling data of ADC2 during the Nth second will be used to calculate the first (M-1) frequency estimation results of (N+1)th second. Similarly, when the PPS of the (N+1)th second is coming, the ADC1 will be reset and restarted, and so on.



(a) No Backup sampling (b) With Backup sampling

Figure 4.17 Frequency measurement results for a 60 pure sinusoidal waveform



(a) No Backup sampling (b) With Backup sampling

Figure 4.18 Frequency measurement results for a 60 sinusoidal waveform with 5% 5th harmonics

For this backup sampling, only one extra ADC or a multiple-channel ADC is required for the implementation. This will add some more computation and memory space to the software. However, the frequency measurement that results by doing the backup sampling is very attractive, since the frequency estimation error could be minimized with 1E-7 Hz, as seen in Figure 4.17

Compared to the reconstruction method proposed in Section 4.1.5, backup sampling suffers much less from harmonics, as shown in the simulation results shown in Figure 4.18 . For a real system, this alternating sampling is a very robust method. It will not introduce any errors to frequency estimation other than the ADC sampling errors. This procedure requires only a quite modest additional computational burden. It is an ideal solution for the FDR to minimize the impacts caused by uneven sampling and phase shift.

4.2 A/D Converter Quantization errors

If the minimum input voltage to the A/D converter is represented by pF_{ADC} (p=0.01~0.1 for normal application), and the required meter accuracy is ε, then (Q) number of bits of the A/D converter can be found from this equation:

$$Q \geq \log_2 \frac{0.5}{p\varepsilon}. \tag{4.1.1}$$

$\pm F_{ADC}$ is the full-scale voltage of A/D converter, and $2F_{ADC}$ is the maximum output voltage swing of voltage follower. The resolution of the A/D converter can be determined once the number of bits of ADC is decided.

Since the technique of the phasor angle analysis is taken as the main algorithm for the frequency estimation in FDR, the accuracy of frequency estimation is mainly determined by the accuracy of the phasor measurement. The phasor accuracy is limited by the data sampling as follows: for a minimum error requirement $E_{\min}$ and a full-scale rating F_s, the resolution of the A/D converter needs the following:

$$Minimum\ resolution = \ln\left[\left(2\sqrt{2}F_s\right)\Big/E_{\min}\right]\Big/\ln(2). \tag{4.1.2}$$

The factor $2\sqrt{2}$ scales the formula from RMS value to bipolar peak value. If the full-scale rate F_s is 7.07 volts, which corresponds to 10 volts peak value, and if a 1% resolution is required, the minimum bits needed for the ADC should be eleven. However practical choices for number of bits are dictated by component availability and are typically eight, twelve, sixteen, or eighteen bits. So in this case, an ADC of twelve bits or more could be selected to meets the accuracy requirements.

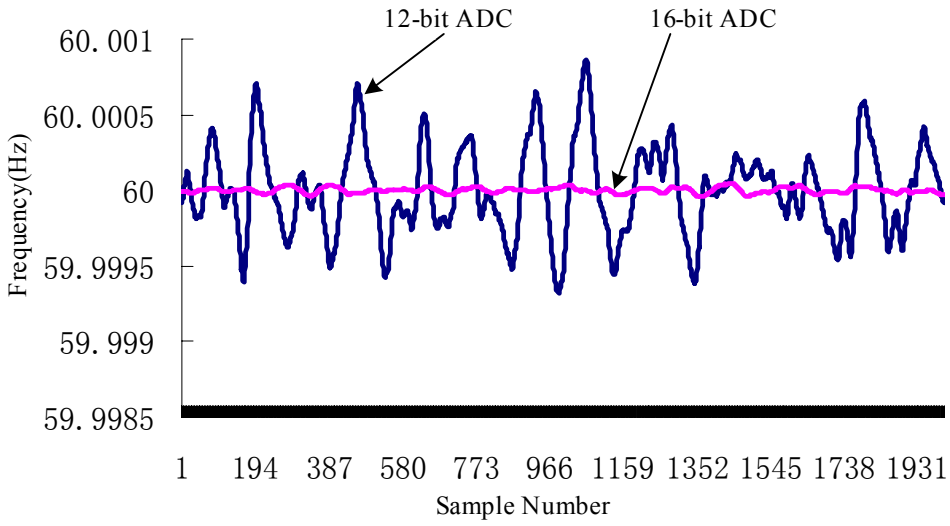


Figure 4.19Frequency estimation results of a 12-bit ADC and a 16-bit ADC

Figure 4.19 shows the frequency estimation results by using ADCs with different numbers of bits: one is twelve bits, and the other is sixteen bits. The simulation results demonstrate that by using the twelve bits ADC; the frequency accuracy can remain within $\pm 0.001\text{Hz}$, so theoretically this ADC can be a good choice for the FDR design. However, for a voltage signal in the real world, where a lot of noise and distortion are present, some reasonable design margins are highly desirable. At this point the 16-bit can be the best solution.

As mentioned early in this paper, the accuracy of the input voltage digitization is an important factor for high-precision frequency estimation. The analog-to-digital converter is the main component for the input voltage digitization process. In general, the ADC's accuracy depends on two kinds of errors in the conversion: quantization error and non-linearity. The quantization errors are defined as the difference between the analog signal and the closest sample value to represent it. These errors are measured in a unit of LSB (least significant bit). For example, for a 16-bit ADC, an error of one LSB is 1/ 65536 of the full signal range, or about 0.0015%.

Quantization error is due to the finite resolution of the ADC, and it is an unavoidable imperfection in all types of ADCs. So generally the magnitude of the quantization error at the sampling instant is between zero and half of one LSB. Quantization errors could also be considered as noise added to an otherwise perfect sample value and the effect of quantization noise is limiting the precision with which a digital sampled signal could represent the original analog signal. This inherent limitation of the ADC process is often expressed as a Signal-to-Noise Ratio (SNR), which is the ratio of the average power in the analog signal to the average power with the quantization noise. In terms of the dB scale, the quantization SNR for uniformly spaced sample levels increases by around 6dB for each bit used in the ADC sampling. For ADCs using R bits per sample and uniformly spaced quantization levels, SNR = 6R - 5 (approximately). Thus, a 24-bit encoding yields a theoretical SNR of 138dB, which is only limited by the electronics of the hardware itself.

All ADCs also suffer from non-linearity errors caused by their physical imperfections, causing their output to deviate from a linear function (or some other function, as in the case of a deliberately non-linear ADC) of their input. These errors can sometimes be mitigated by calibration or prevented by testing.

4.3 Off-nominal frequency component

In Section 2.4, in order to determine the relationship between the phasor angles and frequency tracking, the second term of (2.4.26) is negligible due to the small Y term as defined in (4.2.1)

$$Y = \frac{\sin\frac{\pi\Delta f}{Nf_0}}{N\sin\left(\frac{2\pi}{N}+\frac{\pi\Delta f}{Nf_0}\right)} \tag{4.2.1}$$

The relationship between the Y and Δf with $N=24$ is shown in Figure 4.20. It should be noticed that the Y term is not always so small that can be ignored, especially as $|\Delta f|$ gets larger.

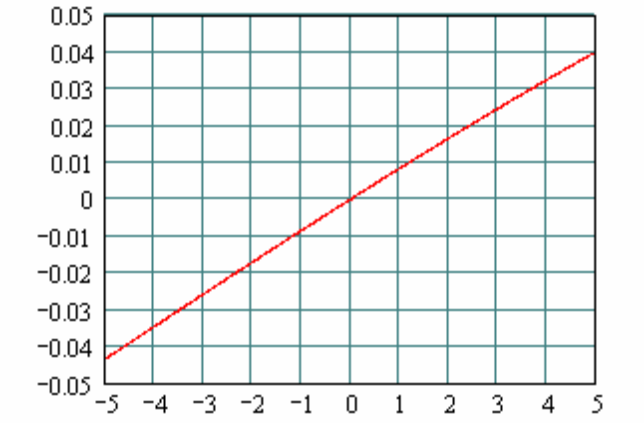


Figure 4.20 Y term versus Δf

In $\Delta f = f - f_0$, f is the real frequency of the input signal; f_0 is the assumed nominal frequency of the input signal, and the associated sampling rate for the signal is $f_s=Nf_0$. From Figure 4.20, one can see that when Δf is zero, the Y term is zero; and the smaller

$|\Delta f|$, the smaller the $|Y|$ term. In other words, the more closely the assumed nominal frequency f_0 is to real frequency f, the smaller the errors that will be introduced to frequency estimation.

It is very important to determine a proper sampling rate for the Frequency Disturbance unit. The measurement accuracy could be jeopardized if there are not enough samples depicting the real input signal, but too many samples could slow down the system with too much sample processing time. As discussed in the previous paragraph, a big part of the built-in errors of the algorithm comes from the assumption that the Y term defined in (4.2.1) is zero. Assuming the frequency deviation is within ±5Hz, plotting the Y term versus the samples per cycle N, Figure 4.21 is obtained. This figure is very helpful for the selection of a proper sampling rate. In can be noted that, for each value of Δf used in Figure 4.21, the Y term remains fairly stable when N is larger than 24. That is, an increase of the sampling rate will not help much for the error reduction of the Y term, if the value of N is already larger than 24. On the other hand, a lower sampling rate means less processing time, hence better real-time performance; so when almost the same accuracy can be achieved, the least samples per cycle are preferable.

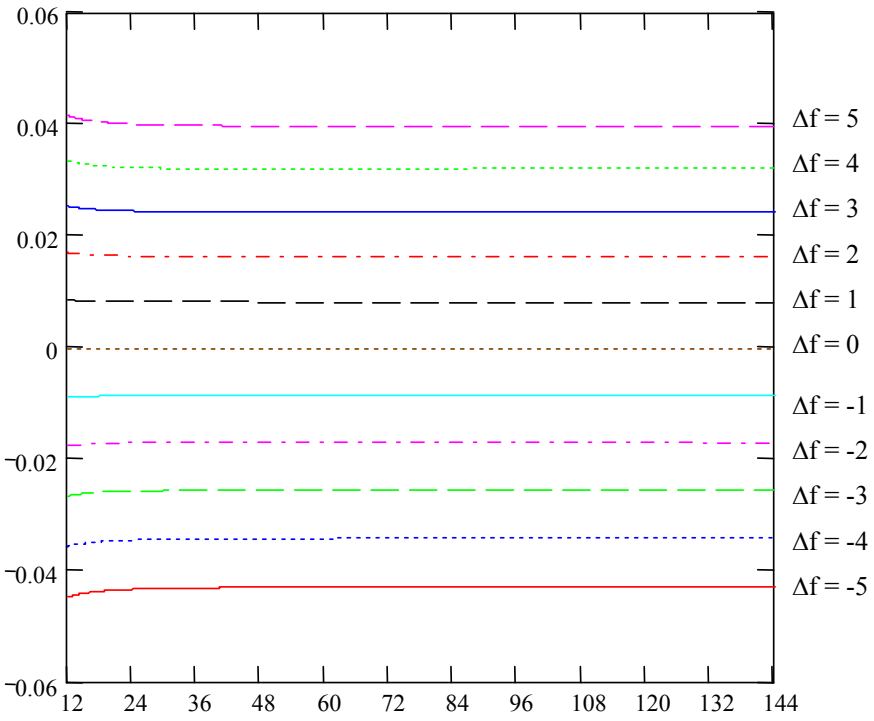


Figure 4.21 Y term effects cause by various sampling rates various Δf

In order to further investigate the issues of an off-nominal term, the phasor estimation of a single-phase quantity at off-nominal frequency is represented in following form:

$$X_r^{'} = PX_p\varepsilon^{jr\Delta\omega\Delta t} + QX_Q\varepsilon^{-jr(2\omega_0+\Delta\omega)\Delta t} \tag{4.2.2}$$

Where $X_r^{'}$ is phasor estimation at off-nominal frequency,

X_P is the correct value of phasor at off-nominal frequency,

X_Q is the conjungate of X_P,

$\Delta\omega = \omega - \omega_0$,

and

$$P = \left\{ \frac{\sin\frac{N(\omega-\omega_0)\Delta t}{2}}{N\sin\frac{(\omega-\omega_0)\Delta t}{2}} \right\} \varepsilon^{j(N-1)\frac{(\omega-\omega_0)\Delta t}{2}} \tag{4.2.3}$$

$$Q = \left\{ \frac{\sin\frac{N(\omega+\omega_0)\Delta t}{2}}{N\sin\frac{(\omega+\omega_0)\Delta t}{2}} \right\} \varepsilon^{-j(N-1)\frac{(\omega+\omega_0)\Delta t}{2}} . \tag{4.2.4}$$

It is apparent that for any practical power system $\Delta\omega$ is most likely to be very small, and $2\omega_0 + \Delta\omega$ very close to $2\omega_0$. Thus the resultant phasor has a magnitude and phase angle variation at a frequency of approximately $2\omega_0$ superimposed on a monotonically rotating component at $\Delta\omega$. Figure 4.22 is the typical plot of the angle of phasor by using the algorithm presented in Chapter 1.

From (4.2.2) and Figure 4.22, it can be seen that the phasor angle ripples are nearly twice the nominal frequency, so a simple averaging filter could be applied to eliminate the ripples. If we consider three samples of phasor estimate Xp, Xq, Xs obtained at t1, t1+T/6 and t1+T/3, and the 2rd harmonic components correspond to Xp, Xq, Xs are Xph, Xqh, and Xsh; we can see the Xqh and Xsh have shifts of 120 degrees and 240 degrees with respect to the first harmonic component. So Xph, Xqh, and Xsh will cancel each other if an average of them is taken.

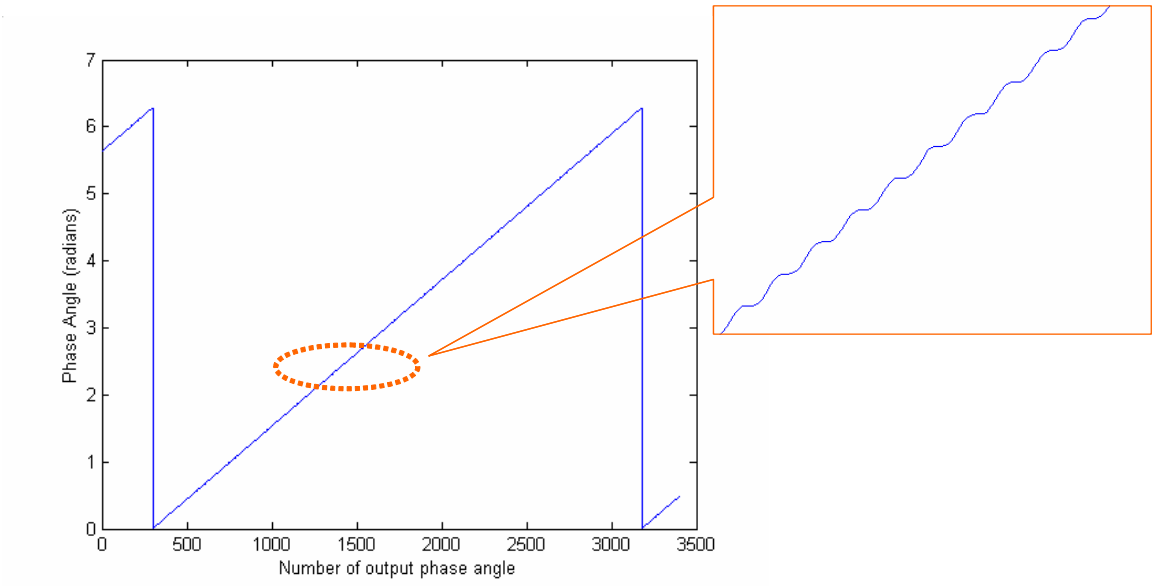


Figure 4.22 Frequency Estimation with Small Ripples

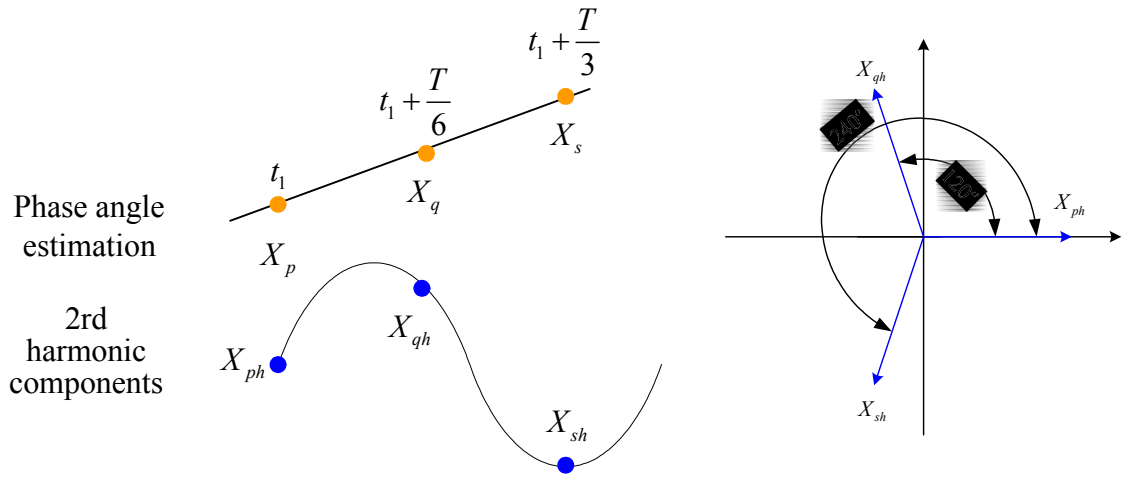


Figure 4.23 3-point Phasor Angle Averaging

For simulation, we applied this simple filter to a sequence of signal:

$$y = 169.7\sin(2\pi \times 60.5 \times \frac{1}{1440} + \frac{\pi}{4}) \tag{4.2.5}$$

The phasor angle estimation results are shown in Figure 4.24, and the frequency error could be obtained by referring to the Angle Averaging column of Table 3.

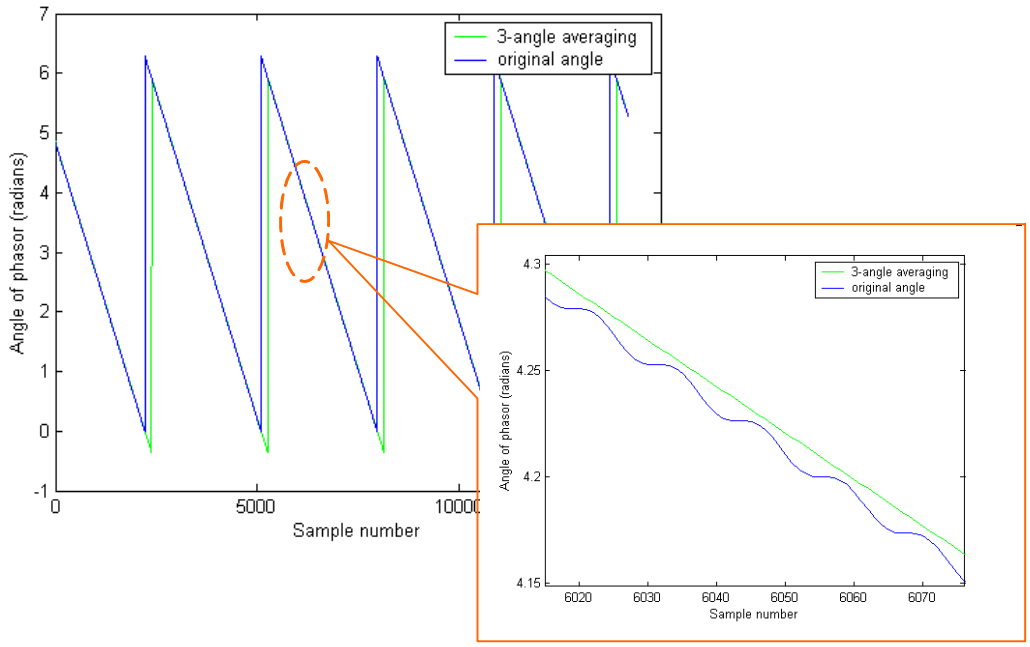


Figure 4.24 Frequency Estimation with 3-point Averaging Filter

Besides the three-point averaging techniques, there is another way to improve the accuracy of the frequency estimation, known as resampling [6]. Assuming the nominal frequency is 60Hz and the sampling rate is 1440 samples per second, which corresponds to $\frac{1440}{60} = 24$ samples per cycle, when the frequency has changed to 55 Hz, each cycle will now have $\frac{1440}{55} = 26.18$ samples. By re-normalizing through resampling, one tries to do a mathematical interpolation so that there will always be 24 samples per cycle regardless of the waveform frequency. To obtain the constant samples per cycle, the signal reconstruction method shown in Figure 4.11 will be used. After resampling points have been found, the phasor angles of the new sampled data are computed, the correction frequency $\Delta f'$ is obtained, and the final frequency estimation is computed using

$$f_{final} = f_0 + \Delta f + \Delta f' \tag{4.2.6}$$

to find the best way for frequency estimation; simulations with inputs of various frequencies and using different frequency estimation improvement techniques are performed.

In the simulations, the input has the following format

$$y = 169.7 \sin(2\pi f_a \times \frac{1}{f_s} + \frac{\pi}{4}) \tag{4.2.7}$$

where f_s is the sampling rate, which is 1440 in this case, and

f_a is the signal fundamental frequency, which ranges from 55Hz to 65Hz.

Table 3 Frequency estimation error range

	Range (Hz)		
f_a	Re-sampling	Angle Averaging	Angle averaging & Re-sampling)
55Hz	2.8000e-005	0.0028	3.0000e-006
56Hz	8.5000e-005	0.0053	7.0000e-006
57 Hz	8.6000e-005	0.0042	5.0000e-006
57.5 Hz	7.1000e-005	0.0029	4.0000e-006

58 Hz	3.0000e-005	0.0017	2.0000e-006
58.5 Hz	3.0000e-005	7.8900e-004	1.0000e-006
59 Hz	1.5000e-005	2.4500e-004	0
59.5 Hz	4.0000e-006	3.4000e-005	0
60Hz	0	0	0
60.5 Hz	6.0000e-006	4.6000e-005	0
61Hz	1.7000e-005	2.7900e-004	0
61.5Hz	3.1000e-005	7.9400e-004	1.0000e-006
62Hz	4.7000e-005	0.0016	2.0000e-006
62.5Hz	5.8000e-005	0.0025	2.0000e-006
63Hz	6.5000e-005	0.0033	4.0000e-006
64Hz	5.2000e-005	0.0033	4.0000e-006
65Hz	2.9000e-005	0.0019	3.0000e-006

Table 3 lists the frequency estimation ranges by using resampling, three-phase-angle averaging, and a combination of resampling and three-phase-angle averaging. It seems that resampling and three-phase-angle averaging combination could greatly improve the frequency estimation accuracy, especially when the fundamental frequency is between 59Hz and 61Hz; the estimation error is almost 0. However it must be noted that using a combination of resampling and three-phase-angle averaging will add more computation burden to the microcontroller, and will impact the real-time performance of the algorithm. So when the computation time is the concern, we can turn to using the resampling technique only, since it provides desirable estimation accuracy with acceptable real-time performance.

However, considering one raw voltage data sequence of a power grid, which includes harmonics and random noise, the resampling and three-phase-angle averaging combination technique doesn't work very well, as seen in Figure 4.24, which shows the phasor angle before and after applying the three-point average filter. In the real system, the errors due to other noise are much greater than the error caused by the $2\omega_0$

component, so the three-point averaging filter doesn't help to improve the frequency estimation accuracy in practical power system measurement.

It is also not surprising to see in Figure 4.25 and Figure 4.26 that the frequency estimation obtained by using resampling and by using the resample and three-point averaging are almost the same. It can be concluded safely that three-point averaging, resampling and their combination could improve the frequency estimation accuracy; however, for both computation speed and accuracy concerns, the resampling is the best choice for FDRs.

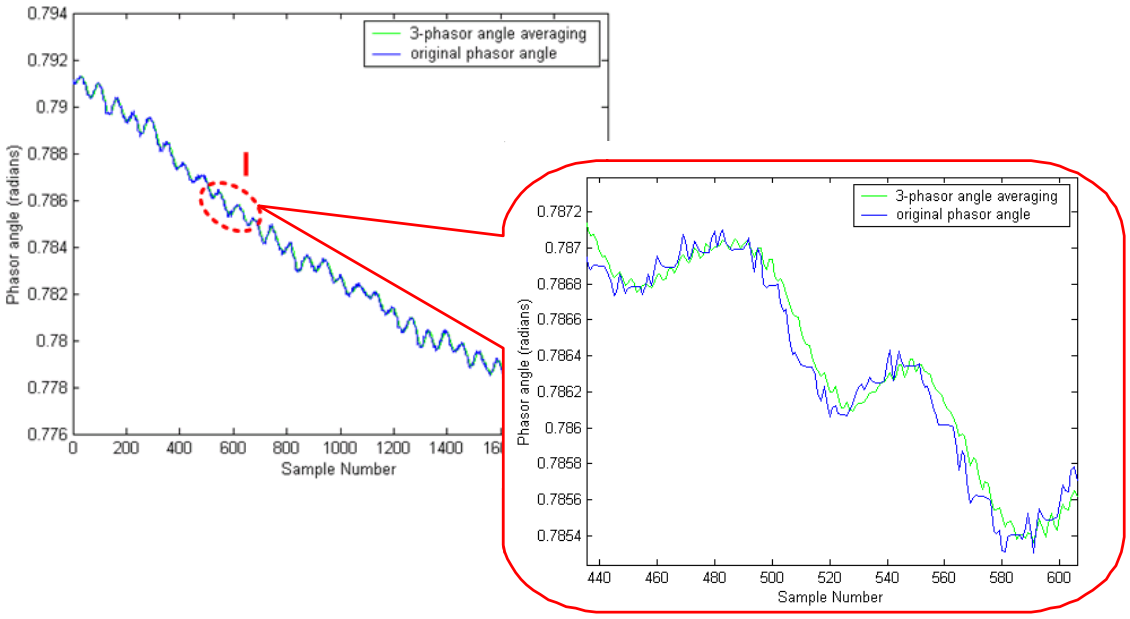


Figure 4.25 Phase Angle of unfiltered raw data from real power system

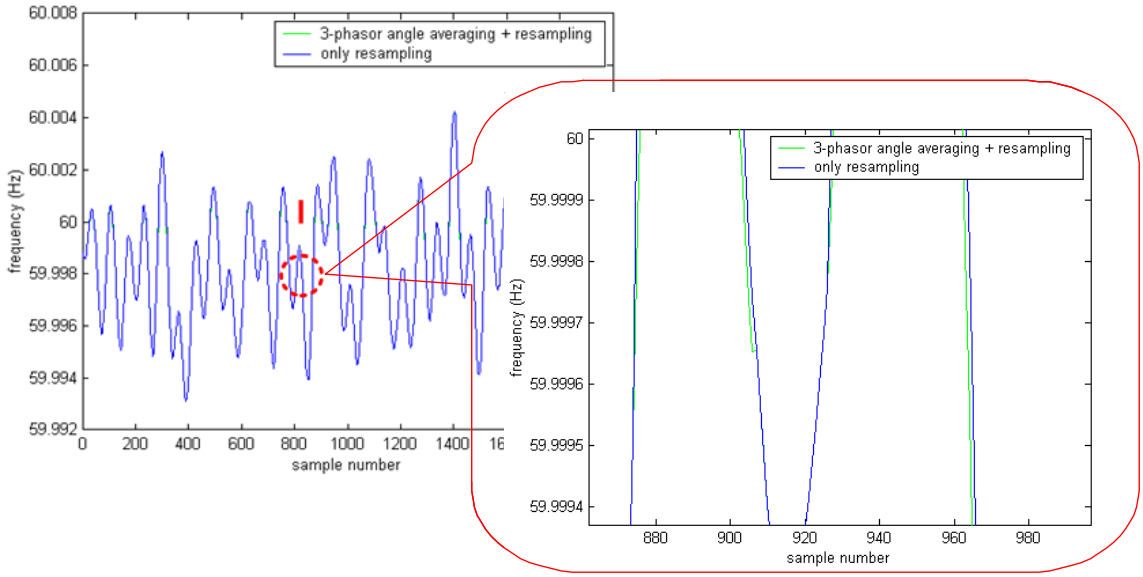


Figure 4.26 Frequency estimation of unfiltered raw data from real power system

4.4 Noise Analysis

A study published by the Electric Power Research Institute[27] has classified power system noise as a combination of

- White noise
- Harmonics noise
- Random spike noise.

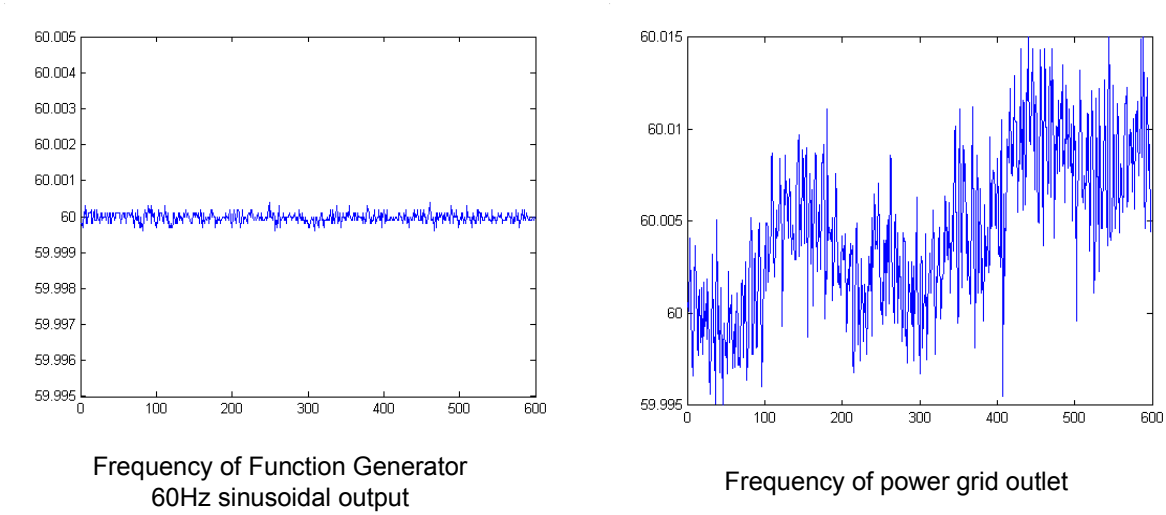


Figure 4.27 Frequency estimation comparison

Noise has been the ultimate challenge for precision measurement since it introduces errors in the measured qualities. In Figure 4.27, the left plot shows the frequency estimation results with an ideal 60Hz sinusoidal input, and the right plot is the frequency estimation results with unfiltered raw input from a wall outlet. The same software process is used to perform the estimation, and it appears that noise and non-fundamental frequency components in the raw wall outlet input smeared the estimation results. Hence, the reduction or elimination of the influence of noise and non-fundamental components in the measurement has been a vital objective.

4.4.1 Amplitude swing effects

When the algorithm was developed, an assumption was made that the input voltage waveform is purely sinusoidal. However, in some transient conditions, the voltage waveform of the real power system may experience amplitude swings. There are reasons for such voltage swings, the primary reasons being sudden load variations, transient faults and line switching [38]. Consider the voltage waveform with swing as the following:

$$X(t) = \sqrt{2}X\sin(2\pi ft) + \sqrt{2}Xm\sin(2\pi f_m t)\sin(2\pi ft) \tag{4.4.1}$$

in which X is RMS value of voltage,

f is the fundamental frequency,

m is modulation index, and

f_m is the modulation frequency .

Typically, in a real system the modulation index will be less than 10%, while the modulation frequency will be around 1Hz. In order to maintain generality, simulations based on a voltage waveform with 5% modulation and 1Hz modulation frequency were performed.

At the nominal frequency, the estimation error of the frequency is a function of the modulation index and modulation frequency, and the gradual effects of the voltage swing are presented in the frequency estimation[38]. More detailed effects of the voltage swing of 10% modulation index and 1Hz modulation frequency are included in Table 4.

Table 4 Errors introduced by amplitude swing

F(Hz) / Error	57	58	59	60	61	62	63
Max(10^{-3} Hz)	0.46	0.44	0.42	0.41	0.41	0.42	0.42

The simulations results show that frequency estimation accuracy won't change much as the fundamental frequency changes.

4.4.2 Harmonics effects

Harmonics is a big concern for real power quality, and the unfiltered low-order harmonics will be presented in the voltage waveform for measuring purposes. A 1440Hz sampling frequency allows twelve harmonics to be presented in the signal without aliasing effects. While some even harmonics may exist, odd harmonics are usually dominant[39, 40]. It is essential that the signal should not contain components with frequencies above the Nyquist rate $fs/2$. The waveform that has a harmonic component while maintaining the constant operating frequency is presented as:

$$X(t) = \sqrt{2}X\sin(2\pi ft) + \sum_{i=1}^{N} p_{2i+1}\sin\left(2\pi\left((2i+1)f\right)t\right) \tag{4.5.1}$$

in which X is RMS value of voltage,

f is the fundamental frequency, and

p_{2i+1} is the $(2i+1)$ th harmonic percentage.

Table 5 Maximum Measurement errors due to harmonics presence

Harmonic Component (%)	Maximum measurement error for different frequency (mHz)		
	$f = 57$	$f = 60$	$f = 63$
5% 3rd	0.0523	0	0.0592
5% 5th	0.0573	0	0.0771
10% 3rd	0.0593	0	0.0862
10% 5th	0.0693	0	0.122

Table 5 shows the effects of harmonics in the frequency estimation calculation. The more harmonics contents there are, the more maximum measurement errors there are in the frequency estimation. Although the measurement errors are fairly small when compared with the algorithm errors and other implementation errors, generally a simple band pass filter could be applied to eliminated the harmonics components and reduce the measurement errors.

4.4.3 Digital filter

In general, digital filters are used for two general purposes:

- Separation of signals that have been combined, and
- Restoration of signals that have been distorted.

A digital filter can use a digital processor to perform numerical calculations on sampled values of a signal. The processor may be a general-purpose computer, microcontroller or a specialized DSP chip. Typically, there are two kinds of traditional digital filters: finite impulse response filters (FIR) and infinite impulse response filters (IIR) [41].

FIR digital filters use only current and past input samples and none of the filter's previous output samples to obtain a current output sample value. FIR filters are also called non-recursive filters. IIR digital filters always require feedback, and each IIR filter output sample depends on previous input samples and previous filter output samples.
The z-domain transfer function of a FIR filter (all-zero) is

$$H(z)=\frac{Y(z)}{X(z)}=\sum_{k=0}^{M}b(k)z^{-k}\,. \tag{4.5.2}$$

The difference equation of a FIR filter is

$$y[n]=\sum_{k=0}^{M}b_k x[n-k]\,. \tag{4.5.3}$$

The z-domain transfer function of an IIR filter is

$$H(z)=\frac{Y(z)}{X(z)}=\frac{\sum_{k=0}^{M}b(k)z^{-k}}{\sum_{k=0}^{N}a(k)z^{-k}}\,. \tag{4.5.4}$$

The difference equation of IIR is

$$a_0 y[n]+a_1 y[n-1]+\cdots+a_N y[n-N]=b_0 x[n]+b_1 x[n-1]+\cdots+b_M x[n-M]\,. \tag{4.5.5}$$

Both FIR and IIR filters have advantages and disadvantages when compared to each other. Although IIR filters have a nonlinear phase, the primary advantage of IIR filters over FIR filters is that they typically meet a given set of specifications with a much lower filter order than a corresponding FIR filter. The most dominant IIR filters are Butterworth, Chebyshev Types I and II, Elliptic and Bessel.

FIR filters have the following primary advantages:

- They can have exactly linear phase.
- They are always stable.
- The design methods are generally linear.
- They can be realized efficiently in hardware.
- The filter startup transients have finite duration.

The primary disadvantage of FIR filters is that they often require a much higher filter order than IIR filters require to achieve a given level of performance.

In the following sections, the two most commonly used filters, moving average (FIR) and Butterworth (IIR), are designed to investigate the possibilities of incorporating digital filter into FDRs.

4.4.4 FIR moving average filter

The moving average filter is the most commonly used filter in digital signal processing, and it is optimal for reducing random noise while retaining a sharp step response. As the name implies, the moving average filter operates by averaging a number of points from the input signal to produce each point in the output signal. The moving average filter can be expressed as:

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j]. \tag{4.5.6}$$

As an alternative, if M is odd, the group of points from the input signal can be chosen symmetrically around the output point:

$$y[i] = \frac{1}{M} \sum_{j=-\frac{M-1}{2}}^{\frac{M-1}{2}} x[i+j]. \tag{4.5.7}$$

In the following simulations 5-point, 11-point and 65-point moving average filters are applied to a voltage sequence with a sample rate of 100KHz. Figure 4.28, Figure 4.29 and Figure 4.30 show the voltage waveforms before and after the filters, and one can notice that the filters smooth the waveform and make the waveforms closer to a pure sinusoidal waveform. Furthermore, these figures show that more points for averaging yield better results.

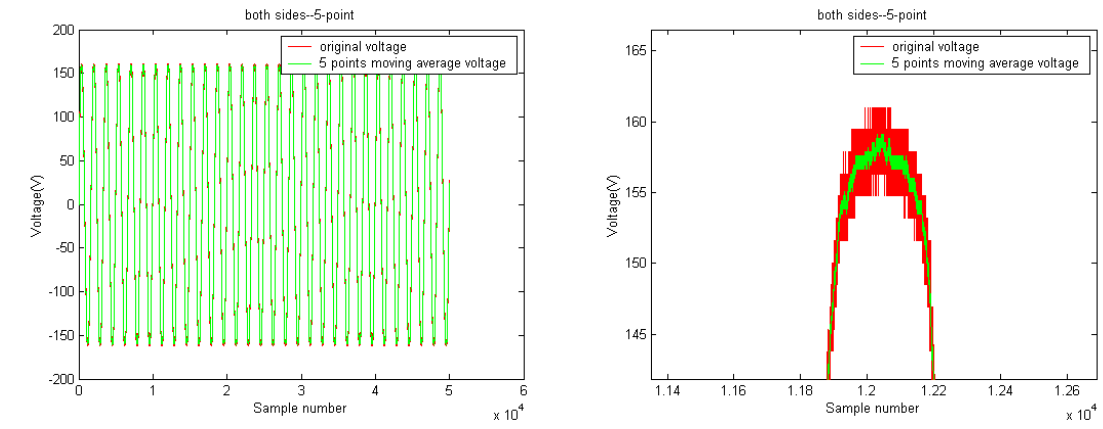


Figure 4.28 Voltage Waveform w/o 5-point Moving Average Filter

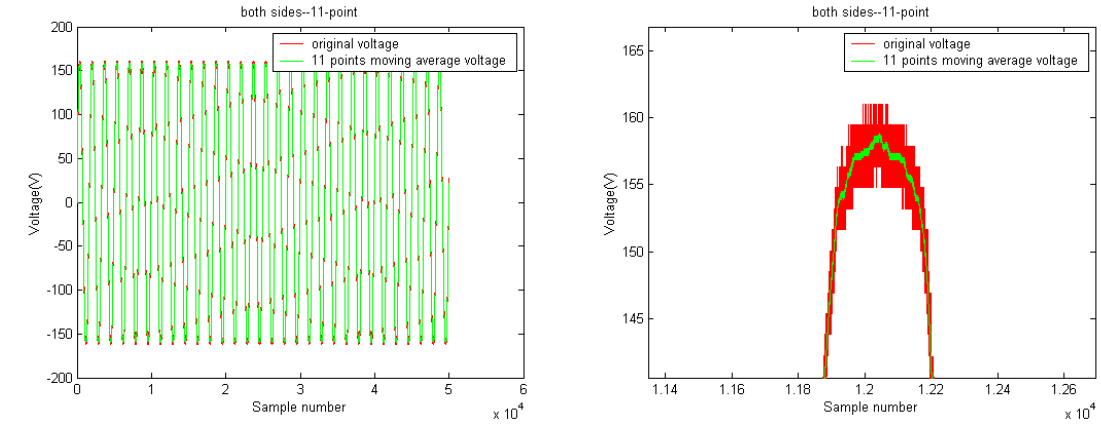


Figure 4.29 Voltage Waveform w/o 11-point Moving Average

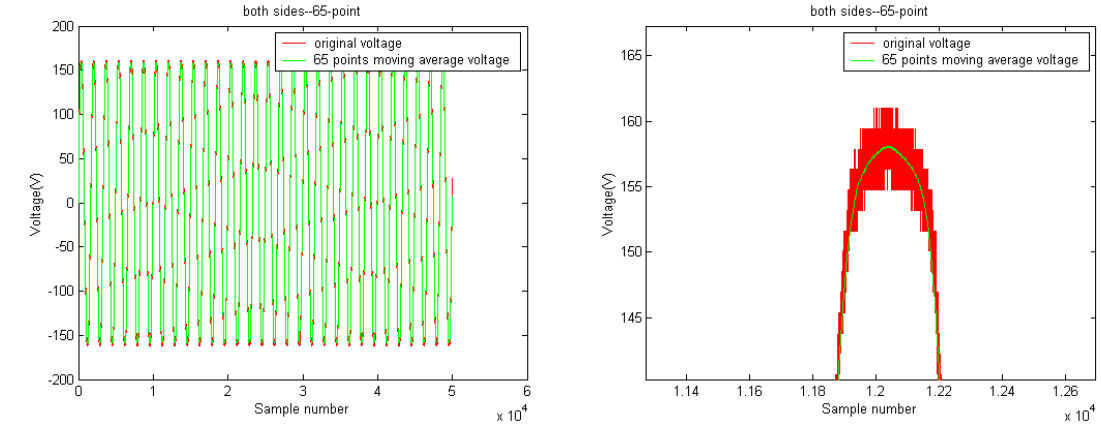


Figure 4.30 Voltage Waveform w/o 65-point Moving Average

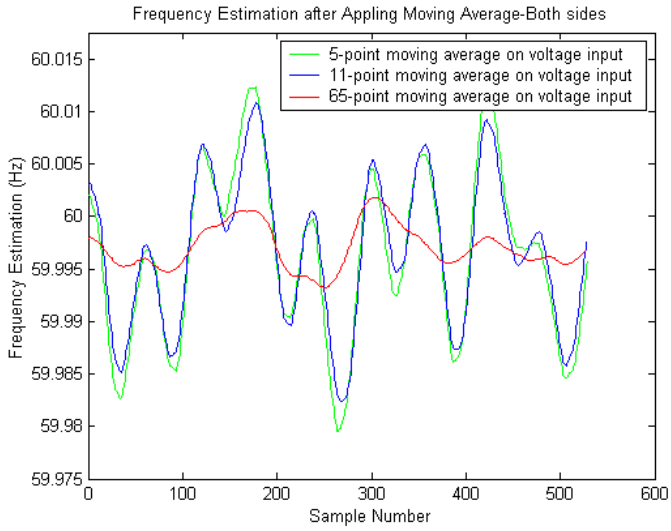


Figure 4.31 Frequency Estimation with Moving Average Filters

Figure 4.31 is the frequency estimation after applying the filters to the voltage waveform. It shows that with a 65-point moving average filter, the frequency estimation can get much more accurate.

4.4.5 IIR bandpass Butterworth filter

The IIR Butterworth filter is another popular choice for most DSP applications. The Butterworth filter has no amplitude ripples in either the passband or the stopband; however, for a given filter order, the Butterworth design has the widest transition region of most filters. There is also a bandpass Butterworth filter. The specifications of the filter are: the attenuation at cutoff frequency is fixed at 3dB, the sample rate is Fs=1440Hz, the cutoff frequencies are Fc1=40 Hz and Fc2 =120Hz, and the filter order is 12. The simulation results show that the bandpass Butterworth filter designed could improve the frequency estimation results when the input is a pure sinusoidal waveform. However, it should be noted that the filter doesn't work well at the inputs where the phase shift, amplitude shifts or other noise is present. In the meantime, a high-order IIR filter is very resource demanding, which could slow down the entire system's performance. Further study and research on IIR filter design and implementation are highly recommended.

4.5 FDR Placement and Communication

Similar to the PMU location analysis, the location of the FDRs' placement should effectively represent the different frequency clusters of inter-area oscillations and cover as broad an area as possible in order to capture the dynamic behavior of a larger system disturbance [42].

The North American Electric Reliability Council, or NERC for short, is a self-regulatory organization, relying on reciprocity and the mutual self-interest of all those involved. Its mission is to ensure that the bulk electric system in North America is reliable, adequate and secure. NERC published the NERC regions and balancing authorities, as shown in Figure 4.32 [43], which is an excellent reference for FDR host location selection.

As discussed in Chapter 3, the FDR units are portable devices and they are easy to install and relocate without additional cost. The FDR can be placed at homes, school labs, and business offices, demanding minimal resources. Presently, the FDR locations are selected to cover the whole U.S power grid, and most of the FDR units are located close to major generation centers, major transmission tie lines and load-concentrated areas. Figure 4.33 shows the FDR location map of FNET, and Table 6 lists the units in service currently, as well as their location and respective reliability council. As more FDRs are shipped out and installed, this map grows denser, and the FNET becomes more meaningful and helpful for power system analysis.

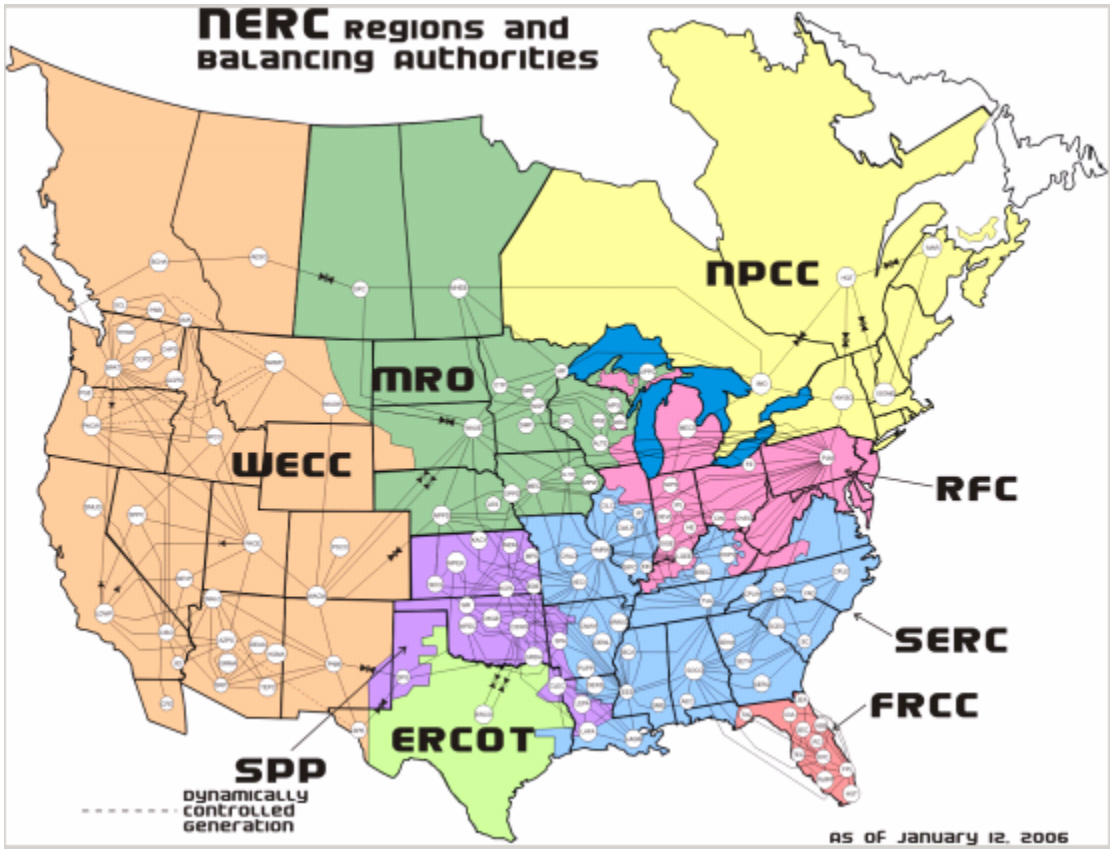


Figure 4.32 NERC regions and balancing authorities

Table 6 FDR Unit Location and RELATIVE SYSTEM Area

Unit ID	Unit Name	Reliability Council	Location
1	NY	NPCC	Schenectady, NY

2	UMR	SERC	Rolla, MO
3	ARI	SERC	Alexandria, VA
4	VT	SERC	Blacksburg, VA
5	Seattle	WECC	Seattle, WA
6	ABB	SERC	Raleigh, NC
7	MISSSTATE	SERC	Starkville, MS
8	CA	WECC	Palo Alto, CA
9	UFL	FRCC	Gainesville, FL
11	Calvin	RFC	Grand Rapids, MI
12	Houston	ERCOT	Houston,TX
13	Midwest ISO	RFC	Carmel,IN
14	ASU	WECC	Tempe, AZ
15	TVA2	SERC	Nashville,TN
16	LA	WECC	Pasadena,CA
17	Tulane	SERC	New Orleans,LA
18	NY	NPCC	New York City, NY
20	TVA1	SERC	Huntsville,TN
21	WSU	RFC	Pullman,WA
22	ROA	SERC	Roanoke,VA
23	FSU	FRCC	Tallahassee,FL
24	Toronto	NPCC	Toronto,ON CA
26	Louisville	SERC	Louisville,KY
27	RPI	NPCC	Troy,NY
28	METC	RFC	Grand Rapids,MI
29	TAMU	ERCOT	College Station,TX
33	UT	SERC	Knoxville,TN

The FNET is a wide-area communication network built on client/server architecture, and FDRs are strategically placed throughout the entire US. The communication and resource sharing between the two main parts of FDRs and the center data server connect all the

components of FNET into a seamless whole. For some specific real-time applications of FNET, such as network control and relaying [44]; communication delay, reliability and security could be a bottleneck. Considering the communication delay, even with the fiber-optic cables, the worst delay could be over 150 milliseconds [45]. However, this limitation will not affect other possible applications, especially the post-processing applications, such as monitoring and model verification.

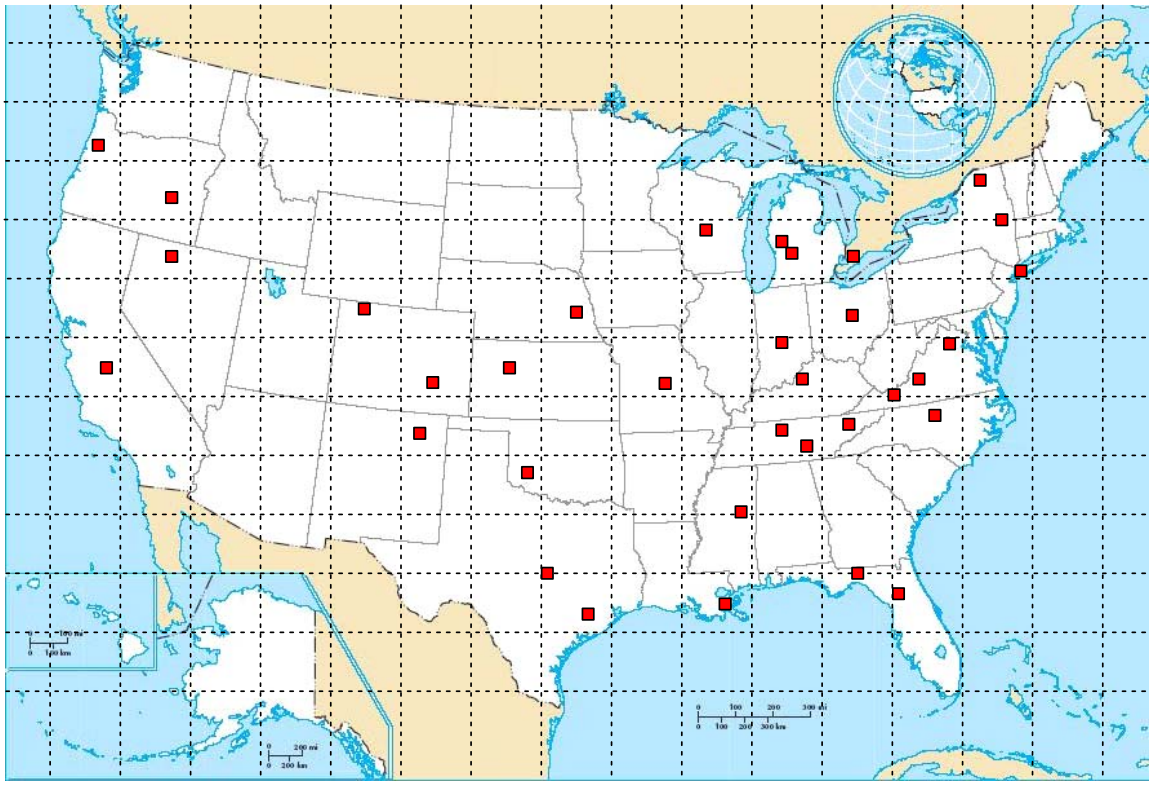


Figure 4.33 FDR placement map

Data redirecting could be the first and simplest way to interface between these systems, but this could cause more communication delays and waste network resources, since the desire for connection will continually build once one communication channel is desired. A better solution is IP multicasting over IGMP (Internet Group Management Protocol), which allows the FDRs' broadcast content to be available to a large, flexible number of subscribers effectively. Through IGMP, clients may request to join or leave a certain multicast group at will, and the network traffic is sent only to the clients registered to a specific GDA (Group Destination Address).

4.6 Unit Testing and Verification Results

In order to verify the accuracy of FDR frequency estimation, several test plans have been developed and executed. Figure 4.34, Figure 4.35 and Figure 4.36 are frequency estimations with pure sinusoidal input of frequencies 57Hz, 60Hz and 63Hz respectively. These test results show that an FDR can provide very accurate estimation in a broad range of frequencies. The test result in Figure 4.37 shows that the FDR has a good dynamic performance and can track the frequency changes precisely. The comparison test results between two FDRs shown in Figure 4.38 demonstrate the great synchronization performance between different FDR units.

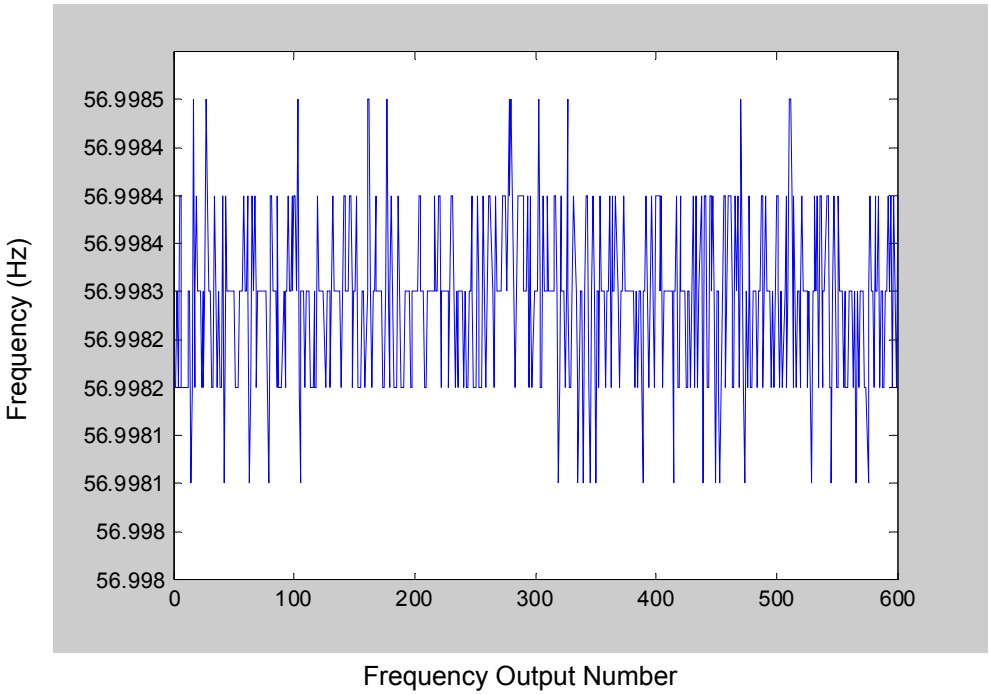


Figure 4.34 FDR frequency estimation with 57 Hz sinusoidal input

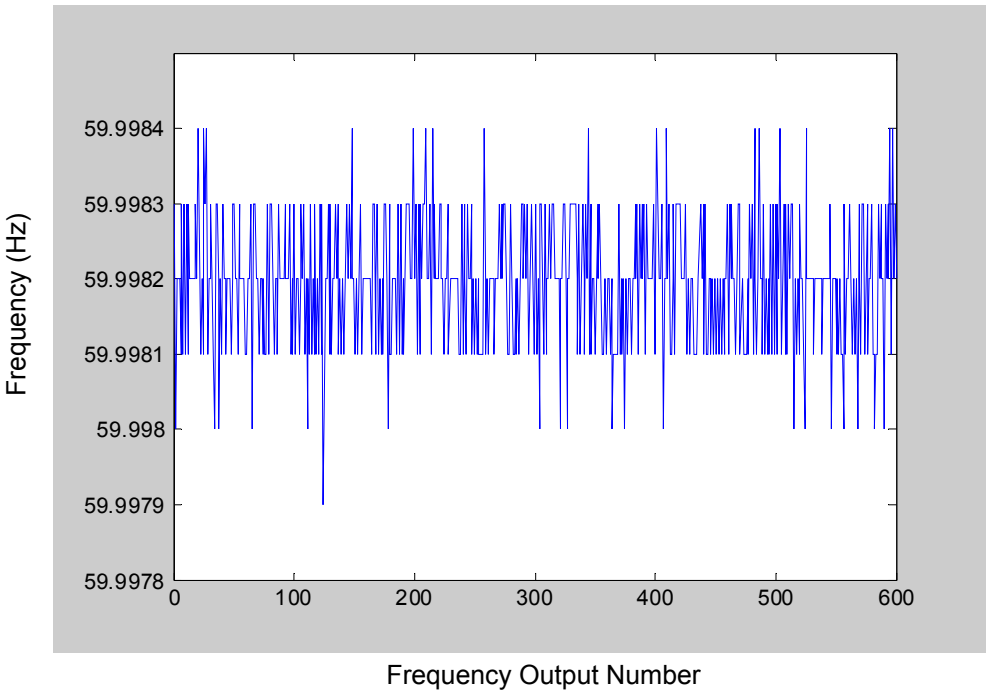


Figure 4.35 FDR frequency estimation with 60 Hz sinusoidal input

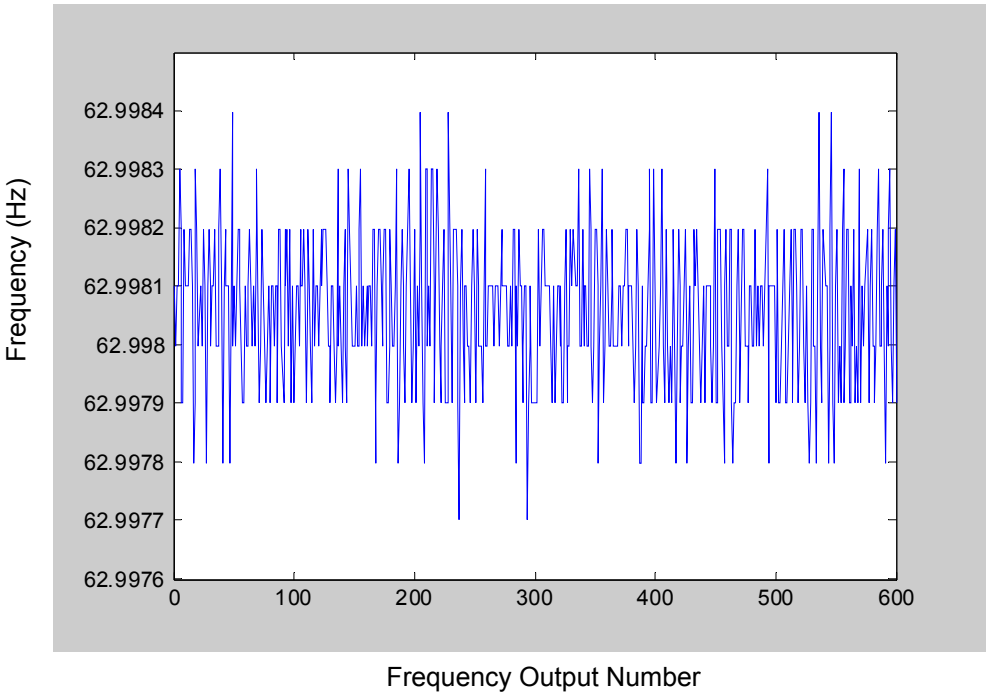


Figure 4.36 FDR frequency estimation with 63 Hz sinusoidal input

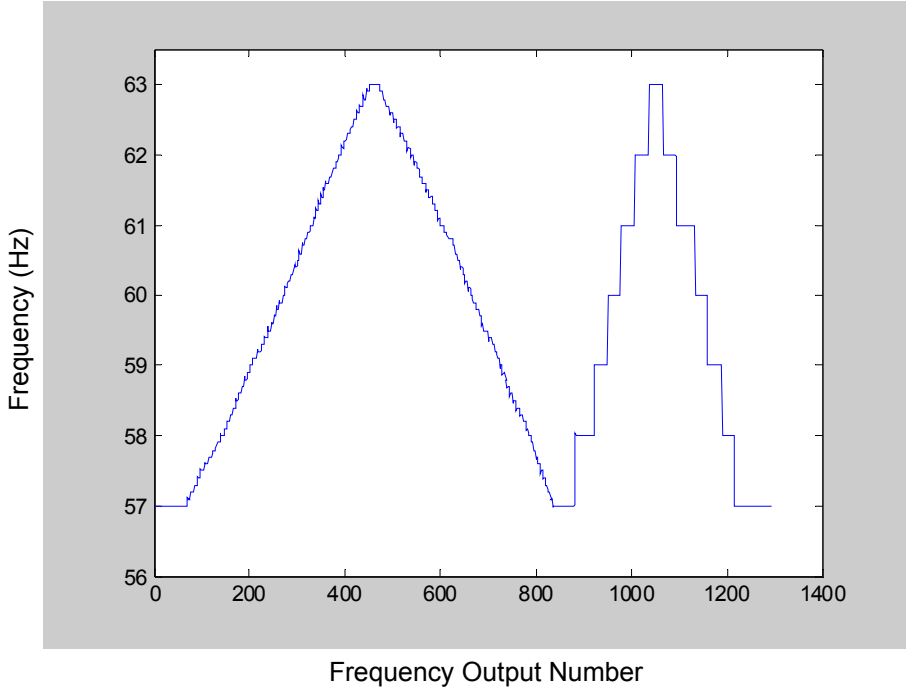


Figure 4.37 FDR frequency estimation with dynamic sinusoidal waveform input

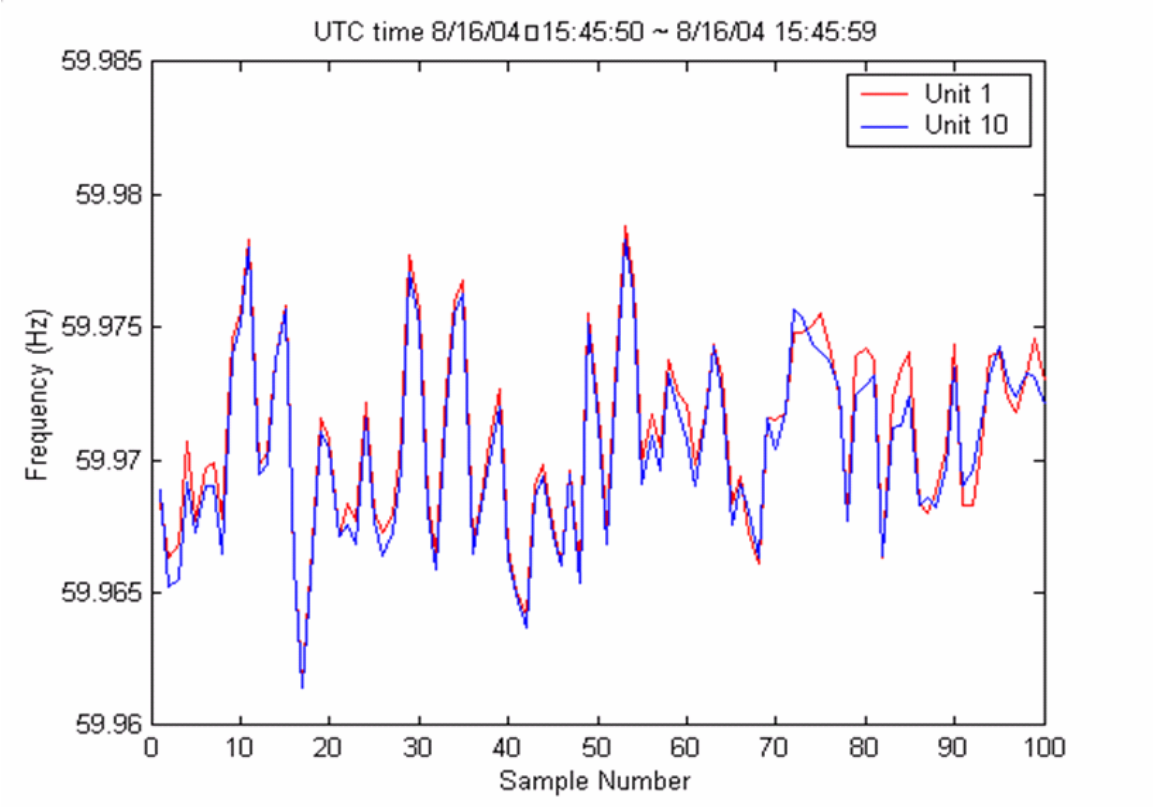


Figure 4.38 Frequency estimation comparison of 2 FDRs with voltage input from the same wall outlet

Some comparison tests also have been executed between FDRs and commercial PMUs (Phasor Measurement Unit). Figure 4.39, Figure 4.40 and Figure 4.41 show the test results of commercial PMUs from different vendors. Figure 4.42 is the FDR testing results. All PMUs and the FDR are connected to the same power outlet, and all the testing results are referenced to a common time standard, UTC time. All PMU testing results are filtered data, and the FDR frequency profile is raw data without any post-processing. The testing results show that all tested PMUs and the FDR can measure the frequency change very precisely, and the FDR can work as well as the commercial PMUs with respect to the frequency measurement.

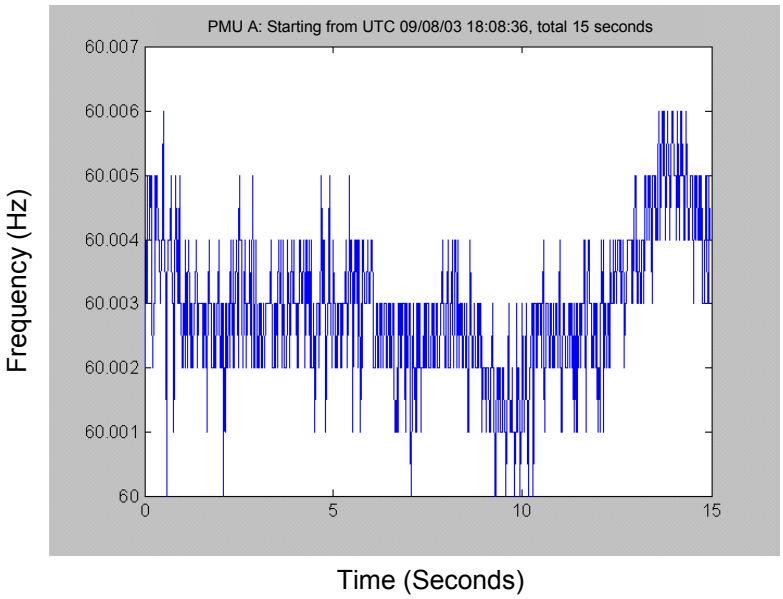


Figure 4.39 PMU A frequency estimation results

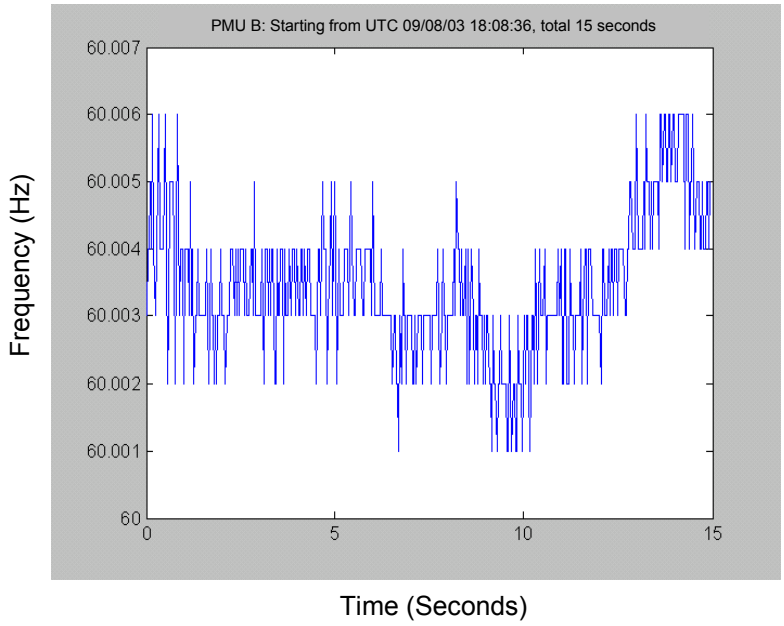


Figure 4.40 PMU B frequency estimation results

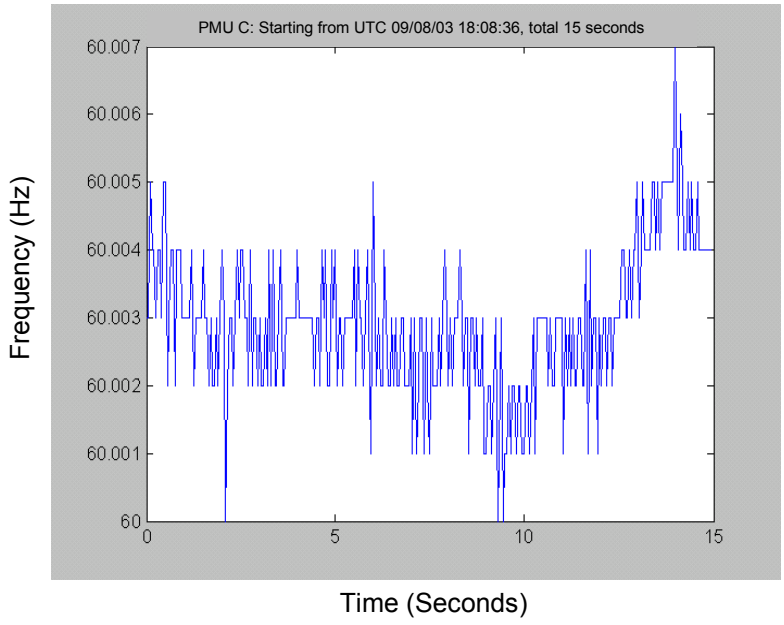


Figure 4.41 PMU C frequency estimation results

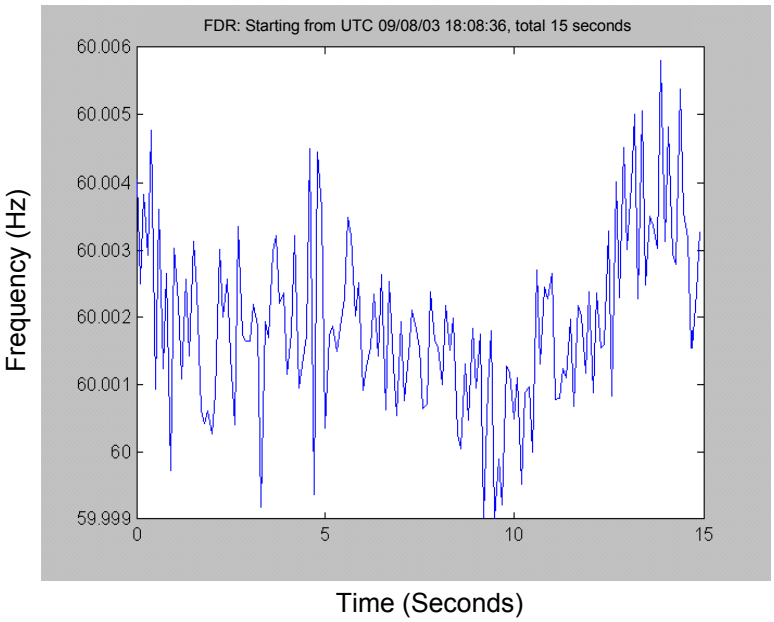


Figure 4.42 FDR frequency estimation results

4.7 Missing Point

The phasor angle analysis technique is based on the assumption that the number of samples between successive 1PPS signals is an integer, and those samples are evenly spaced throughout the second. In the FDR, the number of samples is 24 samples/cycle or 1440 samples for a 60Hz system; and 1200 samples for a 50Hz system. The samples are numbered from 0 to L-1, where L is the number of samples in one second. Sample 0 is the first sample in a new second and is locked with the rising edge of the 1PPS pulse within $\pm 100ns$.

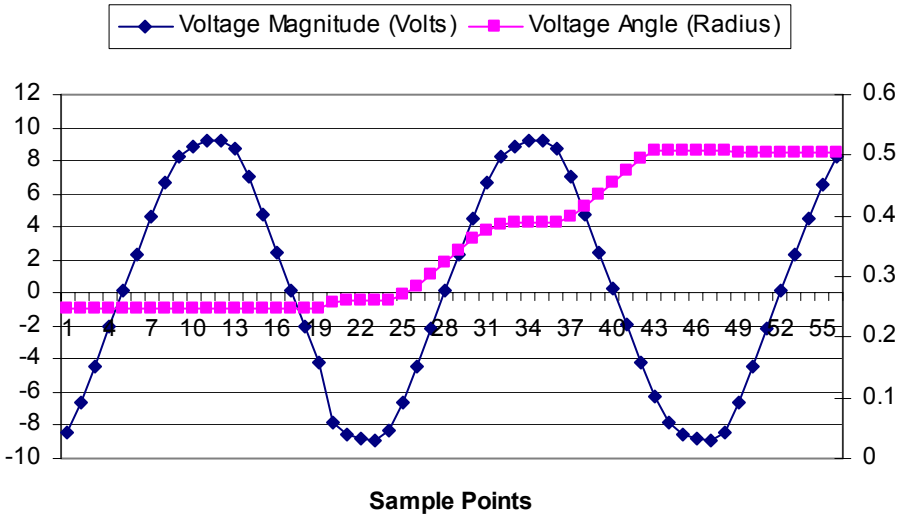


Figure 4.43 Voltage samples and voltage angles in a missing point case

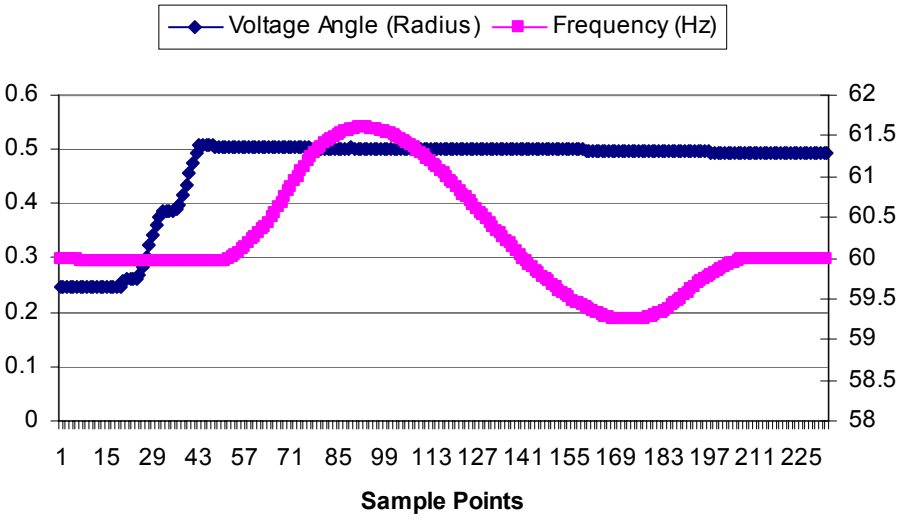


Figure 4.44 Frequency and voltage angles in a missing point case

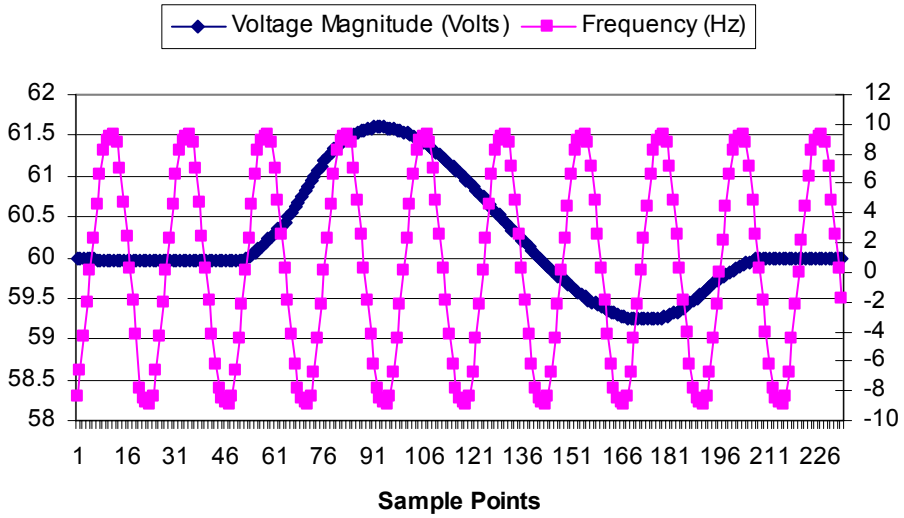


Figure 4.45 Frequency and voltage samples in a missing point case

However, due to the unbalance between the load and generation in transient conditions, TA/TV nonlinear, loss of GPS synchronization signal, and other system errors; the frequency of power system fluctuates around the nominal frequency, and the voltage-sampling scheme does not work perfectly all the time. Some cases of missing samples have be observed from time to time. As shown in Figure 4.43, Figure 4.44 and Figure 4.45, a missing voltage sample can result in a big spike in the frequency estimation. It is hard to avoid all the causes of a missing point, but strictly enforcing the PWM pulse train to restart upon every PPS signal has proved to be an effective way to minimize the occurrence of a missing point.

Chapter 5 FNET Implementation and Applications

5.1 FNET Implementation

The Frequency Monitoring Network (FNET) is a cost-effective wide-area frequency measurement system with high dynamic accuracy, and it can provide valuable frequency information for a variety of power system operation, analysis, and control applications. The FNET system currently consists of 30 FDR units geographically dispersed throughout the continental United States, and an Information Management System (IMS) located at Virginia Tech. The ultimate goal of FNET is to have approximately 50 FDR units around the nation and covering all the NERC regions and balancing authorities, as illustrated in Figure 4.32. The locations of FDRs are carefully selected to reflect different frequency behaviors of inter-connection oscillations, and most FDRs are close to major generation centers, major transmission tie lines, and load-concentrated areas. Corresponding with changing demands, FDRs can be easily relocated without additional installation costs. These FDRs monitor the changing frequency continuously in different locations, and transmit the time-synchronized data to the server every 10^{th} of a second. Many details of FDR design, implementation and placement have been fully discussed in previous chapters.

The IMS is a cluster of workstations and computers that models, coordinates, and integrates the frequency acquisition, processing and display functions. The IMS is designed based the multi-tier structure shown in Figure 5.1, which provides enhanced performance, flexibility, reusability and scalability reliability for the server, while minimizing the resources and complexity of distributed processing. All of the IMS's communication is based on Internet infrastructure, which allows the components of the IMS to be geographically dispersed to different locations. Overall the IMS consists of

three main functional components: an application server program, a database operation service, and a web service.

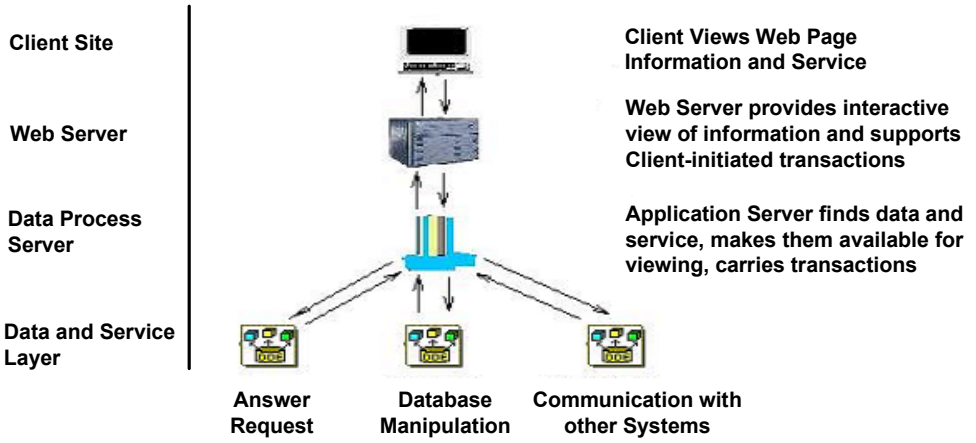


Figure 5.1 Multi-tier Information Management System Structure

The application server program manages and communicates with remote FDRs, collects the real-time frequency measurement data, and process the frequency information for database management and web services. The database operation service manages the data storage and queries in the database. The web service provides data display, and post-processing for a variety of monitoring and analysis purposes. Authorized users can access the FNET web service anywhere via the Internet. Figure 5.2 is a wide-area frequency dynamic map providing a visualization of the real-time frequency information in a geographic context within each interconnection (WECC, Eastern, and ERCOT). This enables the observation of the real-time power system performance nationwide. Figure 5.3 is the web-based FNET information display, and Figure 5.4 is the FNET data streamer. Both are updated every five seconds, and provide color-coded frequency information and the communication status. All of these web services construct the base function of a frequency monitoring server system.

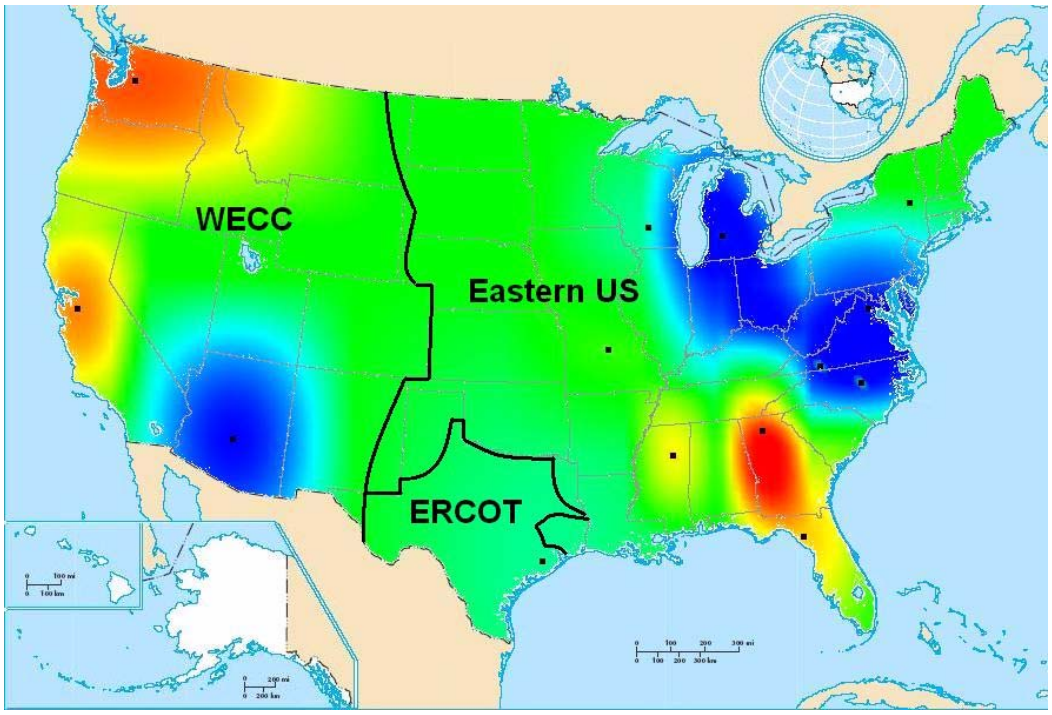


Figure 5.2 Frequency dynamic data visualization map

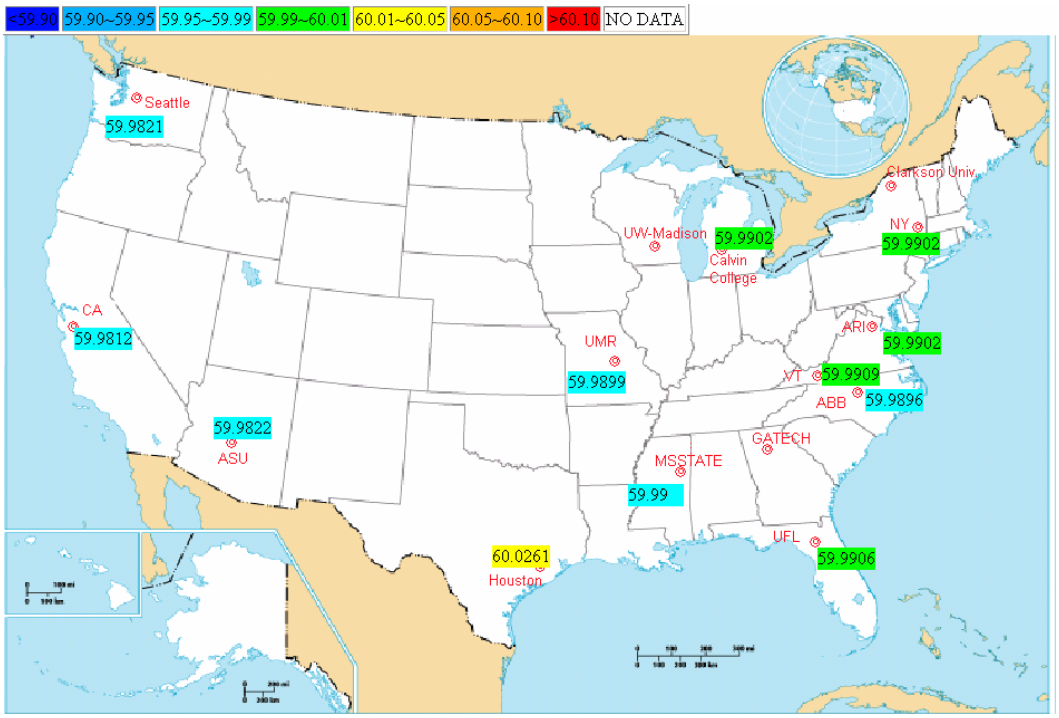


Figure 5.3 FNET data display

Welcome to FNET Server of Virginia Tech
at The Bradley Department of Electrical and Computer Engineering

Color Vs Frequency

<59.90	59.90~59.95	59.95~59.99	59.99~60.01	60.01~60.05	60.05~60.10	>60.10	NO DATA

Time Zone:UTC

NY	NO DATA	NO DATA	NO DATA	NO DATA	NO DATA	NO DATA	NO DATA	NO DATA	NO DATA	NO DATA
UMR	60.0111	60.0129	60.0122	60.0108	60.0094	60.0139	60.0157	60.0158	60.0136	60.0125
ARI	60.0108	60.0124	60.0123	60.0105	60.0092	60.0137	60.016	60.0153	60.0133	60.0128
VT	60.0112	60.0124	60.0117	60.0107	60.0093	60.0134	60.0154	60.015	60.0119	60.0111
MISS	60.0118	60.012	60.0127	60.0106	60.0098	60.0148	60.0163	60.0158	60.0145	60.0128
UFL	60.0114	60.0138	60.0118	60.0079	60.0095	60.0144	60.0159	60.0159	60.0133	60.0117
Calvin	60.011	60.0126	60.0121	60.011	60.0094	60.0131	60.0152	60.0155	60.0136	60.0122
ABB	60.01	60.012	60.0088	60.0099	60.0089	60.0131	60.0146	60.0157	60.0132	60.0119
Tulane	60.0107	60.0113	60.0116	60.0105	60.0095	60.013	60.0162	60.0149	60.0134	60.0114
Seattle	60.0144	60.0105	60.0058	60.0008	59.9996	59.9978	59.9964	59.9956	59.9972	60.0005
EPRI	60.0134	60.0098	60.0052	60.0013	59.9973	59.9964	59.996	59.994	59.997	59.9997
ASU	60.015	60.0118	60.0097	60.0033	59.9995	60.0032	60.0005	59.9992	59.9976	60.0027
LA	60.0148	60.0121	60.0065	60.0013	59.9996	59.9985	59.9961	59.9965	59.9977	60.0018
Houston	60.0614	60.0598	60.0589	60.0621	60.0659	60.0668	60.0753	60.0743	60.0662	60.0732
UTC Time	02/18/2005 18:24:00	02/18/2005 18:24:05	02/18/2005 18:24:10	02/18/2005 18:24:15	02/18/2005 18:24:20	02/18/2005 18:24:25	02/18/2005 18:24:30	02/18/2005 18:24:35	02/18/2005 18:24:40	02/18/2005 18:24:45

Figure 5.4 FNET Data stream of 5-second average

5.2 FNET Applications Overview

FNET provides time-synchronized power system parameters, voltage magnitude, voltage angle, and frequency, which is meaningful information and can be used in monitoring, analysis, system control, model validation and many other applications. Those applications include, but are not limited to [46-59]:

- Developing algorithms to triangulate generator tripping location and predict trip amount in megawatts;
- An adaptive under-frequency load shedding algorithm using wide-area frequency as inputs;
- Real-time observation of system performance;
- Early detection of system problems;
- Information sharing and understanding global frequency dynamics;
- Observing and understanding power system oscillation, which will improve interarea oscillation modes calculation and prediction-based FNET measurement;

- Verifying system models and parameters used in simulations;
- Performing post-disturbance scenario reconstruction and tracking the sequence of events leading to an emergency;
- Understanding the fundamental characteristics and mathematics of the failure of complex systems;
- Providing wide-area information for improving power system control functions;
- Providing near real-time system status for analyzing the underlying causes of cascading events and system blackouts;
- Studying how frequency disturbances travel as electromechanical waves in power systems, i.e. “frequency wave” travel characterizes analysis;
- Developing algorithms to triangulate generator tripping location and predict trip amount in megawatts;
- An adaptive under-frequency load shedding algorithm using wide-area frequency as inputs;
- FACTS/ESS control and coordination in interarea low-frequency oscillation damping;
- ACE accuracy improvement with FNET data (our preliminary results show that ACE can be calculated with better precision with FNET data);
- Wide-area PSS control and coordination using FNET as inputs;
- Distributed generation (DG) control and coordination using regional FNET measurements;
- Validation of dynamic load models;
- Validation and refinement of planning models;
- Calibration and refinement of measurement facilities;
- Power system restoration;
- Data fusion between FDR, PMU and SCADA;
- Wide-area frequency data visualization and graphic user interface (FNET data streamer, dynamic map display);
- Refinement of planning, operation, and control processes essential to best use of transmission assets;
- Assessment of system behavior, comparisons against model results; and

- Performing system tests, disturbance analysis, and method development for grid reliability management.

Chapter 6 Conclusion

The post-event analysis and reports of recent large-scale blackouts revealed that most of the country's power grids are heavily loaded and operate at their maximum capacity, and the systems are very vulnerable to system-wide disturbances and cascading failures. In order to ensure system stability, disturbances should be detected and identified at an early stage, which leads to the right actions being taken quickly. The real-time, wide-area, and synchronized monitoring systems of interconnected power grids have proven effective and necessary for understanding system dynamics and identifying system-wide disturbances.

Frequency dynamics is one of the most important signals of a power system, and it is an indicator of imbalance between generation and load in an isolation system. The Internet based real-time GPS-synchronized wide-area Frequency Monitoring Network (FNET) was proposed to provide essential frequency dynamics information for a variety of system-wide monitoring, analysis and control applications. The implementation of FNET has, for the first time, made the synchronized observation of the entire US power network possible with very little cost.

The FNET is comprised of more than thirty Frequency Disturbance Recorders (FDR) geographically dispersed throughout the US, and an Information Management System (IMS) currently located at Virginia Tech. The FDR works as a sensor, which performs local measurements and transmits calculations of frequency, voltage magnitude and voltage angle to the remote servers through the Internet. Compared with its commercial counterpart, the Phasor Measurement Unit (PMU) invented at Virginia Tech, FDR is a much less expensive version of it. The special single phase algorithm in the FDRs made it possible to measure at 110V level, which is much more challenging than PMUs due to the noise involved at this level.

It has been a big challenge to design and implement the novel FDR to meet the requirements of a high-resolution, real-time and synchronized measurement. The Phasor

Angle Analysis algorithm has been adopted and improved to enable the high-resolution measurement of an FDR installed at residential voltage outlets (110V for a 60Hz system and 240V for a 50Hz system), instead of substation high-voltage inputs. The miniature, embedded 12-channel timing GPS receiver has been integrated to provide an accurate timing synchronization signal, UTC time stamp, and unit location. The Industrial Serial Device Server makes the FDR networked yet portable, which saves tremendous effort and cost for the installation and maintenance of the FDR. The firmware of the FDR is interrupt-driven and is optimized to meet real-time performance.

There are many issues that affect the FDR's accuracy and performance. These include A/D quantization errors, uneven sampling, amplitude and phase shift, harmonics, white noise, voltage swing, and missing points. An anti-aliasing filter, digital filters, coherent sampling, resampling and other solutions have been designed and proposed to solve or reduce the effects of those issues. The verification test results show that the frequency measurement accuracy of the FDR is ± 0.0005 Hz, and the time synchronization error is within ± 500 ns with appropriate GPS antenna installation. According to preliminary research results, the accuracy and performance of the FDR have proven to be satisfactory for most FNET applications, such as disturbance identification and event location triangulation.

Appendix A. Flowcharts of the FDR Firmware

This appendix provides some flowcharts of the FDR firmware design. The purpose of this document is to assist future designers in the further enhancement and development of the FDR.

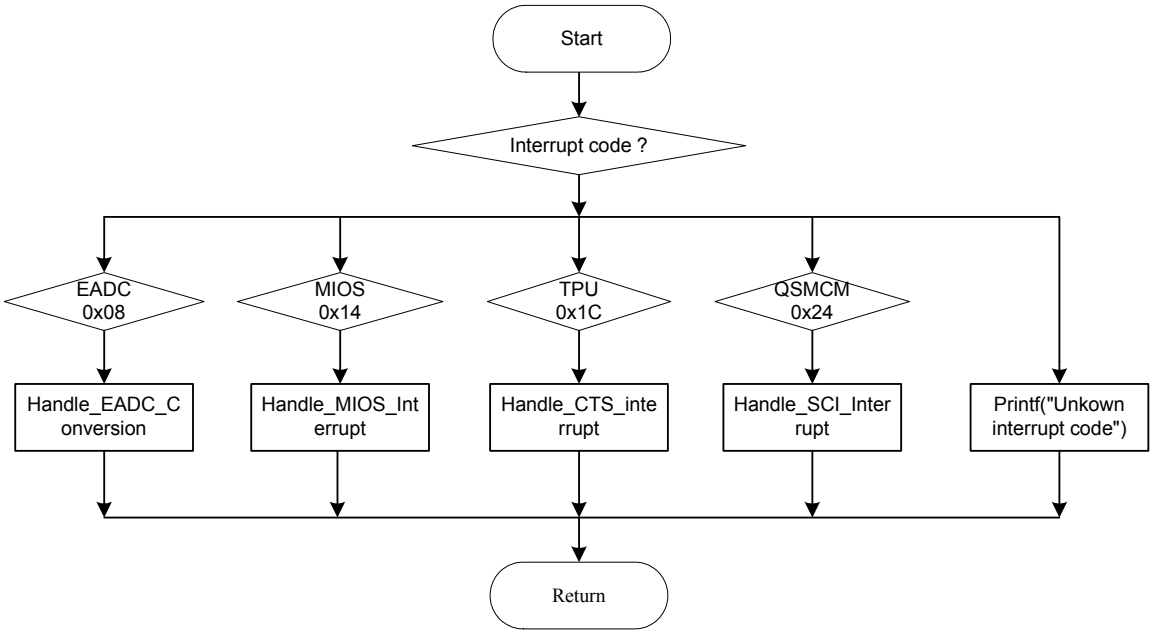


Figure A.1 Flowchart of the FDR interrupt handling

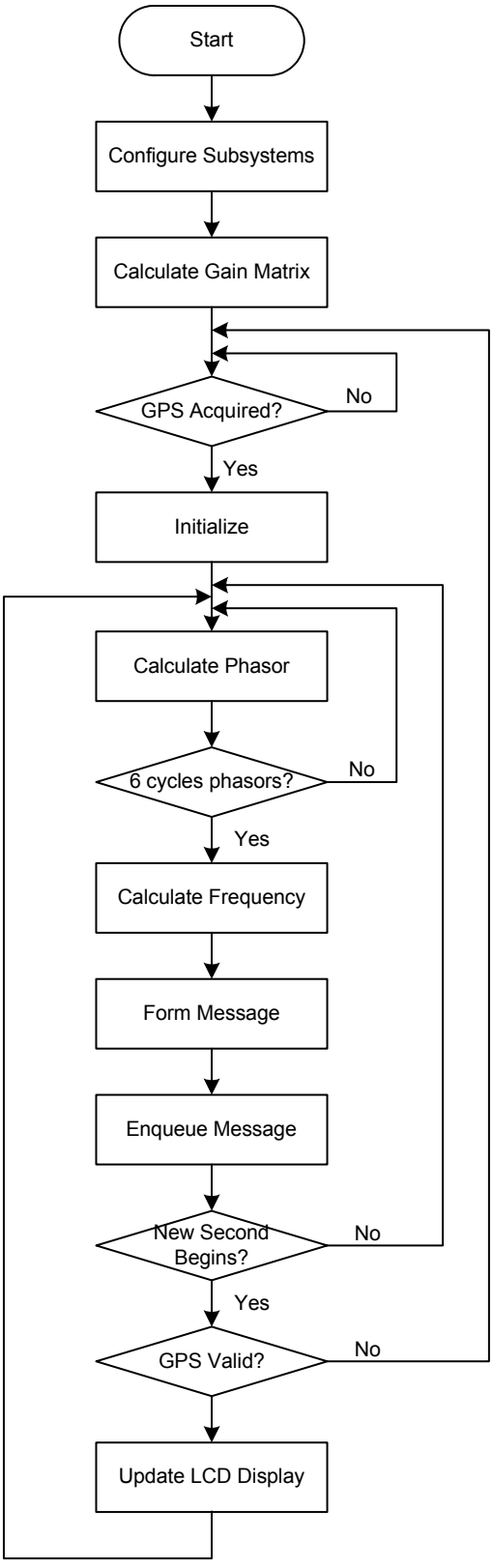


Figure A.2 Flowchart of the main function

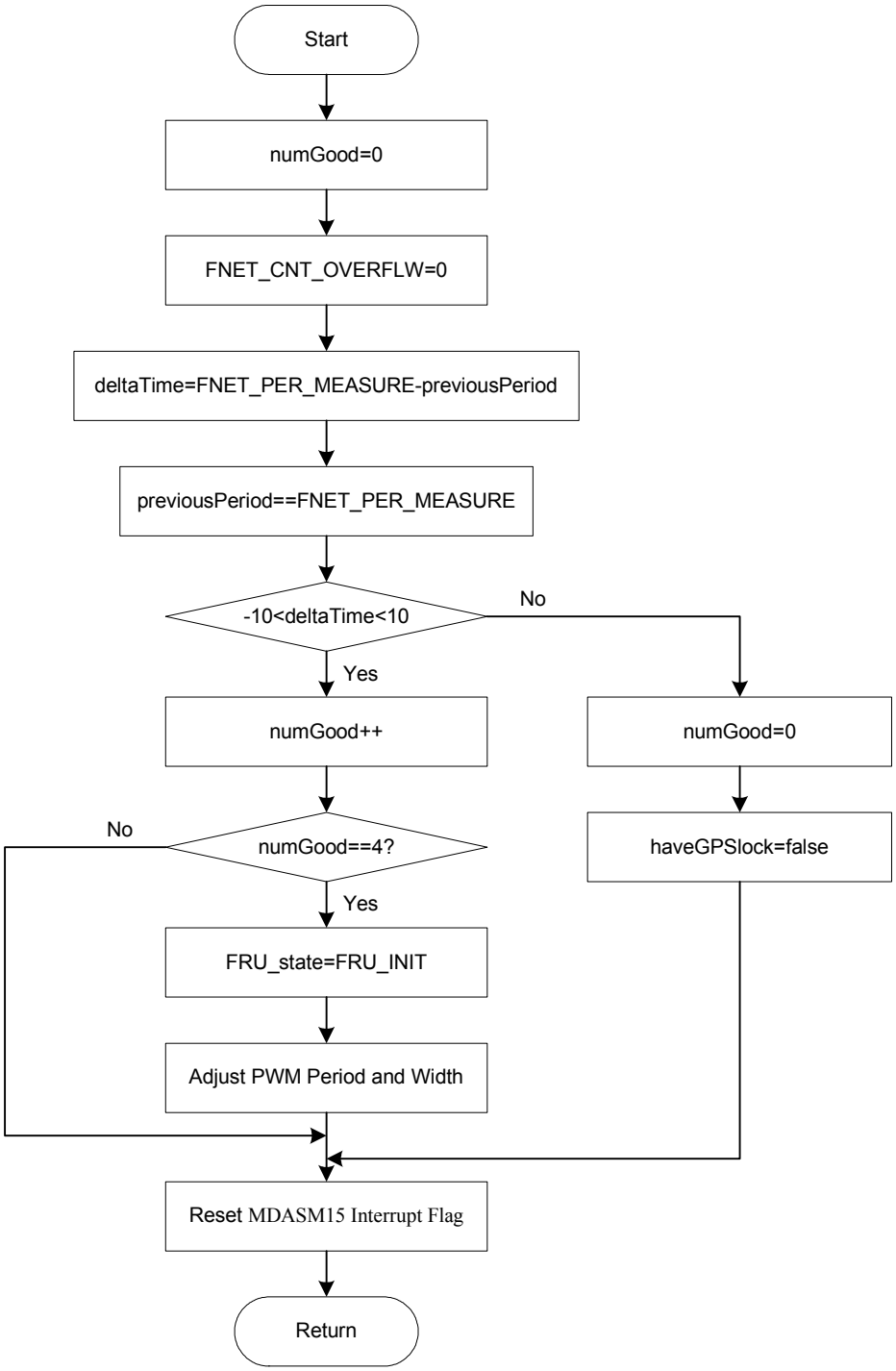


Figure A.3 Flowchart of the acquire mode

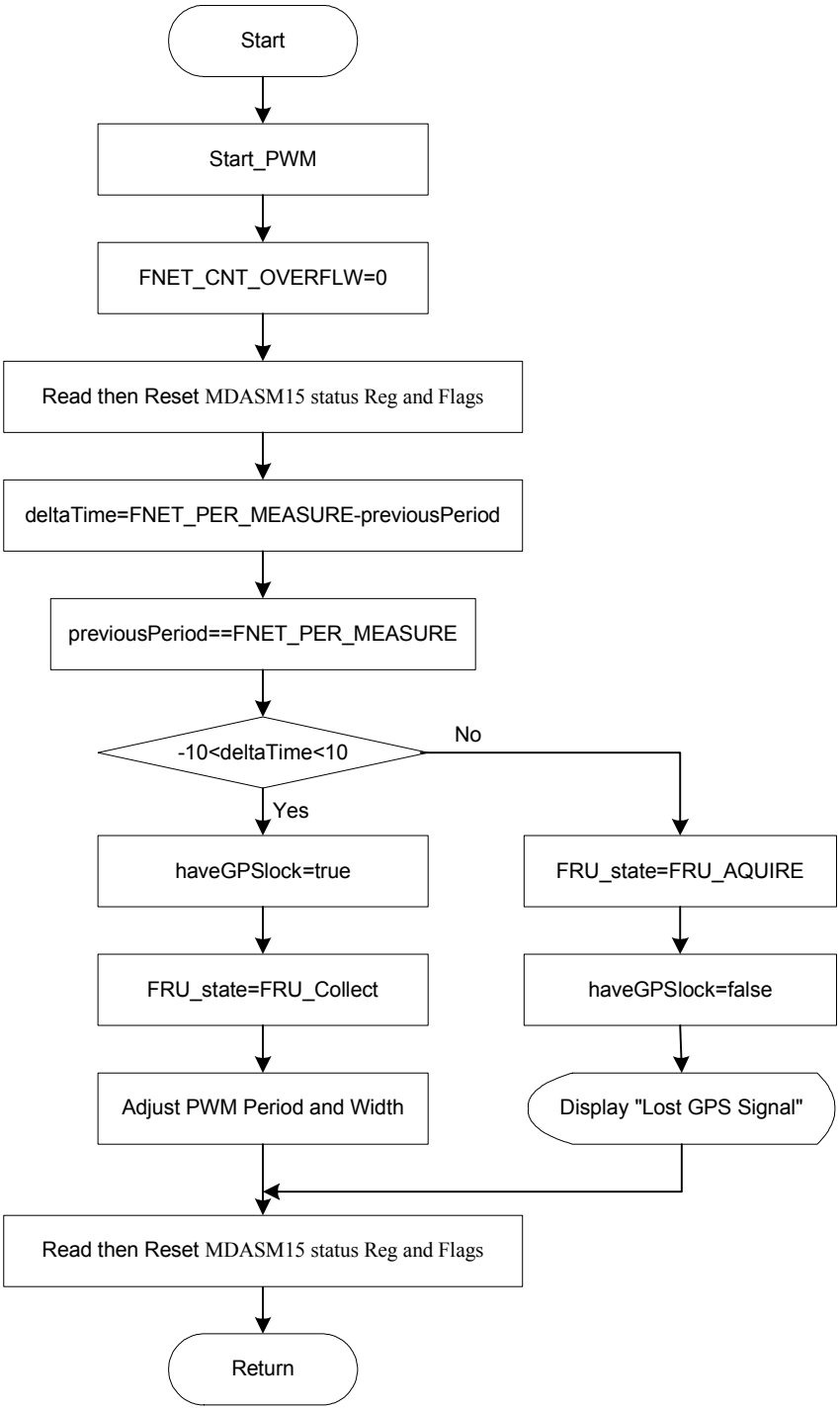


Figure A.4 Flowchart of the collect mode

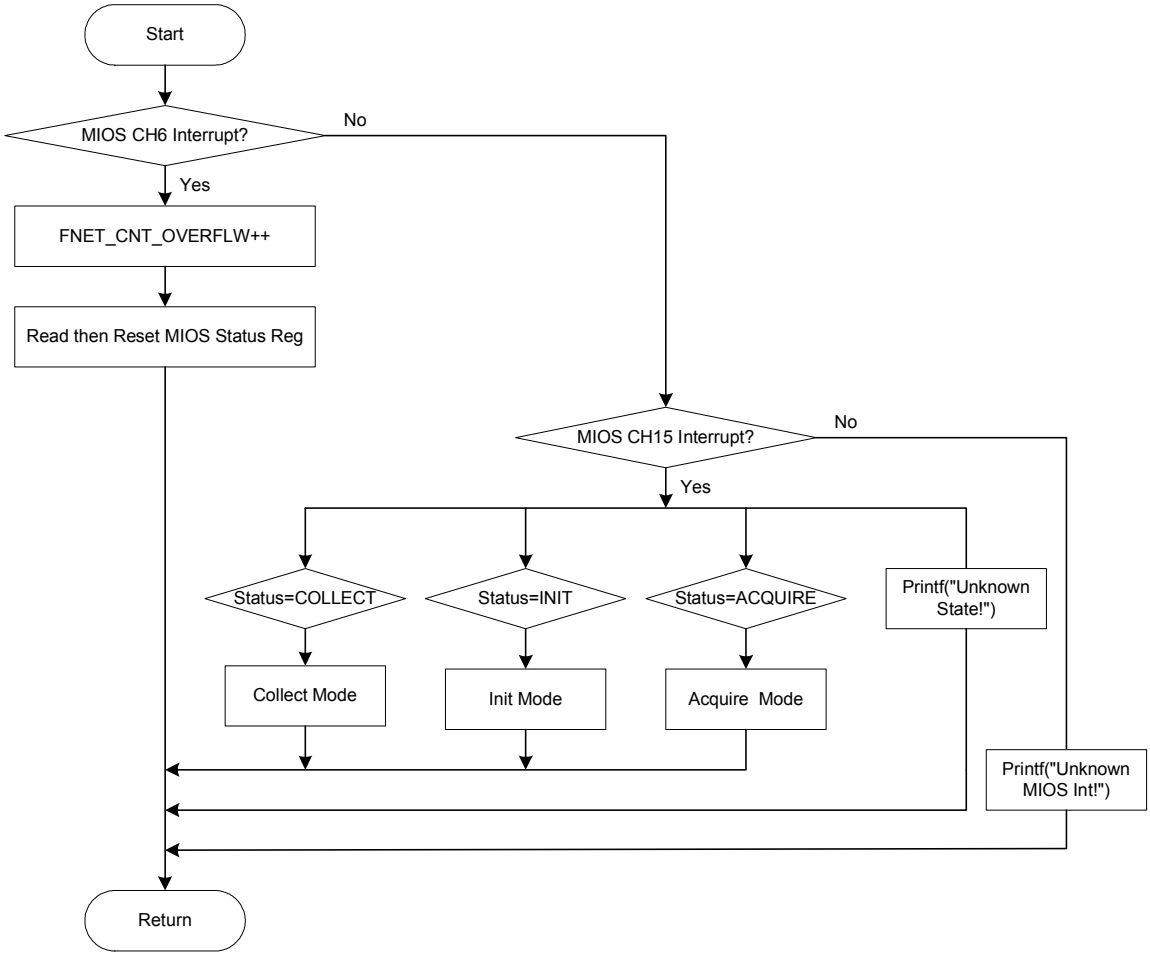


Figure A.5 Flowchart of the MIOS interrupt

Reference

[1] B. Qiu, L. Chen, V. A. Centeno, X. Dong, and Y. Liu, "Internet Based Frequency Monitoring Network (FNET)," *IEEE Power Engineering Society Winter Meeting*, vol. 3, pp. 1166-1171, 2001.

[2] U. S.-C. P. S. O. T. Force, "Cause of the August 14th Blackout in United States and Canada," Nov. 2003.

[3] F. Vandenberghe, "Lessons and Conclusions from the Lessons and Conclusions from the 28 September 2003 Blackout in Blackout in Italy," presented at IEA workshop.

[4] J.-Z. Yang and C.-W. Liu, "A new family of measurement technique for tracking voltage phasor, local system frequency, harmonics and DC offset," presented at 2000 Power Engineering Society Summer Meeting, Piscataway, NJ, USA., 2000.

[5] M. M. Begovic, P. M. Duric, S. Dunlop, and A. G. Phadke, "Frequency Tracking in Power Networks in the Presence of Harmonics," *IEEE Transaction of Power Deliver*, vol. Vol. 12, pp. 1064-1073, 1993.

[6] M. S. Sachdev and M. M. Giray, "A digital frequency and rate of change of frequency relay," *Transactions of the Engineering And Operating Division, Canadian Electrical Association*, vol. 17 part 3, pp. 78-sp-145, 1978.

[7] C. T. Nguyen and K. Srinivasan, "A new technique for rapid tracking of frequency deviation based on level crossings," *IEEE Transaction on Power Apparatus and Systems*, vol. PAS-103, pp. 2230-2237, 1984.

[8] L. Asnin, V. Backmutsky, and M. Gankin, "Comparative characteristics of main methods for dynamic estimation of frequency and magnitude parameters in power systems," presented at 22nd Convention of Electrical and Electronics Engineers in Israel, Piscataway, NJ, USA, 2002.

[9] A. A. Girgis and F. M. Ham, "A new FFT-based digital frequency relay for load shedding," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-101, pp. 433-439, 1982.

[10] A. G. Phadke, J. S. Trorp, and M. G. Adamiak, "A new measurement technique for tracking voltage phasors, local system frequency, and rate of change of frequency," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-102, pp. 1025-1038, 1983.

[11] J. Chen, "Accurate frequency estimation with phasor angles," in *ECE department*. Blacksburg: Virginia Tech, 1994.

[12] F. P. Dawson and L. Klaffke, "Frequency adaptive digital filter for synchronization signal aquisition and synchronized event triggering," presented at PESC 98 Record. 29th Annual IEEE Power Electronics Specialists Conference, New York, NY, USA, 1998.

[13] M. Wang and Y. Sun, "A practical, precise method for frequency tracking and phasor estimation," *IEEE Transactions on Power Delivery*, vol. 19, no.4, pp. 1547-52, Oct. 2004.

[14] J. Xi and J. Chicharo, "A new algorithm for improving the accuracy of periodic signal analysis," *IEEE Transactions on Instrumentation & Measurement*, vol. 45, no.4, pp. 827-31, Aug. 1996.
[15] M. S. Sachdev and M. M. Giray, "A least error square technique for determining power system frequency," *IEEE Transaction on Power Apparatus and Systems*, vol. PAS-104, pp. 437-444, 1985.
[16] M. M. Giray and M. S. Sachdev, "Off-nominal frequency measurements in electric power systems," *IEEE Transactions on Power Delivery*, vol. 4, pp. 1573-1578, 1989.
[17] S. A. Soliman, G. S. Christensen, D. H. Kelly, and N. Liu, "An algorithm for frequency relaying based on least absolute value approximations," vol. 19, no.2, pp. 73-84, Aug. 1990.
[18] A. A. Girgis and T. L. Hwang, "Optimal estimation of voltage phasors and frequency deviation using linear and non-linear Kalman filtering: theory and limitations," *IEEE Transaction on Power Apparatus and Systems*, vol. PAS-103, pp. 2943-2950, 1984.
[19] J.-Z. Yang and C.-W. Liu, "A New Family of Measurement Technique for Tracking Voltage Phasor, Local System Frequency, Harmonics and DC Offset," presented at Power Engineering Society Summer Meeting, 2000.
[20] J. Z. Yang and C. W. Liu, "A precise calculation of power system frequency," *IEEE Transactions on Power Delivery*, vol. 16, no.3, pp. 361-6, July 2001.
[21] A. Mariscotti, J. Z. Yang, and C. W. Liu, "Discussion of "A precise calculation of power system frequency and phasor" Closure to discussion"," *IEEE Transactions on Power Delivery*, vol. 15, no.4, pp. 1319-20, Oct. 2000.
[22] Jun-Zhe Yang and C.-W. Liu, "A precise calculation of power system frequency and phasor," *IEEE Transactions on Power Delivery*, vol. 15, no.2, pp. 494-9, April 2000.
[23] T. Lin, M. Tsuji, and E. Yamada, "A wavelet approach to real time estimation of power system frequency," presented at SICE 2001, Tokyo, Japan, 2001.
[24] P. K. Dash, D. P. Swain, A. Routray, and A. C. Leiw, "An Adaptive Neural Network Approach for the Estimation of Power System Frequency," *Electrical Power Systems Research*, vol. 41, no 3, pp. 203-210, June 1997.
[25] I. Kamwa and R. Grondin, "Fast adaptive schemes for tracking voltage phasor and local frequency in power transmission and distribution systems," *IEEE Transactions on Power Delivery*, vol. 7, no.2, pp. 789-95, April 1992.
[26] D. Hart, D. Novosel, Y. Hu, B. Smith, and M. Egolf, "A new frequency tracking and phasor estimation algorithm for generator," *IEEE Transactions on Power Delivery*, vol. Vol. 12, pp. pp1064-1073, 1997.
[27] EPRI, "Study of distribution system surge and harmonic characteristic," November 1980 1980.
[28] I. S. 1344-1995(R2001), "IEEE standard for synchrophasors for power systems," 2001.
[29] "IEEE standard for synchrophasors for power systems," 2001.
[30] A. G. Phadke, "Synchronized Phasor Measurement in Power Systems," *IEEE Computer Applications in Power*, vol. 6, pp. 10-15, 1993.
[31] W.-K. Chen, *Passive and active filters*: John Wiley & Sons, 1986.

[32] J. S. T. A. G. Phadke, *Computer relaying for power systems*: Research studies press ltd.
John Wiley and sons Inc., 1988.
[33] www.moxa.com.
[34] http://www.iana.org/assignments/port-numbers.
[35] "IEEE Standard Common Format for Transient Data Exchange (COMTRADE) for Power Systems," 1999.
[36] J. Dunlop, J. Fuchs, and S. Mihalik, "MPC555 Interrupts," in *Mototola Semiconductor Application Note*, July 2001.
[37] B. Lowe, *The Fundamentals of Mixed Signal Testing* June 2002.
[38] M. Sanaye-Pasand and J. AN, "Study, comparison and simulation of power system swing detection and prediction method," presented at IEEE Power Engineering Society General Meeting, Piscataway, NJ, USA, 2003.
[39] M. M. Begovic, P. M. Djuric, S. Dunlap, and P. AG., "Frequency tracking in power networks in the presence of harmonics," *IEEE Transactions on Power Delivery*, vol. 8, no.2, pp. 480-6, April 1993.
[40] L. Toivonen and J. Morsky, "Measurement and processing of distortion quantities in a portable, multi-purpose analyzer," *IEEE Transactions on Power Delivery*, vol. 8, no.4, pp. 1736-46, Oct. 1993.
[41] R. G. Lyons, *Understanding digital signal processing*: Addison Wesley Pub. Co., 1997.
[42] T. L. Baldwin, L. Mili, M. B. Boisen, and R. Adapa, "Power System Observability with Minimal Phasor Measurement Placement," *IEEE Transaction o f Power System*, vol. 8, no. 2, May 1993.
[43] www.nerc.com.
[44] K. Martin, "Time synchronization in electric power systems," 2000.
[45] B. Naduvathuparambil, M. C. Valenti, and A. Feliachi, "Communication Delays in Wide Area Measurement Systems," presented at Proceedings of the Thirty-Fourth Southeastern Symposium on, 2002.
[46] S. E. Stanton, C. Slivinsky, K. Martin, and J. Nordstorm, "Application of phasor measurements and partial energy analysis in stabilizing large disturbances," *IEEE Transaction o f Power System*, vol. 10, no. 1, pp. 297–306, Feb. 1995.
[47] http://www.wecc.biz/.
[48] C. W. Liu and J. Thorp, "Application of Synchronized Phasor Measurements to Real-Time Transient Stability Prediction," *IEE Proceedings. Generation, Transmission & Distribution*, vol. 142, no.4, pp. 355-60, July 1995.
[49] R. P. Schulz, L. S. VanSlyck, and S. H. Horowitz, "Applications of Fast Phasor Measurements on Utility Systems," presented at Conference Papers 1989 Power Industry Computer Application Conference, New York, NY, USA, 1989.
[50] R. O. Burnett, M. M. Butts, and P. S. Strelina, "Power system applications for phasor measurement units," *IEEE Computer Applications in Power*, vol. 7, no. 1, pp. 8-13, Jan. 1994.
[51] I. Kamwa, R. Grondin, and Y. Hebert, "Wide-area measurement based stabilizing control of large power systems-a decentralized/hierarchical approach," *IEEE Transaction ON Power Systems*, vol. 16. no. 1, pp. 136-53, Feb. 2001.

[52] B. Fardanesh, S. Zelingher, A. Sakis Meliopoulos, G. Cokkinides, and J. Ingleson, "Multifunctional synchronized measurement network," *IEEE Computer Applications in Power*, vol. 11, no.1, pp. 26-30, Jan. 1998.

[53] B. J. Rehtanz C, "Wide area measurement and protection system for emergency voltage stability control," presented at 2002 IEEE Power Engineering Society Winter Meeting, Piscataway, NJ, USA.

[54] R. Klump, R. E. Wilson, and K. E. Martin, "Visualizing real-time security threats using hybrid SCADA/PMU measurement displays," presented at Proc. 38th Ann. Hawaii Int. Conf. System Sciences, Jan. 2005.

[55] D. Karlsson, L. Broski, and S. Ganesan, "Maximizing power system stability through wide area protection," presented at 2004 57th Annual Conference on Protective Relay Engineers Conference, Piscataway, NJ, USA, 2004.

[56] 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," presented at 2004 IEEE Power Engineering Society General Meeting, Piscataway, NJ, USA, Jun. 6–10, 2004.

[57] B. Bhargava, G. D. Rodriguez, and A. Salazar, "Use of synchronized use of synchronized phasor measurement system phasor measurement system for enhancing for enhancing power system transmission reliability," in *Eastern Interconnect Phasor Project (EIPP) Meetings*. Columbus, Ohio: http://phasors.pnl.gov/eipp_meetings.html, Sep. 2004.

[58] A. Bykhovsky and J. H. Chow, "Power system disturbance identification from recorded dynamic data at the Northfield substation," *International Journal of Electrical Power & Energy Systems*, vol. 25, no.10, pp. 787-95, Nov. 2003.

[59] J. F. Hauer, "WAMS outreach project: sharing of knowledge & technology," presented at DOE Transmission Reliability Program Peer Review, January 27-29, 2004,.

[60] Y. J. Shin and A. C. Parsons, "Time-frequency analysis of power system disturbance signals for power quality," presented at IEEE Power Engineering Society Summer Meeting, July 1999.

[61] T. Hashiguchi and M. Yoshimoto, "Oscillation mode analysis in power systems based on data acquired by distributed phasor measurement units," presented at Proceedings of the 2003 International Symposium on Circuits and Systems, 25-28, May 2003.

[62] A. Bykhovsky, "Disturbance analysis using power system measured data " in *Electric Power Engineering*, vol. Doctor of Philosophy. Troy, New York: Rensselaer Polytechnic Institute, Sep. 2001.

[63] R. P. Schulz, "Modeling of Governing Response in the Eastern Interconnection," presented at Proceedings of IEEE Winter Power Meeting, 1999.

[64] R. P. Schulz and B. B. Laios, "Triggering tradeoffs for recording dynamics," *IEEE Computer Applications in Power*, vol. 10, No. 2, pp. 44-49, April 1997.

[65] K. Mei and S. M. Rovnyak, "Response-based decision trees to trigger one-shot stabilizing control," *IEEE Trans. on Power Systems*, vol. 19, No. 1, pp. 531-537, Feb. 2004.

[66] A. Bykhovsky and J. H. Chow, "Dynamic data recording in the new england power system and an event analyzer for the northfield monitor," presented at VII SEPOPE, Curitiba, Brazil, May 2000.

[67] R. P. Schulz and W. W. Price, "Classification and Identification of Power System Emergencies," *IEEE Transaction on power apparatus and systems*, vol. 103, No. 12, pp. 3471-3479, December 1984.

[68] R. J. Murphy, "Disturbance recorders trigger detection and protection," *IEEE Computer Applications in Power*, vol. 9, no.1, pp. 24-8, Jan. 1996.

[69] C. S. Hyung, G. L. Jang, and I. J. Gyu, "Mobile station location using hybrid GPS and a wireless network," presented at 57th IEEE Semiannual Vehicular Technology Conference, Piscataway, NJ, USA, 2003.

[70] Y. Ying and H. F. Silverman, "An improved TDOA-based location estimation algorithm for large aperture microphone arrays," presented at 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, Piscataway, NJ, USA, 2004.

[71] J. Bard and F. Ham, "Time difference of arrival dilution of precision and applications," *IEEE Transactions on Signal Processing*, vol. 47, no.2, pp. 521-23, Feb. 1999.

[72] O. Jahromi and P. Aarabi, "Theory and design of multirate sensor arrays," *IEEE Transactions on Signal Processing*, vol. 53, no.5, pp. 1739-53, May 2005.

[73] J. S. Thorp, C. E. Seyler, and A. G. Phadke, "Electromechanical wave propagation in large electric power systems," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 45, pp. 614-622, June 1998.

[74] L. Huang and A. G. Phadke, "Electromechanical wave propagation in power systems with boundary conditions," presented at Proceedings of the IASTED International Conference Power and Energy Systems, Anaheim, CA, USA., 2001.

[75] M. Parashar, J. S. Thorp, and C. E. Seyler, "Continuum modeling of electromechanical dynamics in large-scale power systems," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 51, pp. 1848 - 1858, Sept. 2004.

[76] S. S. Tsai, L. Zhang, A. G. Phadke, and Y. Liu, "Study of global frequency dynamic behavior of large power systems," presented at 2004 IEEE PES Power Systems Conference & Exposition, Piscataway, NJ, USA, 2004.

Vita

Chunchun Xu was born in Nantong, Jiangsu Province, China on February 4th, 1977. She received her Bachelor and Master degree in Electrical Engineering from Nanjing University of Aeronautics & Astronautics, in 1999 and 2002, respectively. In 2002, she joined in the Interdisciplinary Microsystems Group (IMG) at University of Florid. As a transfer student, she started to pursue her PhD degree at Virginia Tech in 2003. During her graduate study, she participated in the cooperative education program with General Electric, gaining valuable industrial work experience. Her research interests include real-time system, embedded system, distributed control system, model-base design, hardware-in-the-loop simulation, algorithm implementation, data acquisition and analysis, and digital signal processing.