ABSTRACT

I/Q IMBALANCE COMPENSATION FOR WIDEBAND ELECTRONIC INTELLIGENT RECEIVERS

by Vincent Christopher Mancuso

An Electronic Warfare (EW) receiver is part of an electronic intelligent (ELINT) receiver. It is under intensive development to adapt to the ever-evolving complex signal space. EW receivers are required to detect and characterize multiple incoming signals in real time and the receivers are designed without any prior information of possible incoming signals. It is therefore desired to have a wide working bandwidth. There are two parts in this thesis: Bandwidth improvement in EW receiver development and the creation of an encoder to detect unknown signals. One scheme to improve the wideband capacity is through the implementation of in-phase and quadrature (I/Q) module. With this implementation, the working bandwidth doubles. However, the low instantaneous dynamic range due to imbalance I/Q module prevents this from being useful. An algorithm to mitigate the imbalance effect was recently developed. Part of this thesis is to study the mitigation algorithm and implement it for real time processing using a field-programmable gate array (FPGA). The other part of this thesis is to create an encoder for multiple signal detection. Three different radar signal types, continuous-wave (CW), linear chirp and binary phase shift keying (BPSK) are considered in this study.

I/Q IMBALANCE COMPENSATION FOR WIDEBAND ELECTRONIC INTELLIGENT RECEIVERS

A Thesis

Submitted to the

Faculty of Miami University

In partial fulfillment of the requirements for the degree of

Master of Science in Computational Science and Engineering

Department of Electrical & Computer Engineering

by

Vincent Christopher Mancuso

Miami University

Oxford, Ohio

2013

Advisor_____

Dr. Chi-Hao Cheng

Reader _____

Dr. Donald Ucci

Reader_____

Dr. Lihyeh Liou

Reader_____

Dr. Qihou Zhou

Table of contents

List of Tables

List of Figures

Acknowledgments

This thesis would not have been possible without the help of a number of people. First, I would like to thank my advisor, Dr. Chi-Hao Cheng; he has given me an exuberant amount of help throughout the last two years and without him, this work would have not been possible. Without all of the meetings, discussions, and various lessons, I would not have had the success I had with this thesis. There were multiple times when my work seemed to hit a dead end and Dr. Cheng did all in his power to guide me to find my way out of these troubles. He not only provided me with the knowledge for completing this work, but he also gave me the skill set to deal with any possible problems on my own.

Second, I would like to thank the Dayton Area Graduate Studies Institute (DAGSI) for providing me with the means to further my ability to complete this work. Specifically, the members of the Air Force Research Laboratory (AFRL) at Wright Patterson Air Force base were extremely important. The members who were of particular help were Dr, Lihyeh Liou, Thomas Pemberton, Peter Buxa, and Matthew Longbrake; whenever I had issues with theory or implementation of this work, they did not hesitate to help. Everyone went beyond what was expected to make this work possible; they spent valuable time teaching me the knowledge necessary to complete this work.

Third, I would like to thank my committee members: Dr. Lihyeh Liou, Dr. Donald Ucci, and Dr. Qihou Zhou. Their insight and knowledge provided me with the ability to not only further the quality of my thesis, but to succeed with any endeavors I may encounter later in my career.

Finally, I would like to thank my parents, Joe and Diana Mancuso. They encouraged me to push through all adversity during the duration of this work and I would not have been able to complete this work without their relentless and loving support.

Chapter 1

Introduction

1.1 Background

Electronic warfare (EW) receivers are designed to detect an unknown number of radar signals. There is a requirement for real-time data acquisition and processing with very precise characterization of the signal. There have been many advances in receiver applications such as in the field of communications, which has many similarities with that of EW receivers. However, unlike communication receivers, the EW receiver has no prior knowledge of the incoming signal(s) that needs to be detected and the signal may also have a very short duration, furthering the difficulty to detect it. This presents a problem to EW receiver designers due to the necessity to have accurate and prompt signal detection(s) and estimation(s) over a broad bandwidth; these receivers are used in different military aircraft and, in most situations, a precise detection is vital [1].

The EW receiver has two main, real-time requirements: 1) to detect multiple signals and 2) precisely estimate the signal's characteristics. To develop a receiver that meets these two requirements, different techniques are needed to bypass the natural problems and hardware limitations associated with receiver design; this includes introducing additional hardware such as an In-phase/Quadrature (I/Q) based channel receiver and several digital signal processing techniques such as the application of a windowing function and frequency and power interpolation. Depending on certain choices, the receiver may be designed to detect weak signals, strong signals, or both. However, there is a trade off when it comes to this choice; detecting strong signals will diminish the ability to detect weaker signals simultaneously and vice-versa. This leads to a problem when taking into account that there is no prior knowledge of possible incoming signals. Due to the signal frequency being unknown, the working bandwidth of the EW receiver needs to be as broad as possible [2]. For a typical communication receiver, the designer will know the signal characteristics such as signal frequency, modulation schemes,

etc., for which the receiver will be receiving, which gives the designer certain guidance when choosing specifications. However, an EW receiver designer needs to create a design that will allow the receiver to detect a wide variety of signal(s) over a broad bandwidth [1, 2].

1.2 Goals

The purpose of this thesis is to develop a receiver that will accurately detect and characterize three signal types: Continuous wave (CW), chirping, and Binary Phase Shift Keying (BPSK) signals. This receiver will also incorporate an I/Q based channel to double the working bandwidth to further strengthen the detection capabilities. Once this portion of the receiver is simulated using MATLAB®, it will be implemented for real-time use in hardware using a Field-Programmable Gate Array (FPGA).

1.3 Overview

This thesis will discuss the methodology of creating a working I/Q based channel EW receiver. Chapter 2 will be a review of the different digital signal processing techniques that are used to design the receiver such as the Discrete Fourier Transform (DFT), windowing functions, and frequency and power interpolation. Chapter 3 details the EW receiver, including the different hardware associated with the receiver, design requirements, and signals that this thesis will consider. Chapter 4 will show the performance of the EW receiver and its effectiveness on detecting all three of the proposed signal types. Chapter 5 focuses on the I/Q based channel receiver, the problems that accompany this technology, and the techniques to alleviate these problems. Chapter 6 discusses the hardware implementation of a compensator to improve the IQ receiver performance and its results. Finally, chapter 7 concludes this thesis and proposes possible future work.

Chapter 2

Digital Signal Processing Techniques

2.1 Discrete Fourier Transform

The first step of the digital EW receiver is to gather any incoming, analog data and transform it into the digital domain. This process is conducted with an Analog-to-Digital Converter (ADC). Once the signal passes through an ADC and is in its digital representation, the DFT is used to convert the incoming signal from the time-domain to the frequency domain. Assuming the incoming signal, *x(t)*, is a CW signal, it can be represented as:

$$x(t) = \cos(2\pi f t), \tag{2.1}$$

where f is the signal's frequency.

The analog signal is sampled at discrete times using an ADC. Due to its physical limitations, the ADC has a finite sampling frequency, which limits the working bandwidth of the receiver [3]. The sampled signal of *x(t)* can be represented below:

$$x[n] = \cos(2\pi f n / f_s), \tag{2.2}$$

where f_s is the sampling frequency and *n* is an integer representing the sampled points of the signal. Finally, to convert this sampled signal to the frequency domain, the DFT is computed based on a segment of x[n] following the equation below [3]:

$$X\left(e^{jw}\right) = \sum_{n=0}^{N-1} x[n] e^{-j2\pi f n}. \tag{2.3}$$

where *N* is the number of the data point in one frame. The computational complexity of

performing the DFT on a set of data in real-time is on the order of N^2, i.e., $O(N^2)$; when N is large, the computation load may be too great for a real-time EW receiver. To combat this issue, the Fast-Fourier Transform (FFT) is used to calculate the DFT. The computational complexity of the FFT is $O(nlog(n))$, which is more suitable for real-time processing [3]. However, even with the faster computational time, it is still impractical to calculate a long FFT in a short period of time because of hardware and speed limitations. This is an issue which will be discussed in Chapter 6.

When converting the incoming signal into its frequency domain, the FFT is only able to output correct frequencies that are a multiple of the frequency resolution. The frequency resolution defined as follows:

$$f_{res} = f_s/N, \tag{2.4}$$

where f_s is the sampling frequency and N is the frame size of the FFT computation. The inability to have perfect frequency resolution (i.e., $f_{res} \sim 0$) creates two problems, namely, 1) inaccurate frequency and power estimation and 2) the phenomena of spectral leakage, which limits EW receiver's capability of detecting strong and weak signals simultaneously.

2.2 Frequency and Power Interpolation

Due to the imperfect frequency resolution mentioned in the previous section, the FFT might not output the correct frequency and power. Figure 2-1 visualizes this problem; when conducting the FFT upon a sinusoidal signal whose frequency is not multiple of the f_{res}, the FFT output (produced in blue) shows the error in frequency and power in comparison to the actual signal frequency and power (shown in red). This apparent minimal separation is actually substantial because of the necessity to have very precise signal estimation and characterization to identify the incoming signal. To help minimize the separation between the location of the FFT output and the actual location, frequency and power interpolation is used. This process works as follows with the utilization of the Blackman window: 1) find the peak of the signal's FFT output, 2) examine the two adjacent points on each side of the peak and choose the maximum of the two

adjacent points (this establishes to which side of the FFT output the estimated frequency will shift), 3) take the ratio of the peak and the point found in Step 2, and 4) use this value in two separate equations for frequency and power interpolation. It is worthy of notice that Blackman window is applied before the FFT calculations to reduce spectral leakage which will be covered in the next section.

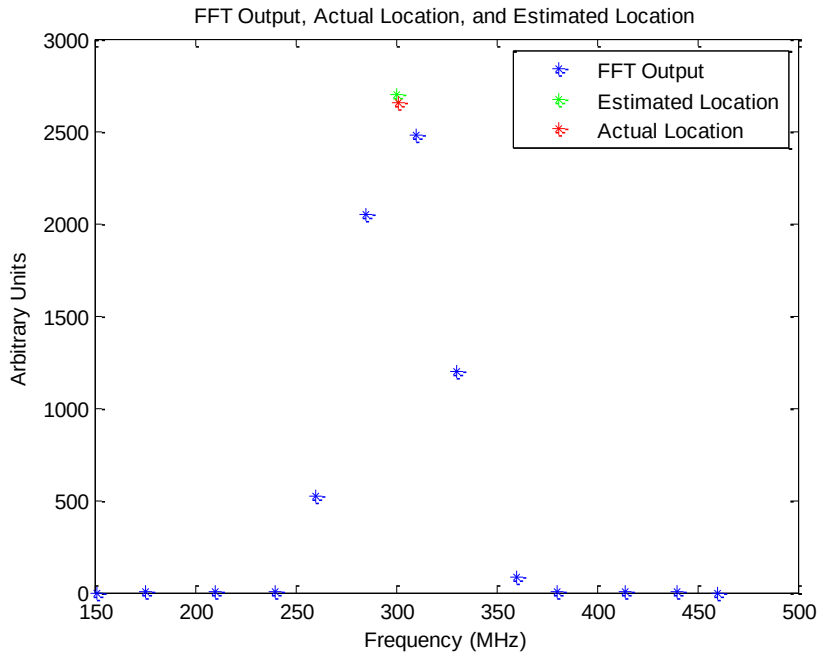
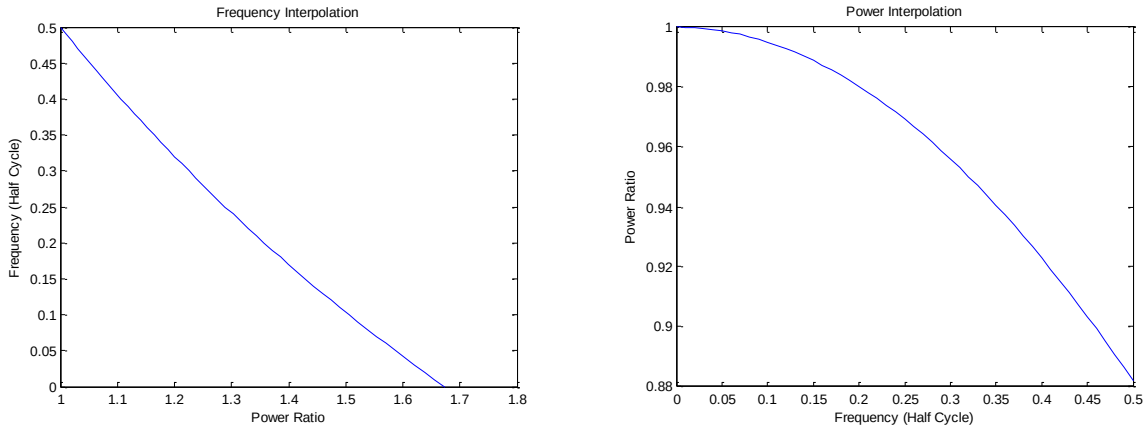


Figure 2-1: The FFT output peak of a signal (solid line) vs. the actual location of the peak

Figure 2-2 shows the graphs used to determine the power and frequency interpolation equations to obtain a more precise frequency and power estimation; these graphs and the power and frequency interpolation equations, Eqs. (2.5) and (2.6), were obtained by the following a series of steps: The graph in Fig. 2-2 (a) was created by plotting the ratio of the FFT peak and its maximum adjacent signal vs. the actual signal frequency. The signal frequency was swept from $k \times f_{res}$ to $(k+1/2) \times f_{res,}$ where k is an integer and f_{res} is defined in Eq. (2.4). Figure 2-2 (b) was created by plotting the ratio of the FFT peak power of the different frequencies within the same range as Fig. 2-2 (a) and the FFT peak power when a signal frequency was right on FFT bin frequency.



(a) (b)

Figure 2-2: (a) Frequency interpolation and (b) power interpolation for a more accurate estimation

From these graphs, using the curve fitting tool in MATLAB®, two equations were yielded that were used to give a more precise estimate for frequency and power. They are:

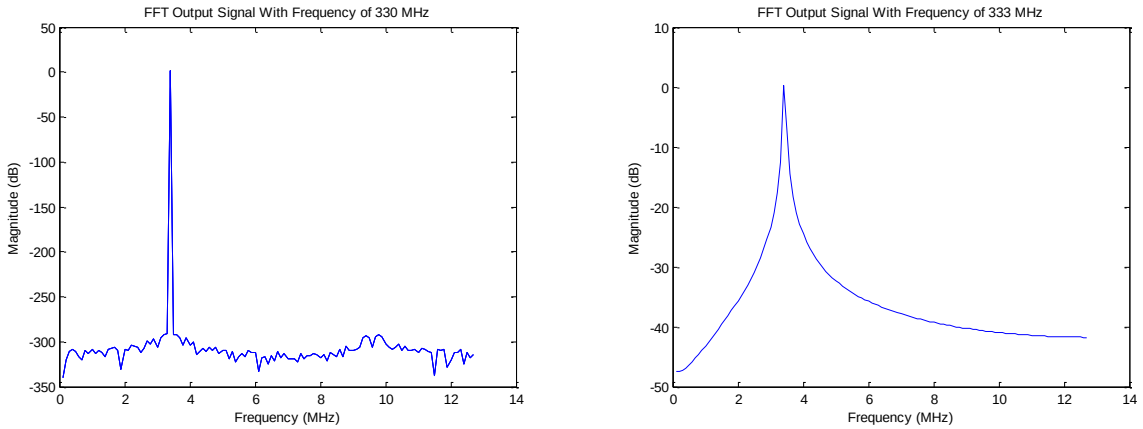
$$f_{est} = 0.31146x^2 \ -1.5696\text{x} + 1.7562, \tag{2.5}$$

$$P_{est} = \ -0.45288f_{est}^2 - 0.011894 + 1.0005. \tag{2.6}$$

The variable x is the ratio of the peak and largest adjacent peak (step 3). Once a more accurate frequency estimation, f_{est}, is found, that value is then used in Eq. (2.6) to get a better power estimation. This finally yields the more precise frequency and power estimates, visualized by the green marker in Fig. 2-1.

2.3 Spectral Leakage

Spectral leakage occurs when the signal's frequency is not a multiple of the frequency resolution. The outcome of this undesirable effect results in the FFT output spreading its energy and leaking it into surrounding frequency bins [4] as shown in Fig. 2-3.



(a) (b)

Figure 2-3: Spectrum of (a) a signal with frequency that is a multiple of the frequency resolution and (b) a signal that is not a multiple.

As seen in Fig. 2-3(a), when the frequency of the signal is a multiple of the frequency resolution, the surrounding bins have a floor of around -300 dB. However, as seen in Fig. 2-3(b), when the frequency of the signal is not a multiple of the frequency resolution, the energy from the signal spreads into the neighboring bins and has a floor of around -30dB. This spreading of energy has the possibility of masking signals with a smaller magnitude that may be located in or near the flooded bins and may cause the receiver to miss these signals. To counter this phenomenon, windowing functions are used to reduce the leaked energy [4].

There are a few possible windowing functions that could be used and each has its own key characteristics [4]. These characteristics include two main features: 1) the width of the main lobe and 2) the height of its side lobes; the width of the main lobe affects how closely the receiver can differentiate between signals that are in neighboring frequency bins, while the height of the side lobes lessens the effects of spectral leakage by lowering the energy floor. Since we need to develop an EW receiver that can detect signals with a 50 dB power difference, the Blackman window is chosen from other options because it successfully lowers the floor to an acceptable level (i.e., –58 dB as shown in Fig. 2-4 (b)) [4]. Figures 2-4 (a) and (b) display the time and frequency domain of a Blackman window, respectively.

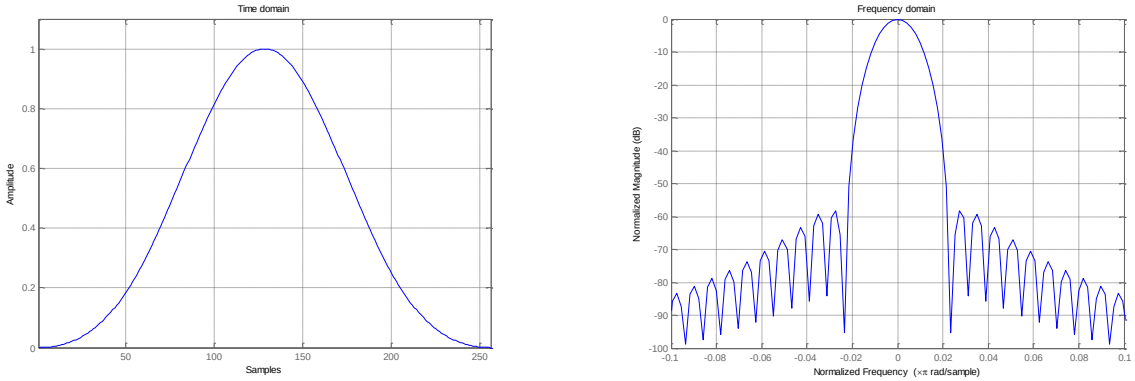


Figure 2-4: (a) Time domain representation of a Blackman Window and (b) the frequency domain representation of the Blackman window.

In Fig. 2-5, the Blackman window is applied to the same signal shown in Fig. 2.3 (b) that caused the energy to spread into adjacent frequency bins. There is a noticeable widening of the main lobe of the signal; however, the important outcome of the windowing application is to significantly reduce spreading energy. This is visualized in Fig. 2-5 by lowering the energy from around –30 dB to below –60 dB.

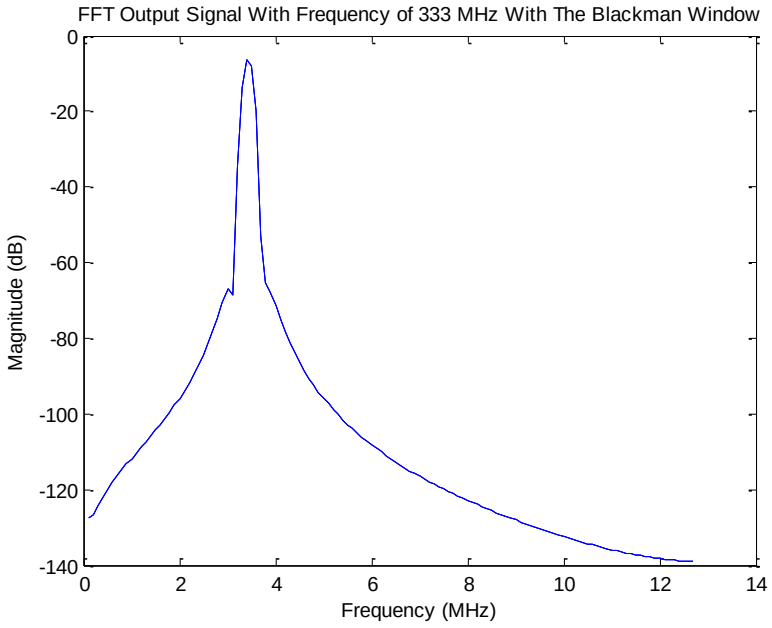


Figure 2-5: The signal with spread energy with a Blackman window applied.

Chapter 3

Electronic Warfare Receiver Introduction

3.1 EW receivers

An EW system consists of a receiver, processor, and countermeasure generator, as visualized in Fig. 3-1. The receiver gathers any incoming signals and produces a Pulse-Descriptor Word (PDW) consisting of the signal's characteristics. Then, the processor takes this PDW and attempts to match the signal characteristics with known signal types that are stored in a database. If any matches are discovered, the technique generator will follow with an appropriate action [1].

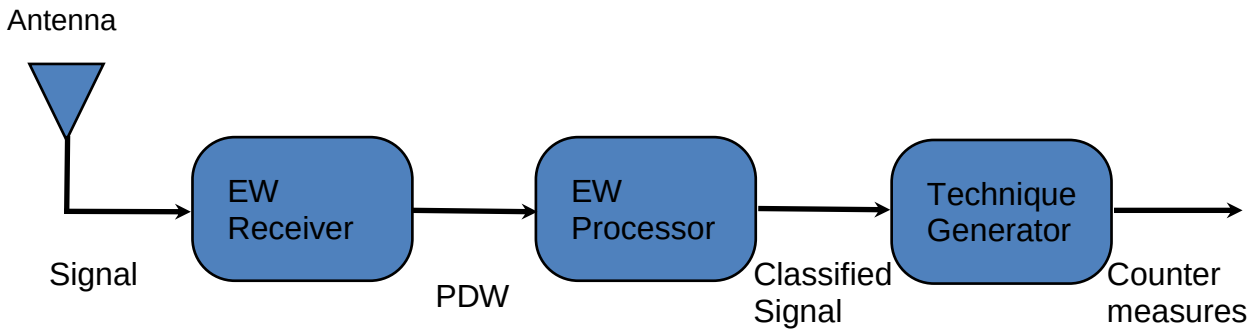


Figure 3-1: Hardware layout of a typical full EW system

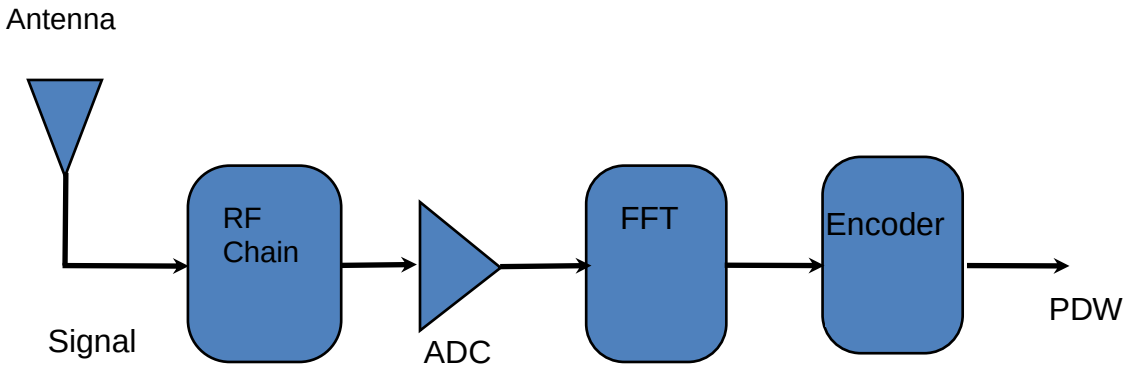


Figure 3-2: Hardware layout of a typical EW receiver

The focus of thesis is the EW receiver illustrated in Fig. 3-2. The main function of EW receiver is to produce the PDW containing the signal's characteristics for further evaluation. Figure 3-2 shows that the antenna continuously receives signal and passes it through the ADC, the information is converted into the frequency domain using the FFT. The working bandwidth of an EW receiver is limited by the ADC's sampling rate and determined by the Nyquist theorem. The encoder is the portion of the receiver that analyzes the FFT output and produces the PDW; the PDW then provides five important characteristics: power, frequency, Time of Arrival (TOA), Pulse Width (PW), and Angle of Arrival (AOA). To detect AOA, the receiver needs multiple antennas [1]. In this study, only a single antenna system is considered, so the AOA is not included in this PDW. It is worthy of notice that the FFT-based EW receiver introduced here is widely used and can be considered as the standard for EW receiver implementation [5-8].

Along with the PDW, there are other important requirements that the receiver must meet. One important EW receiver requirement is to have a very broad working bandwidth. Since the EW receiver is designed with no prior knowledge of possible signal, having a limited bandwidth greatly reduces the effectiveness of its signal detection capabilities. It would be ideal to have a receiver that could cover a range from 2 to 18 GHz [2]. However, this range would be difficult to achieve with standard EW receiver technology. Based on Nyquist theory, assuming the ADC sampling rate is f_s, the receiver's working bandwidth is at most $f_s/2$. One method to improve the situation is to incorporate an I/Q based channel receiver. This has the ability of doubling the working bandwidth without increasing sampling rate. The I/Q based channel receiver will be discussed in Chapter 6.

Another requirement is the ability to detect multiple signals. This adds challenges to EW encoder designers because there is no prior knowledge of incoming signals or of the type of the any captured signals. The receiver should be able to recognize multiple signal types. For this research, we focus on three typical radar signals: CW, chirping and BPSK signals.

3.2 False Alarm and Threshold

Due to interference from noise, other signals, and quantization errors, there is a possibility of the receiver providing the characteristics of a nonexistent signal. The probability of this happening is referred to as the False Alarm Rate (FAR) [9]. A false alarm may result in unnecessary counter measure actions, and it is obvious that the FAR should be kept at a low level. [9].

The threshold is the smallest power limit at which the receiver will consider an FFT peak to be a possible signal. In the EW receiver, the threshold is determined based on a given FAR. To determine the threshold, a study was conducted on the amplitude of a FFT output with the input as just a real-valued thermal noise. The noise has a Gaussian distribution whose power is determined based on the receiver's working bandwidth [9]. The first step is to store the different amplitudes (considered as a random variable) of one selected FFT bin from the real random noise from 1,000 trials. The probability density function of the amplitude can be modeled as a Rayleigh distribution, as shown in Fig. 3-3. This is given as:

$$p(r) = \frac{r}{\sigma^2} e^{\frac{-r^2}{2\sigma^2}}, \tag{3.1}$$

where r is the amplitude of the random variable and σ can be derived either from the random number's mean or variance using the following equations [9]:

$$\sigma = \frac{m}{\sqrt{\frac{\pi}{2}}} \tag{3.2}$$

and

$$\sigma = \sqrt{\frac{2v}{4-\pi}}, \tag{3.3}$$

where m is the mean and v is the variance of the amplitude of the chosen frequency bin. Once σ is calculated, the threshold can be determined based on the desired FAR and σ.

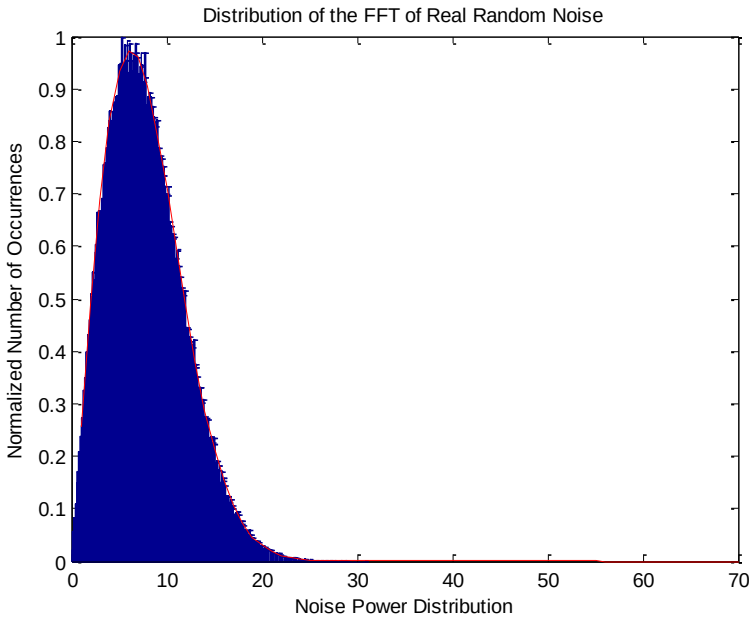


Figure 3-3: The amplitude distribution of real, random noise in the frequency domain following a Rayleigh distribution.

Threshold determination is a function of the noise distribution and the FAR. Though it would be ideal to choose a high threshold to guarantee the lowest possible FAR, it would have negative consequences if not chosen correctly [9]. A high threshold will prevent the receiver from detecting weak signals. A low threshold will improve detecting weak signals, but it will deteriorate the FAR performance.

3.3 Sensitivity and Instantaneous Dynamic Range

Instantaneous dynamic range is defined as the power ratio (in dB) between the signals of the highest and the lowest power that a receiver would be able to detect simultaneously (visualized in the graph of Fig. 3-4 (a)). Sensitivity is defined as the lowest possible signal that the receiver would be able to detect with a 90% probability of detection (visualized in the graph of Fig. 3-4 (b)) [9]. It would be ideal to have a large instantaneous dynamic range and high sensitivity. However, these two values have tradeoffs; if a large instantaneous dynamic range is chosen; high powered signals would be able to be detected. However, high powered signals would raise the noise floor, leading to the possibility of masking lower powered signals and harming the

sensitivity of the receiver. Moreover, if the receiver is designed to be highly sensitive, it would diminish the ability to detect high powered signals, for the same reason as stated above [9].

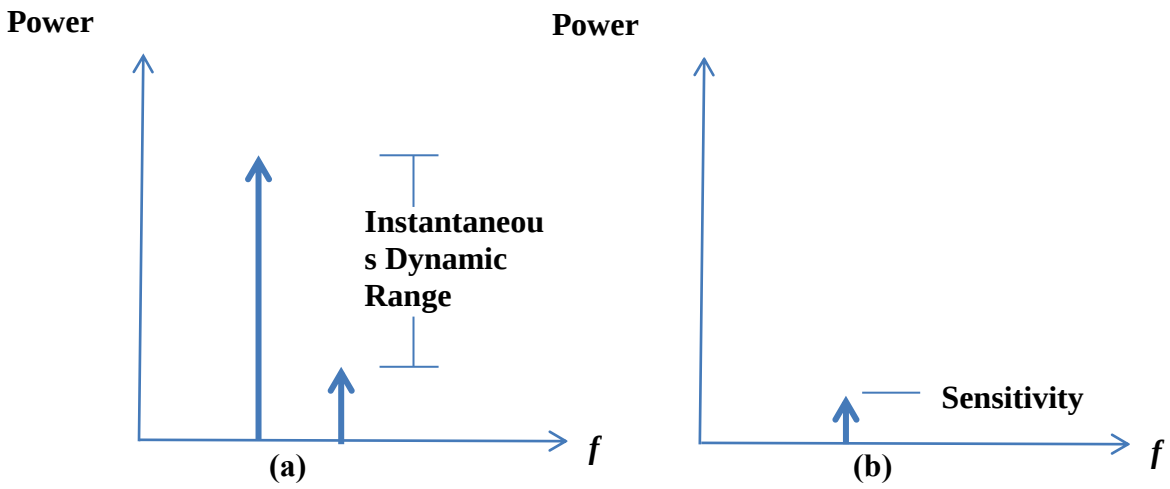


Figure 3-4: (a) Illustration of the instantaneous dynamic range and (b) illustration of the sensitivity of a receiver.

3.4 Signals of Consideration

The three radar signals considered in this study are CW, chirping, and BPSK signals. The EW receiver's PDW output might need to provide additional characteristics than the standard five discussed in Section 3.1, depending on the signal type. Since the EW receiver processes signals frame-by-frame without any synchronization between receiver and transmitter, it is more difficult to generate the PDW for some signal types more so than other types of signals. The three different signals with their respective PDW output requirements and encoder challenges are as follows:

1) Continuous Wave

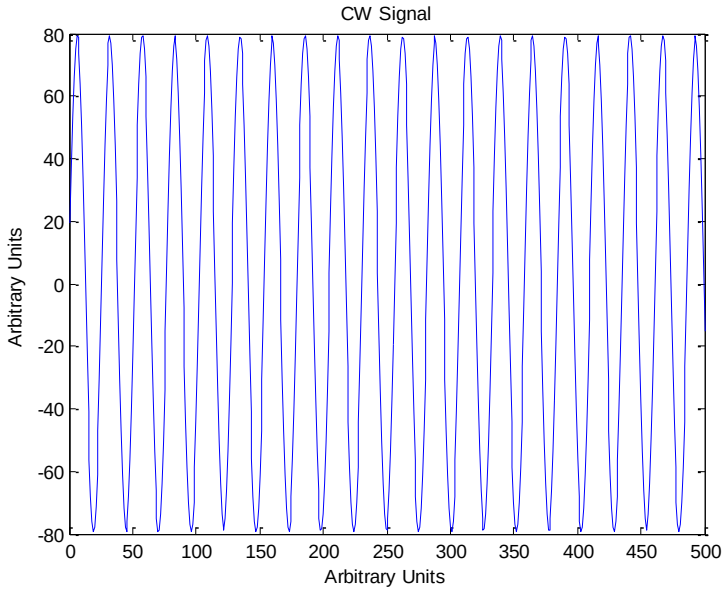


Figure 3-5: CW signal

The continuous-wave signal can be represented as:

$$x(t) = A\cos(2\pi f t). \tag{3.4}$$

Characteristics:

The CW signal has constant frequency and amplitude. This type of signal can either be pulsed or continuous. Only frequency, power, TOA, and PW are of concern.

Challenges:

No added encoder challenges are introduced.

2) Chirping (Linear)

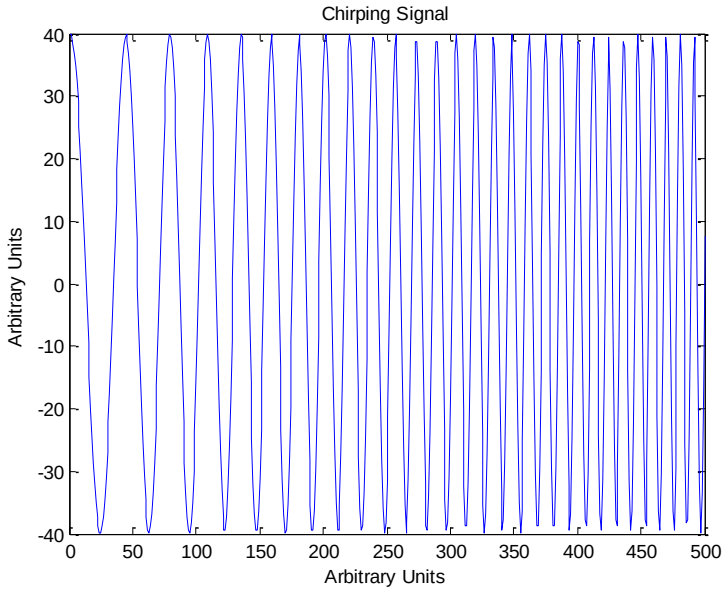


Figure 3-6: Chirping signal

A linear chirping signal can be represented as:

$$x(t) = Acos[f_o t + kt^2/2], \tag{3.5}$$

where f_o is the initial frequency and k is the chirp rate [10].

Characteristics:

The chirping signal starts at an initial frequency and changes its frequency linearly with time. All other signal characteristics remain constant. Along with the standard signal characteristics, the chirp rate is required to be a part of the PDW.

Challenges:

The different frequencies in each sequential frame can cause confusion when determining if it is part of a preexisting chirping signal or the start of a new signal. In this study, we only consider a chirp signal with a positive chirp rate, but the developed algorithm can be easily extended for a chirp signal with a negative chirp rate as well.

3) BPSK

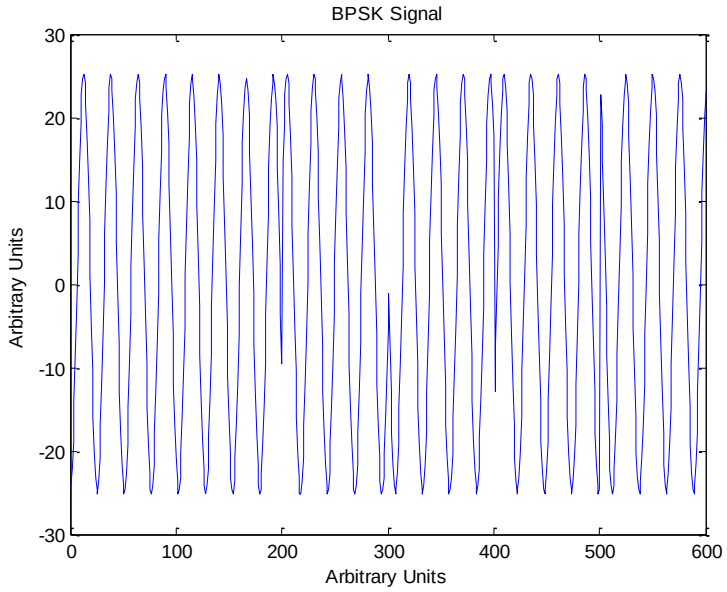


Figure 3-7: BPSK signal

(3.6)

A BPSK signal can be represented as:

$$x(t) = A\cos(2\pi f t + \phi)$$, where ϕ is either 0 or π [11].

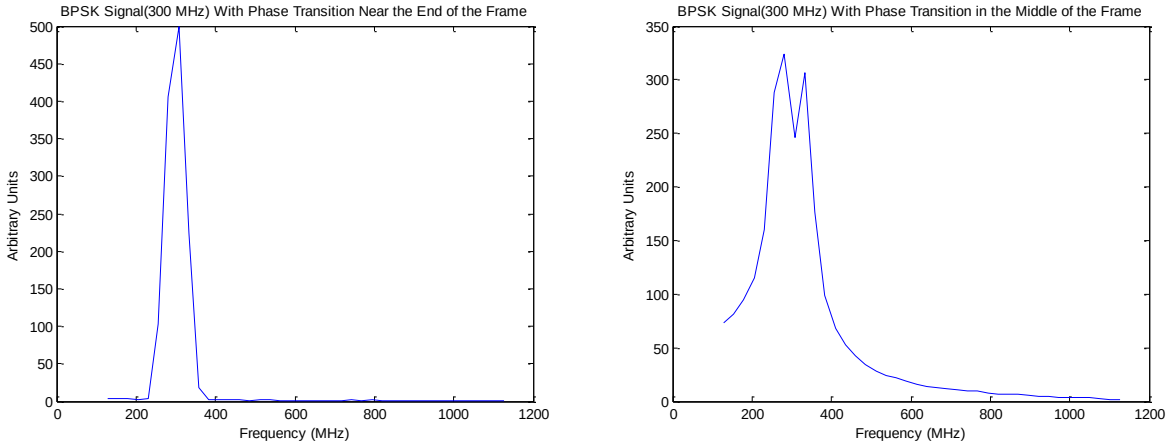
Characteristics:

The BPSK signal is a CW signal that periodically changes its phase. The phase can either be 0 or π. Along with the standard signal characteristics, the chip time is desired be a part of the PDW output. The chip time defines how fast signal phase transitions might occur.

Challenges:

Depending on where the phase change happens in a frame, the signal's spectrum may have a different appearance. When the phase change happens in the beginning or end of a frame, the spectrum of a BPSK signal is similar to the one of CW signal. However, when a phase change

happens in the middle of a frame, it causes two peaks at different but close frequencies. These two scenarios are shown below in Fig. 3-8. As a result, it is difficult to differentiate a CW signal from a BPSK signal and this difficulty will be addressed in Chapter 4.



(a) **(b)**

Figure 3-8: (a) FFT output of a frame of BPSK signal with a phase transition at near end and (b) FFT output of a frame of BPSK signal with a phase transition in the middle.

Chapter 4

PDW Generator

4.1 Description

An encoder has been created using MATLAB®. This encoder can precisely detect different types of signals. Since this system is designed to work in real time, the receiver only sees and evaluates a frame of data at a time (visualized in Fig. 4-1); therefore, the encoder must track information on all possible signals for the entirety of their lifespans and then average this information from all the frames in which the signal(s) were detected. Having the possibility of multiple signal types adds challenges with frame-by-frame detection. There needs to be logic distinguishing between the different types of signals. For example, the chirping signal might have an FFT peak in a different frequency bin in each sequential frame and when identify a peak in a FFT frame, the encoder will need to be able to decide if it is indeed part of a previous chirping signal or the start of a new signal. Another example is with a BPSK signal; when a phase change happens in the middle of the frame, the FFT output produces two separate peaks above the threshold. Again, there needs to be logic to check if these two peaks are caused by one BPSK or two CW signals that are close in frequency.

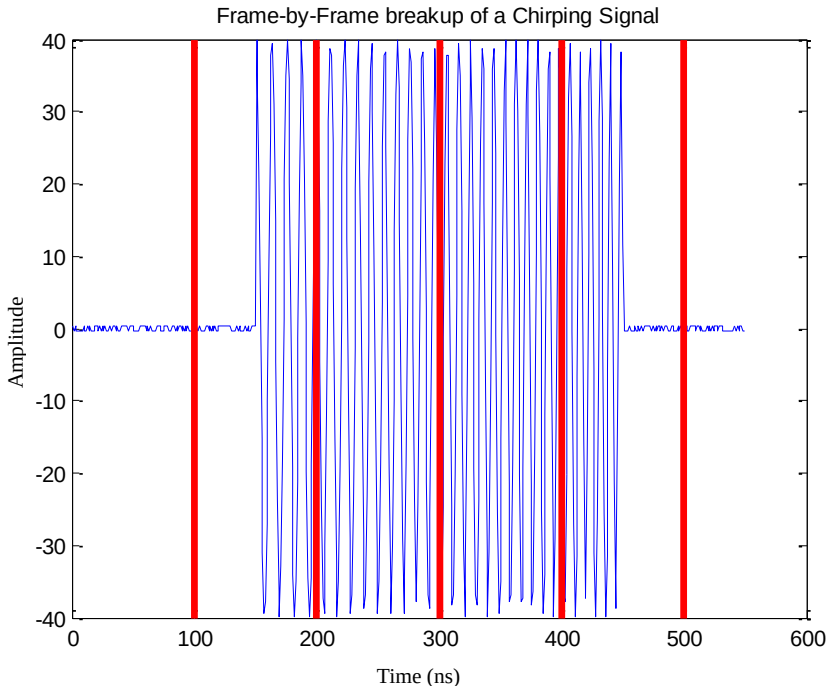


Figure 4-1: Frame-by-frame breakup of a signal as seen by the encoder

The PDW generator must be able to detect multiple signal types at any given moment in time. The methodology of how this generator functions will now be described in 9 steps; a flowchart will accompany this description following the completion of this section in Figs. 4-2 and 4-3.

First the PDW generator extracts any possible signal information on the first occurrence that a frame has a possible signal(s) over the threshold.

1) *Upon receiving the first frame of valid data (an FFT peak above the system's threshold) containing information, store each piece of information (power, frequency, phase, and starting frame) and consider it as the beginning of a possible signal.*

This initial stored data will be used as a benchmark to compare future information in the subsequent frames to determine what type of signal may be present.

2) *Each subsequent frame will have all relevant data above the threshold extracted and each of the possible signals within that frame will be moved on for further processing.*

Once the next frame of data has been received, three checks are performed on each piece of data extracted from the frame that has a power exceeding the threshold; these checks are performed on the previously stored data:

3) *If the frequency increases by 2 or more frequency bins and the power is similar to the previous frame of data, it will be flagged as a "possible chirping signal".*

4) *If the frequency is within 1 frequency bin and the power is similar to the previous frame of data, it will be flagged as a "possible continuous wave signal".*

5) *If the data does not match with any prior signal information, a new "possible signal" will be created and stored for future comparisons.*

The amount that the signal can differ in frequency and power was decided empirically by tracking the average difference for these values viewed when building and enhancing this PDW generator for improvements over the course of its development.

Once a possible CW or chirping signal has at least 3 frames of data, the encoder will flag that signal as being a real signal and it will lose the tag of "possible". Once this tag is removed, the encoder then performs checks on any CW signal to determine if it is a BPSK signal. There are 2 checks for the determining the existence of a BPSK signal:

6) *Check to see if the current frame has a phase shift in the middle of the frame (2 similar peaks will be present in the FFT output); if the two peaks have similar frequency, power, and phase, then that signal will average the frequency, power and phase and combine the two signals into one and flag it as a BPSK.*

7) Perform a double phase difference check [12]; this requires having at least 3 frames of data. The double phase difference determines if there is a phase shift within the duration of three frames. This is found by taking the phase of three consecutive frames and performing the following equation [12]:

$$Double\ Phase\ Difference = (\Theta_3 - \Theta_2) - (\Theta_2 - \Theta_1), \quad (4.1)$$

where Θ_i represents the phase of the detected signal in a given frame.

If the double phase difference results with and outcome of 3.14 plus/minus 1.5 radians, then that indicates that there was a phase change within those three frames and the signal will be flagged as a BPSK signal.

The double phase difference is one method proposed by Dr. James Tsui to determine the existence of a phase change [12]. The last step is to check the current "possible" and "flagged" signals to see if they have been updated to determine if each are either a real signal or an aberration.

8) After determining where each possible signal in the current frame belongs (either a continuation of a previous signal or the start of a new signal), the PDW generator then looks through the stored signals to see if there has been an update and continuation to each of the previous "possible" or "flagged" signals; if there was no update, that signal is marked with having 1 missed frame.

This encoder allows for multiple missed frames throughout the duration of a signal; however, there cannot be consecutive missed frames.

9) At the end of step 7, once there are 2 consecutive skipped frames for any "flagged" or "possible" signals, the encoder takes one of the two following actions: Delete the information if the detected frames were less than three frames, or produce a

PDW containing the pertinent information (frequency, power, time of arrival, pulse width, and/or chirp rate) if there are three or more frames of information.

The encoder allows for non-consecutive missed frames for two main reasons: the possibility of a high powered noise spike and the effects of spectral leakage. The spectral leakage issue can occur if another, high-powered signal arrives or leaves in the middle of a frame. This high-powered signal will spread its energy and possibly cover up a lower-powered signal, which will cause a missed detection.

Steps 2-9 will then repeat themselves continuously as long as the receiver is operational. Using this method described above, a comprehensive test was performed on the encoder to test its effectiveness and accuracy. Below, a flow chart visualizing the steps above is displayed:

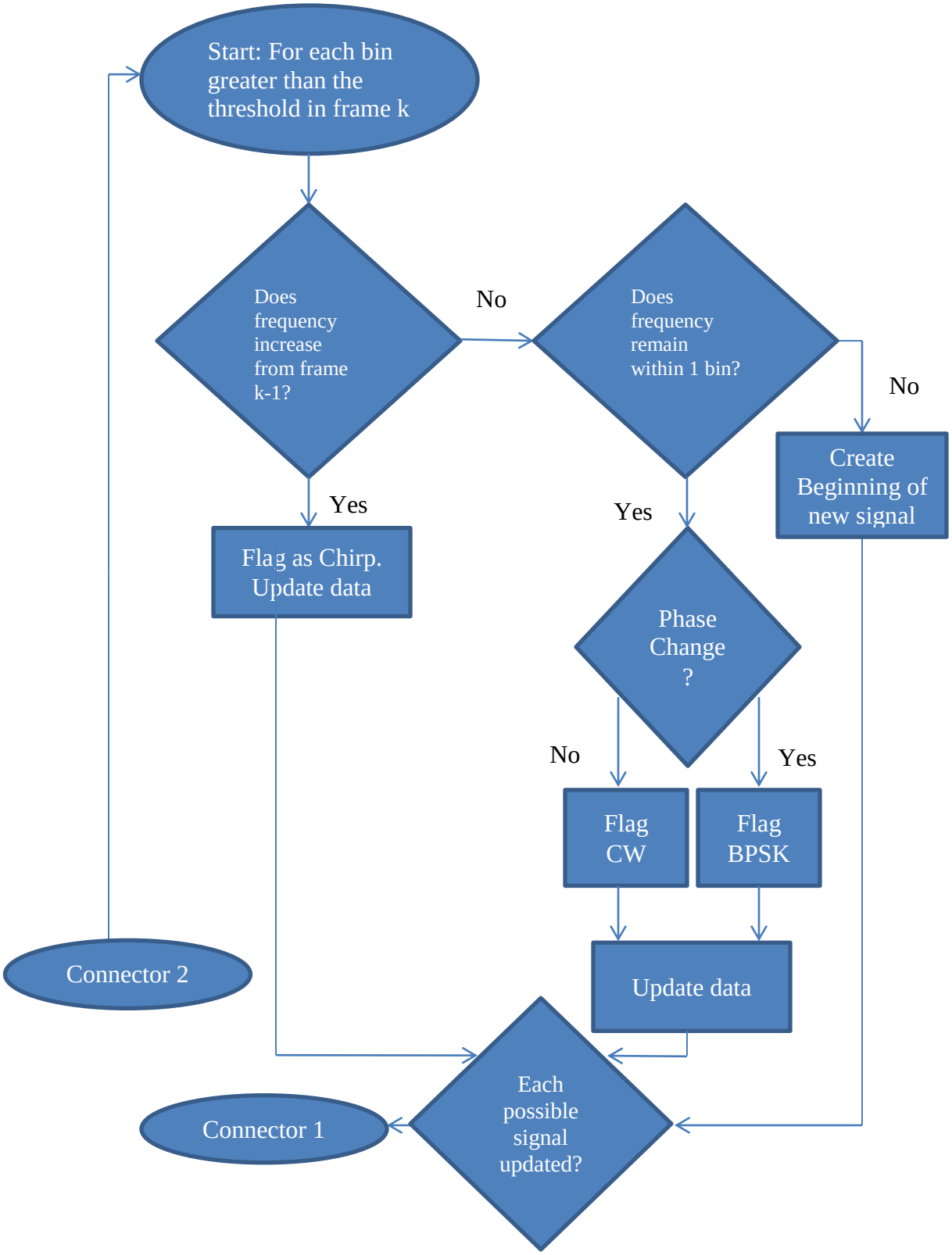


Figure 4-2: PDW flow chart part 1.

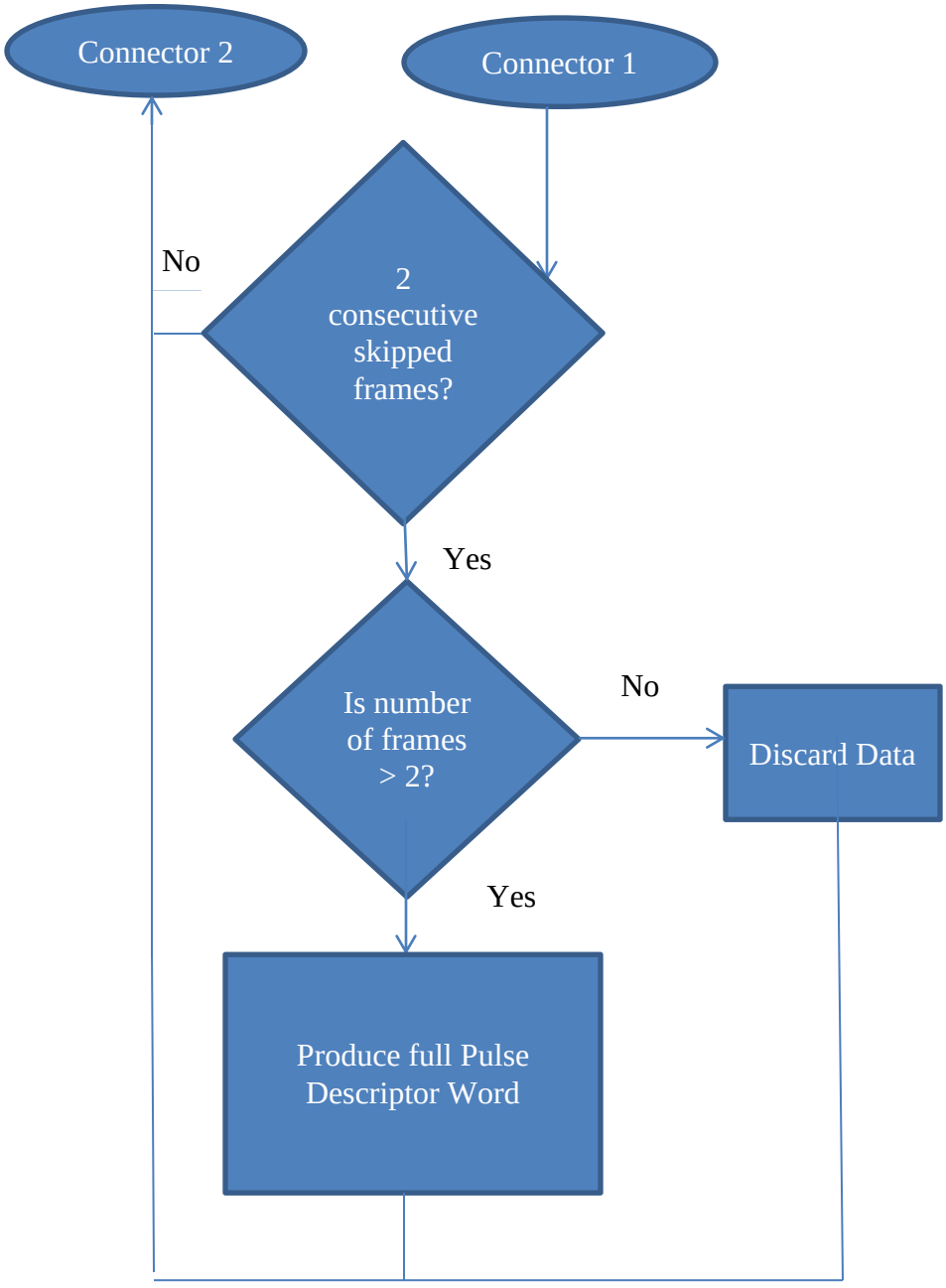


Figure 4-3: PDW flow chart part 2.

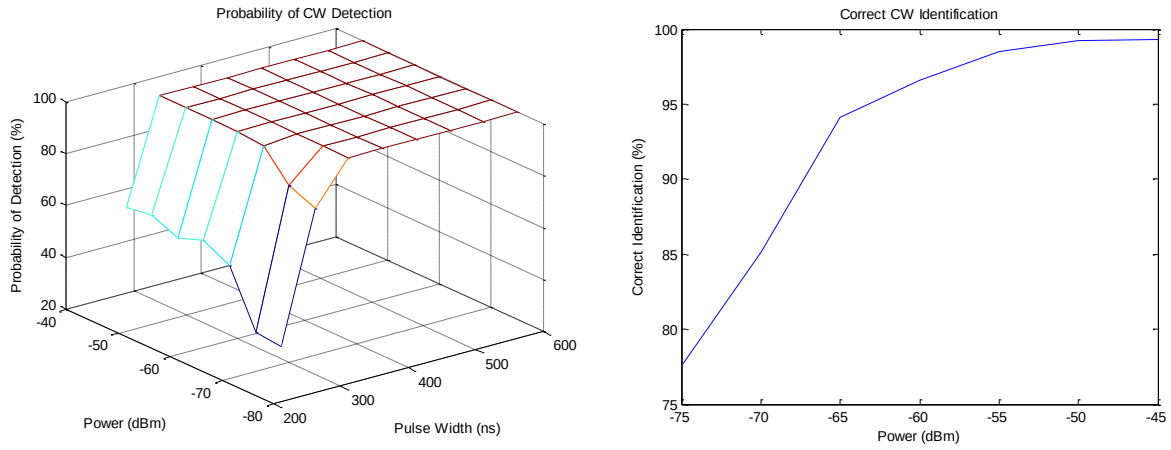
4.2 Comprehensive study

A simulation study was conducted to test the encoder's effectiveness on the performance accuracy by covering a wide variety of cases for each of the three signals types: CW, chirping, and BPSK. For each testing scenario, a sampling frequency of 2.56 GHz and a FFT frame size of 256 points were chosen, yielding a frequency resolution of 10 MHz and a frame length of 100

ns. A 256-point Blackman window was multiplied with the incoming signal before FFT was conducted. Thermal noise was taken into account in the simulation. Also, the signal went through a 10 bit quantization to give the signal a more realistic setting. Each of the following three sections will contain five plots: probability of detection, frequency estimation error, power estimation error, time of arrival estimation error, and pulse width estimation error. Along with those plots, the cases for CW and BPSK will also contain a plot showing the accuracy of classifying the signal correctly.

4.2.1 Continuous Wave Signal

Before performing the efficiency test of the encoder's accuracy for detecting a CW signal, two additional tests were carried out: The probability of detection and probability of correct signal identification. Figure 4-3 (a) shows the CW signal's probability of detection vs. signal power and pulse width. Figure 4-3(b), shows the probability of detection vs. signal power.

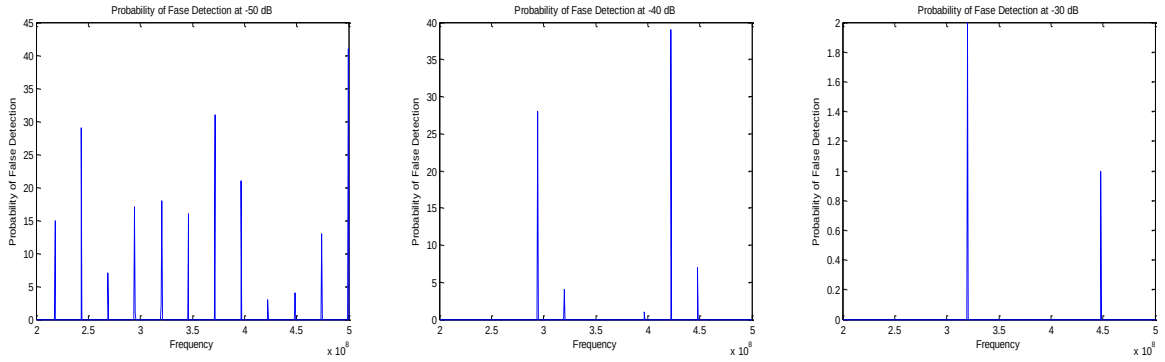


(a) **(b)**

Figure 4-3: (a) Probability of detection and (b) Probability of correct identification.

As shown in Fig. 4-3, the probability of detecting an incoming CW signal is high when the signal is an adequate length; however, when the pulse width is shorter, the probability of detection drops significantly. This is especially the case with lower powered signals. The probability of correctly identifying the signal correctly as a CW signal has an interesting characteristic. The probability of correct detection for low signal power is expected; a lower powered signal's phase

is more likely to be skew by noise and cause the encoder to misclassify it as a BPSK signal when using the double phase difference method. However, when the power is higher, it is expected that the encoder would identify the signal correctly 100 percent of the time and encoder did not achieve 100 percent correctness. This phenomenon deserves a detailed analysis. As stated previously, the encoder uses the double phase difference method to determine if a phase change is present in three frames of continuous data; however, this method has a flaw that there are CW signals with certain frequencies that occasionally cause the double phase difference to yield a value of π, which will be considered a phase change. Figure 4-4 demonstrates this issue and it is shown below:



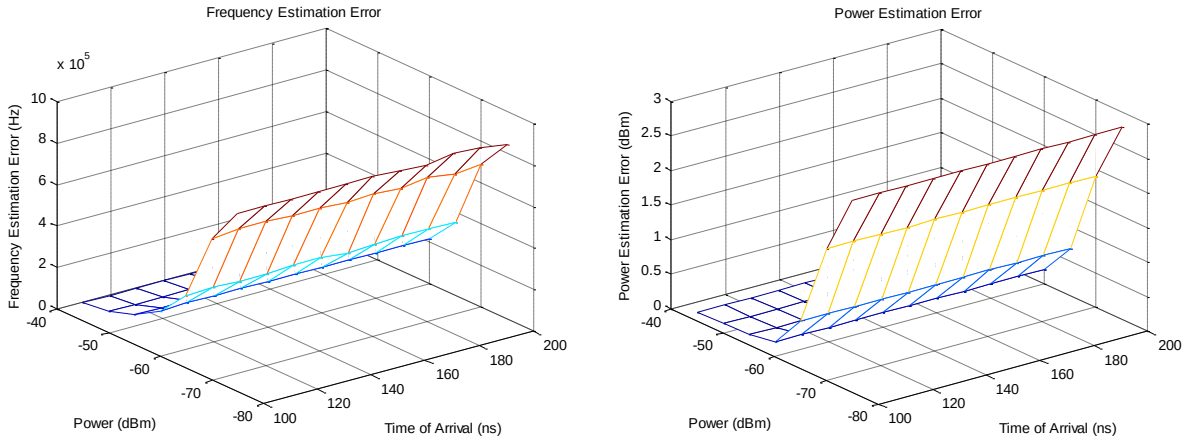
(a) (b) (c)

Figure 4-4: CW probability of false identification at signal power: (a) -50 dBm, (b) -40 dBm, and (c) -30 dBm

Figure 4-4 contains three plots detailing the probability of identifying a CW signal as a BPSK signal. As seen clearly in Fig.4-4 (a), every 25 MHz, the double phase difference will produce a double phase difference of π instead of 0 about 25% of the time. In a similar trend to the probability of correct identification plot in Fig. 4-3, when the power is low, there is a greater probability of false identification at those specified frequencies (every 25 MHz). Even with the higher power signal in Fig. 4-4 (c) (signal power: –30dBm), there is still a small chance of a false detection. For example, when there is a correct identification, the double phase difference should result in 0 or 2π. For instance, when one of the frequencies that artificially created a phase change had a correct result, the three consecutive phases resulted in Θ_3 = 2.58, Θ_2 = -0.72, Θ_1 = 2.28 yielding a double phase difference of 6.3. However, due to the 2π phase ambiguity, at that same frequency there was a chance that the three consecutive phases resulted in Θ_3 = 2.66,

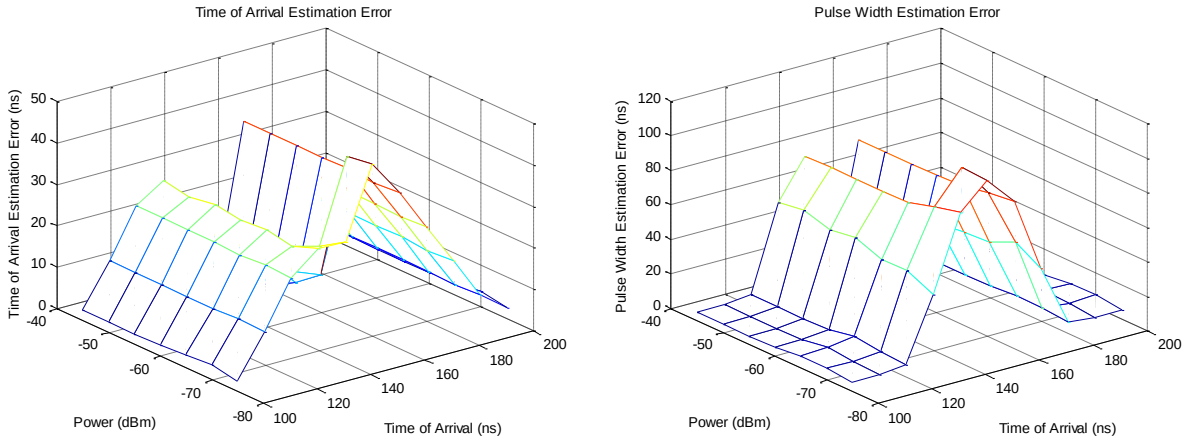
Θ_2 = -0.71, Θ_1 = -0.87 yielding a double phase difference of 3.2 thus causing an incorrect signal classification.

After analyzing the probability of detection and identification accuracy, a comprehensive test was performed on a CW signal. A cosine signal was tested with 3 controlled statistics: *Time of arrival*, *signal power*, and *pulse width*. Time of arrival was swept from the initial frame being completely full at 100 percent to the first frame only being partially full occupying 10 percent of the initial frame. The power of the signal ranges from low (−75 dBm) to high power (−45 dBm). Also, in this test, the pulse width is set to a constant 900 ns. The only other characteristic is frequency and it is a random variable between 300 MHz and 1000 MHz. Each combination of time of arrival and power was averaged over 1000 times. Four different testing scenarios are shown below: Frequency, power, time of arrival, and pulse width estimation error.



(a) **(b)**

Figure 4-5: CW (a) frequency estimation error and (b) power estimation error.



(a) **(b)**

Figure 4-6: CW (a) TOA estimation error and (b) PW estimation error.

As one can see in Fig. 4-5, the performance of the encoder is very accurate when the signal is of higher power for the frequency and power estimation; however, the accuracy deteriorates as the power decreases. The time of arrival and pulse width estimation error follow similar trends as shown in Fig 4-6; the estimation is worse when the frame occupation is around 30% and 70%. This outcome is due to the interpolation equation used. We estimated the signal duration in the first and last frame based on the ratio of signal power in the first and last frames with the averaged signal power of the other, or middle, frames. This interpolation formula works the best when the signal occupied the frame from the ranges of 30-70%. Signals occupying a small portion of frame might not be detected and signals occupying over 70% of frame are more likely to be viewed as occupying the entire frame. Also, when the frame occupation is 50% or at 50 ns and the power is very low, the estimation for time of arrival and pulse width is the worst due to the increased chance of miss detecting the starting and ending frames.

4.2.2 Chirping Signal

The probability of detection for the chirping signal follows the trend of the CW signal in that it has poor accuracy only when the signal power and the pulse width are low as shown in Fig 4-7. The plot indicating the encoder's ability to classify the signal as a chirping signal was omitted because it was correct 100 percent of the time when the signal power was high and the pulse width was long.

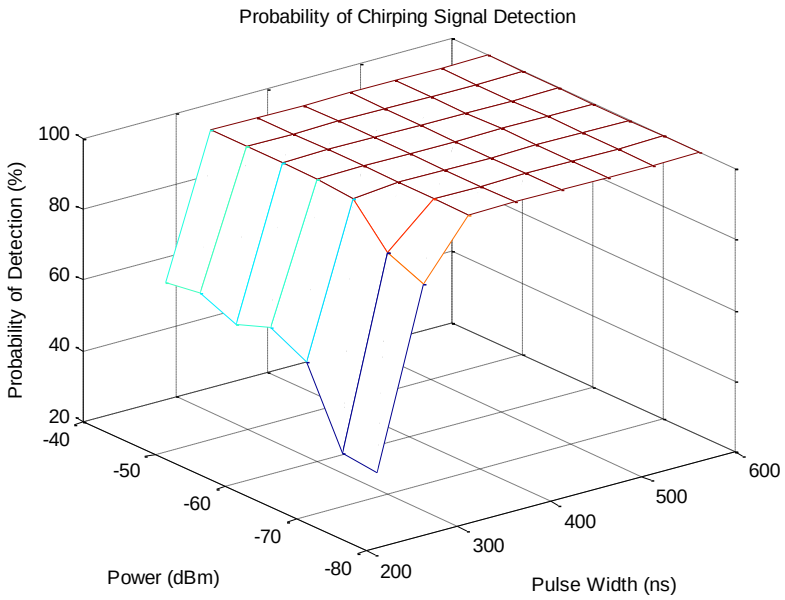
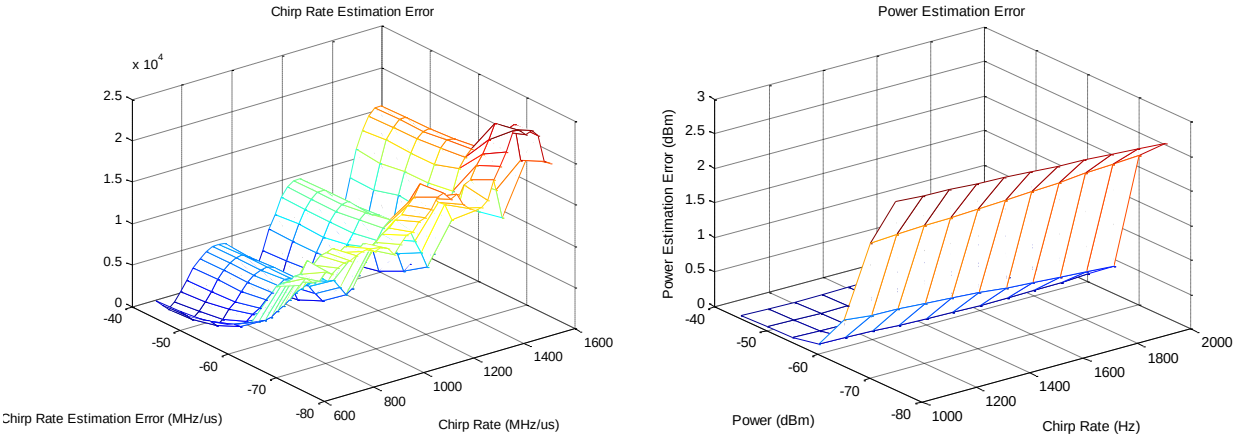


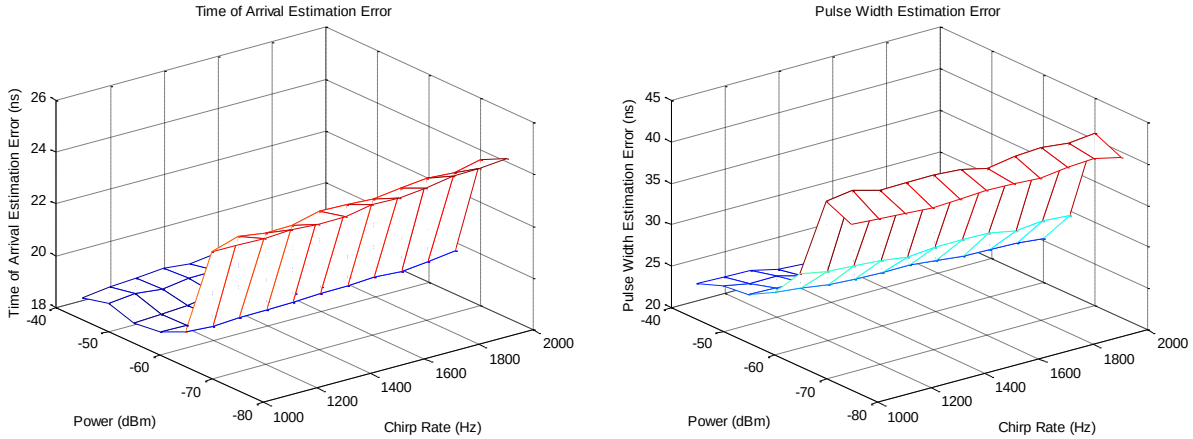
Figure 4-7: Chirping signal probability of detection.

A chirping signal was tested with 3 controlled statistics: chirp rate, signal power, and pulse width. Chirp rate was swept from 600 MHz/µs to 1600 MHz/µs. For this study, only an increasing chirp rate is tested. However, this encoder would easily be able to handle a decreasing chirp rate by using the same principles. The power of the signal ranged from low (–75 dBm) to high (–45 dBm). Also, in this test, the pulse width is set to 500 ns. The initial frequency and TOA were both random variables. Each combination of chirp rate and power was averaged over 1000 times. Four different testing scenarios are shown below: *chirp rate*, *power*, *time of arrival*, and *pulse width* estimation error.



(a) **(b)**

Figure 4-8: Chirping (a) chirp rate estimation error and (b) power estimation error.



(a) **(b)**

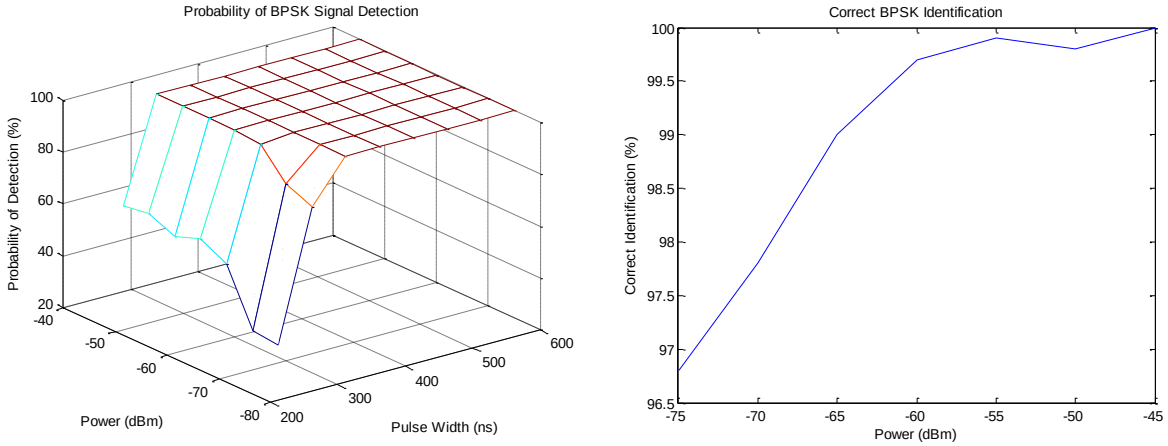
Figure 4-9: Chirping (a) TOA estimation error and (b) PW estimation error.

Viewing the plots in Figs. 4-8 and 4-9, chirp rate estimation and power estimation error both lose their precision as the power decreases. However, the chirp rate estimation error also loses precision as the chirp rate increases. This is due to the frame length and time resolution being fixed at 100 ns; as the chirp rate increases, the signal's frequency changes more significantly in one frame thus causing an inaccurate center frequency estimation. Another feature presented in the chirp rate estimation error plot of Fig. 4-9 is the wave-like phenomenon. This is present because at certain chirp rates, the encoder will receive frequencies that are a multiple of the frequency resolution; this causes the frequency estimation to be more precise where the error dips and the error will then increase when the chirp rate introduces frequencies that are not a multiple of the frequency resolution [13]. Expectedly, the TOA and PW estimation errors worsen as the power decreases as shown in Fig. 4-9.

4.2.3 Binary Phase Shift Keying Signal

The probability of detection and correct identification results for BPSK signals are shown in Fig. 4-10. Again the probability of detection follows in a similar trend as the CW and chirping signals as illustrated in Fig. 4-10(a). The graph in Fig. 4-10 (b) showing the percent of correctly

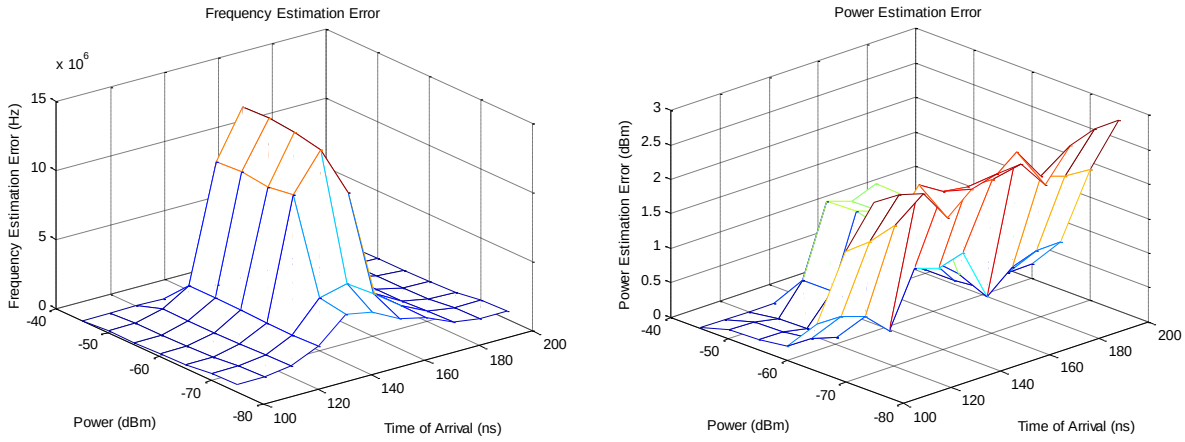
identifying the signal as a BPSK signal is much more correct when comparing it to that of the same graph for the CW signal in Fig. 4-3(b). This outcome is due to the fact that is much easier to classify a BPSK signal in comparison to a CW signal due to the double phase difference issue mentioned previously.



(a) **(b)**

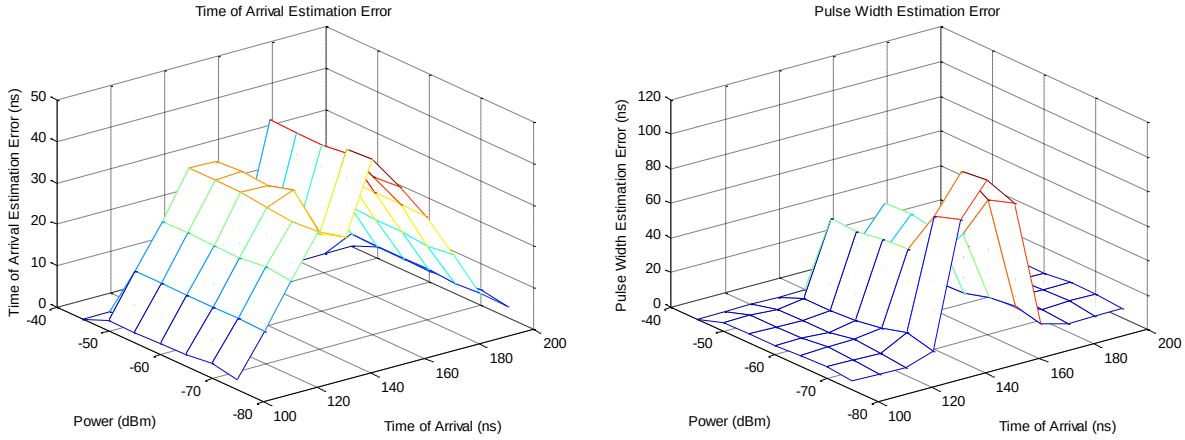
Figure 4-10: BPSK (a) Probability of detection and (b) Probability of correct identification.

Again, similar to the CW test, a cosine signal was tested with 3 controlled statistics: *time of arrival*, *signal power*, and *pulse width*. Time of arrival was swept from the initial frame being completely full at 100 percent to the first frame only being partially full occupying 10 percent of the frame. The power of the signal ranges from being low (–75 dBm) to high power (–45 dBm). Also, in this test, the pulse width is set to 900 ns. For this case, the frequency and chip time were randomly set. Each combination of time of arrival and power was averaged over 1000 times. Four different testing scenarios are shown below: Frequency, power, time of arrival, and pulse width estimation error. The simulation results are shown in Figs. 4-11 and 4-12.



(a) **(b)**

Figure 4-11: BPSK (a) frequency estimation error and (b) power estimation error.



(a) **(b)**

Figure 4-12: BPSK (a) TOA estimation error and (b) PW estimation error.

One of the biggest differences for the results of this testing scenario is the frequency and power estimation error shown in Fig. 4-11. One will note the elevated estimation error in the middle of these plots when the time of arrival is near the middle of a frame. In this test, the phase remained constant for one full frame; therefore, when the TOA starts in the middle of a frame (130-160 ns), each phase change will then occur in the middle of a frame. As we know, when there is an instance when the phase change occurs in the middle of a frame, the FFT output will produce two very similar and close signal peaks and the elevated error is due to those two peaks having

their characteristics averaged. The time of arrival and pulse width both follow the same trends that are seen with the CW signals as shown in Fig. 4-12.

The simulation results presented in this Chapter demonstrate that the developed encoder program can successfully detect different signals and generate accurate PDW when some conditions (signal power, PW, etc) are satisfied. The issue of correctly differentiating between CW and BPSK signals remains an unsolved issue and will be a topic for future study. The next chapter will present the I/Q channel base receiver design which can double the working bandwidth of an EW receiver.

Chapter 5

I/Q Based Channel Receiver

5.1 Introduction

As we know, to increase the performance of an EW receiver, it is desired to have a broad working bandwidth since there is no prior knowledge of the incoming signal's frequency. However, hardware limits the working bandwidth in typical receiver designs. A typical receiver will be able to detect signals that are in the first Nyquist zone, or signals whose frequencies are less than that of half the sampling frequency [14]. If the signal's frequency is not in the first Nyquist zone, aliasing will occur and this will cause erroneous results, which would force the EW system to spend valuable resources in trying to process erroneous signals. For example, conducting FFT upon a real-valued signal whose frequency is in the second Nyquist zone will produce a symmetrical spectrum as seen in the left plot of Fig. 5-1. The symmetrical spectrum makes the signal in the first and second Nyquist zones undistinguishable thus causing the second half of the spectrum to be redundant and useless [14].

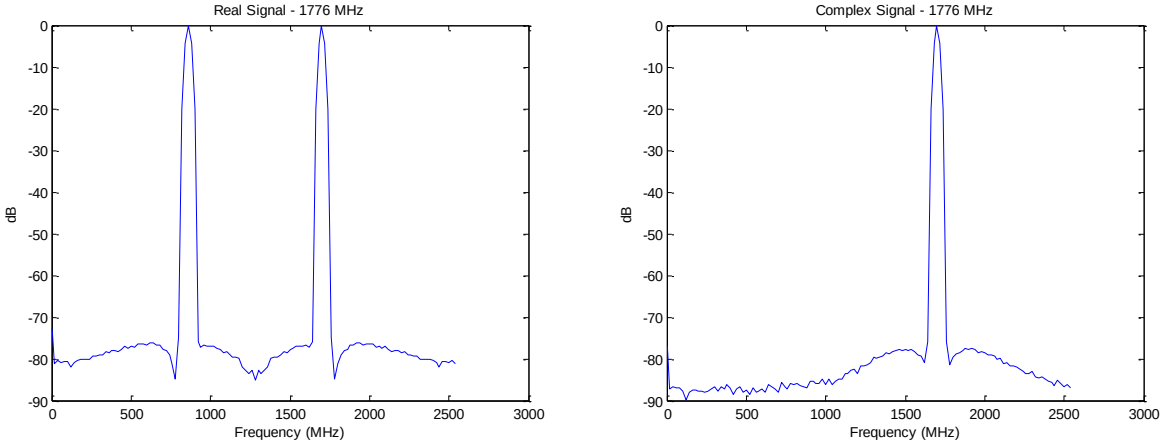


Figure 5-1: FFT output (left) of a real input signal and the FFT output (right) of a complex Signal

An EW receiver designer is limited by the ADC when choosing a sampling frequency for the system. The receiver's working bandwidth is less than half of the sampling rate, f_s. One method to increase the receiver's working bandwidth without increasing f_s is to introduce an I/Q based channel into the EW receiver design.

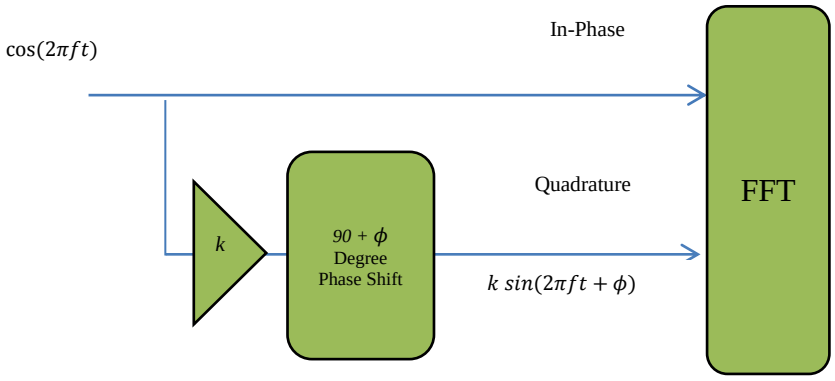


Figure 5-2: I/Q based channel receiver

An I/Q based channel receiver, shown in Fig. 5-2, takes the incoming, real-valued signal and splits it into two ways: an in-phase (I) signal and a 90 degree shifted quadrature (Q) signal. This is achieved by using a 90 degree coupler. By shifting the phase of the second signal, the real-valued signal is transformed into a complex signal (I-real-part, Q-imaginary part). When performing the FFT of a complex signal, the output does not produce a symmetrical spectrum. This can be seen in the right plot of Fig. 5.1. This effectively doubles the working bandwidth by removing the previous redundant copy shown in the left plot of Fig. 5-1. However, this simulation is conducted assuming a perfect coupler. In reality, there are certain amplitude and phase mismatches in the quadrature signal that can produce a nonexistent signal to show up on the spectrum. Figure 5-2 shows the path that the quadrature signal takes. In this path, the signal passes through an amplifier and phase shifter. In the ideal case, the value of *k* of the amplifier should be 1 and the phase should be 90 degrees with ϕ equal to 0. However, in practice, the value of the amplifier and phase shift will not be perfect. If the value of *k* is off by a minimal margin or the phase is marginally different from 90 degrees, a "ghost" image will appear [15]. The signal spectra of an I/Q receiver with a perfect coupler and a realistic coupler is shown below for comparison in Fig. 5-3.

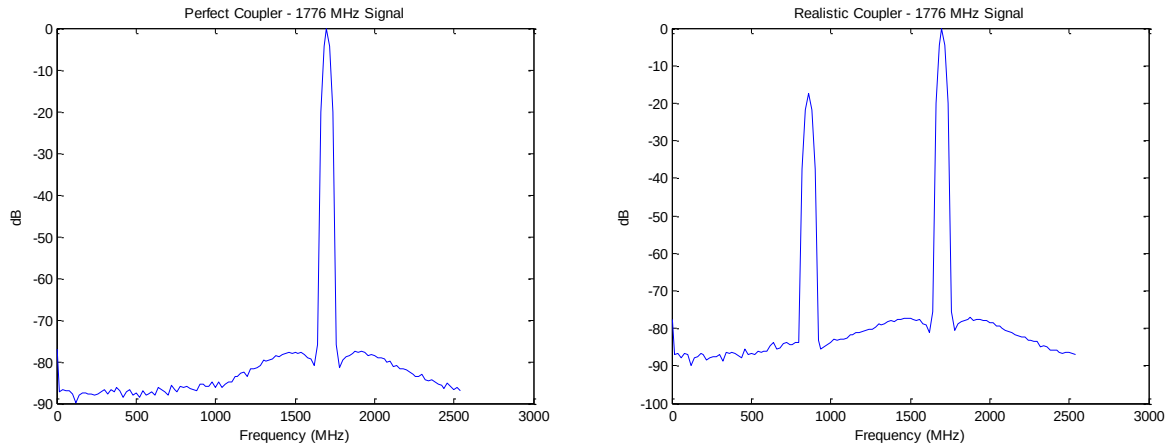


Figure 5-3: FFT output (left) of a perfect coupler and the FFT output (right) of a realistic coupler.

This nonexistent signal, shown in the right graph of Fig. 5-3, will force the threshold to be heightened to diminish the possibility of identifying the false image signal as an actual signal. This will cause the instantaneous dynamic range of the receiver to greatly suffer. Furthermore, this signal could be in a location in the spectrum that could mask a real signal with a lower magnitude that shares a close frequency bin. To diminish the false signal, compensation methods are needed. This is referred to as I/Q compensation.

5.2 Amplitude/Phase Compensation

We noted that, the amplitude and phase imbalance caused by an imperfect coupler can create an image signal and thus reduce the instantaneous dynamic range of an EW receiver. As a result, the I/Q channel based EW receiver has not been widely deployed. One of the issues of compensating the I/Q imbalance in an EW receiver is that, unlike a communication receiver, the EW receiver assumes no *a prior* knowledge about incoming signal frequency and the amplitude and phase imbalance of a realistic coupler is frequency dependent. Many imbalance methods focus on communication systems [16 - 23] and are not suitable for the EW system with no prior signal knowledge and that will need to be able to detect multiple signals at once.

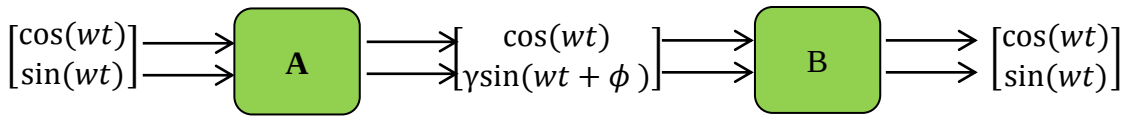


Figure 5-4: I/Q compensation block diagram

Dr. Lihyeh L. Liou of the AFRL developed an I/Q imbalance compensation method covering two Nyquist zones for EW receiver applications [24]. Since the amplitude/phase imbalance of a realistic coupler can be characterized in advance, it is possible to develop a filter to compensate the I/Q imbalance over the receiver's working bandwidth. The basic concept is to characterize the coupler's amplitude/phase frequency dependent imbalance and derive its inverse system (the equalizer) with a complex valued Finite Impulse Response (FIR) filter. Figure 5-4 shows the path through which the incoming signal travels. The incoming signal passes through matrix A, which represents the phase and magnitude imbalance of the realistic 90 degree coupler. Then, matrix B represents the filter through which the imbalanced signal travels resulting in a signal with a diminished "ghost" image. In Fig. 5-4, γ represents the amplitude mismatch and ϕ represent the phase mismatch. Notice that the coefficients of matrices *A* and *B* are frequency dependent. Matrix *A* can be determined based on the measurement result of the coupler and matrix *B* can then be determined from matrix *A*. Revising the matrix representation of matrix B into a complex-number format and taking the inverse Fourier transform of the resulting coefficients in the frequency-domain, we can derive the operation of the compensator in the time-domain, which will be in the format of an FIR filter whose representation is given in the Eq. 5.1 [24].

$$z_{out,k} = \sum_{m=-M}^{M} c_m z_{in,k-m} + \sum_{m=-M}^{M} d_m z^*_{in,k+m}\,, \qquad (5.1)$$

where c_m and d_m are the filter tap coefficients, z_{in} is complex valued inputs (I and Q channels), z_{out} is the complex valued output (I and Q channels), and there are (*2M+1*) tap coefficients. The coefficients c_m and d_m are derived based on frequency dependent matrix *B* shown in Fig. 5.3.

The I/Q compensation algorithm summarized here can be found in Dr. Liou's upcoming publication and AFRL technical report [24].

5.3 I/Q Compensation Simulation

The first step to implement this compensation scheme in hardware was to simulate the behavior using a hardware language for an FPGA and compare it to the verified MATLAB® results; the utilized FPGA language was VHDL. We would like to verify that the code and simulations written in MATLAB could be replicated and matched in VHDL. Not only will the following plots show the validity of the VHDL code, but they will also demonstrate the effect different amplitude and phase imbalance profiles have on the effectiveness of the proposed compensation scheme. For this test, a sampling frequency of 2.56 GHz was chosen with a 256 point FFT and the frequency of the signal was set arbitrarily to 1776 MHz (located in the 2^{nd} Nyquist zone). Figure 5-5 shows the coupler's amplitude and phase mismatch used to produce the false image. The spectra of coupler's imbalanced signal and the equalized signals using the compensation scheme in both the MATLAB® and VHDL programs are shown in Fig. 5-6. This imbalance case was created to show the performance of the compensation scheme when the imbalances are almost constant (spanning from an amplitude ratio of .8 to .8001 and a phase difference of -.2 to -.1999 radian). Notice that testing signals used in following simulations are multiplied with a Blackman window after A/D conversion to reduce spectral leakage. The equalizer tap length is 11. Figures 5-6 illustrate the effectiveness and success of the VHDL implementation of this compensation scheme; the amplitude and phase imbalance were artificially created to replicate the effects a hybrid could have on an incoming signal.

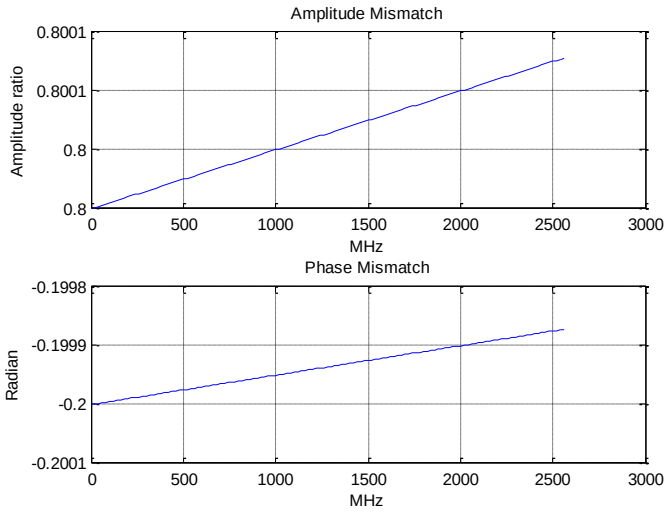


Figure 5-5: Amplitude (upper) and phase (lower) imbalance profiles.

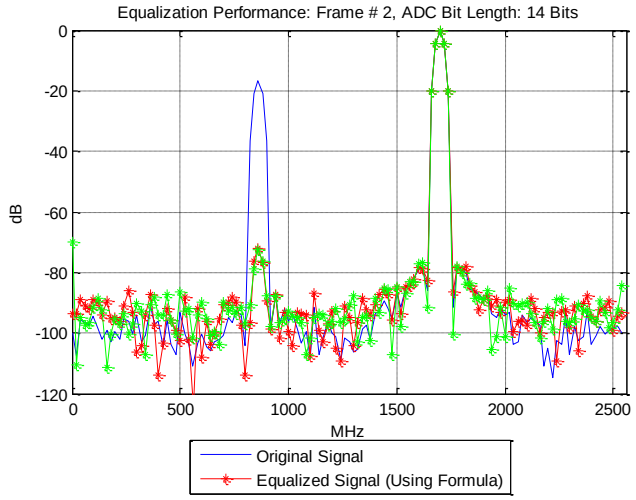


Figure 5-6: Compensation scheme verification 1

As seen in Fig. 5-6, this almost constant imbalance profile produces a false image, as expected, but the equalizer drops the power of the false image greatly by almost 60 dB. Also, in this plot, the VHDL implementation (shown in green) of the compensation scheme is verified as it matches the MATLAB® outcome, represented in red.

Another test to verify the validity of the VHDL code was performed with a non-symmetric amplitude and phase imbalance profile which is shown in Fig. 5-7. In comparison to the

previous imbalance profile, the mismatch for the amplitude and phase deviate greatly from their starting locations (spanning from an amplitude ratio of .8 to 1.05 and a phase difference of –0.2 to .05 radian).

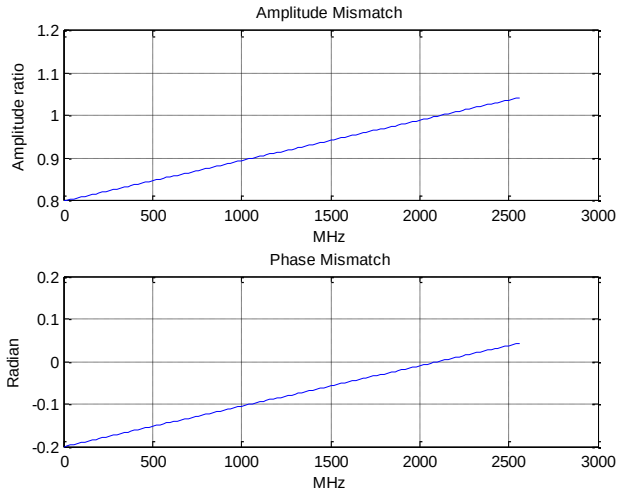


Figure 5-7: Amplitude (upper) and phase (lower) imbalance profiles.

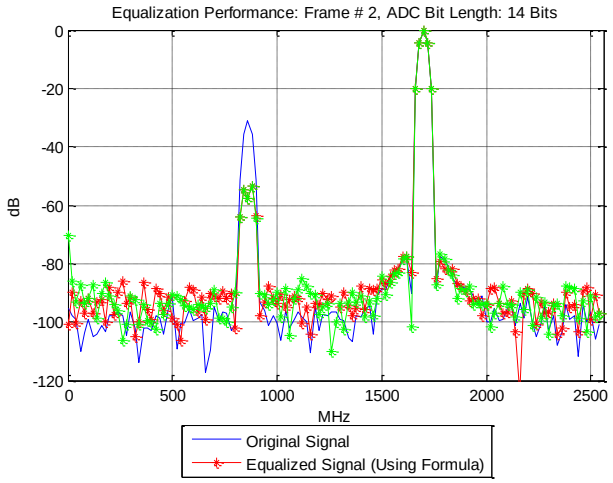


Figure 5-8: Compensation scheme verification 2

The spectra of coupler output and equalizers' output are illustrated in Fig. 5-8. The VHDL outcome (shown in green) is once again verified, as it matches the MATLAB® results in red; however, it is interesting to note the effects the imbalance profile has on the compensation scheme. In comparison to the previous false image power drop, the false image is only lowered by 25 dB, which is still a significant reduction.

To show the effectiveness of the compensation scheme on the full working bandwidth of a particular case, another phase and amplitude imbalance vs. frequency was created and shown in Fig 5-9. Figure 5-10 shows the amount in dB the image signal's peak drops for the full bandwidth after being passed through the equalizer. The performance of equalizer is frequency dependent and the image deduction ranges from 55dB to 28dB. It is interesting to note that there is an image power drop for all frequencies except for frequencies around 1.28 GHz, or half of the sampling frequency of 2.56GHz. This is because that frequency is the folding frequency located at half the sampling rate and the original and image signals occupy the same frequency and cannot be separated.

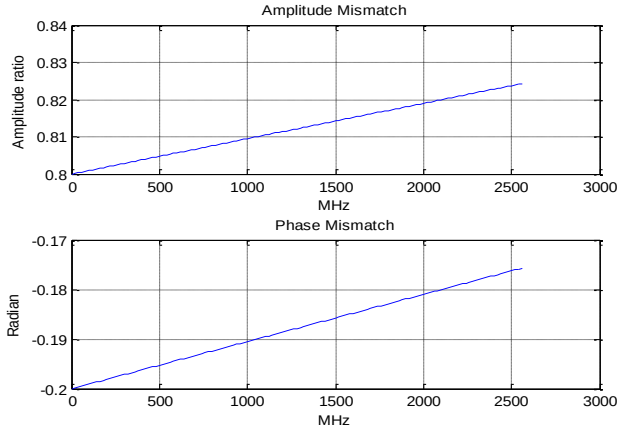


Figure 5-9: Phase and amplitude imbalance 3

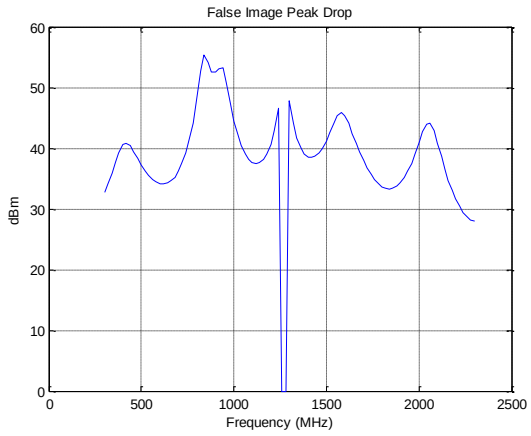


Figure 5-10: Filter performance on the entire working bandwidth

Hardware Implementation

6.1 Introduction

The implementation of the I/Q compensation scheme was the next step after verifying its viability in simulation. The most concerning difference between hardware and simulation were the limitations of the hardware constraints; specifically, the clock speeds and the processing power of the device itself forced the implementation to be carried out differently than it was in simulation. The Calypso-V6 device was used to carry out this project at Wright Patterson Air Force Base. On this board, the Virtex-6 was the FPGA utilized and the incoming data was passed through two ADCs (one for each of the I and Q channels).

6.2 Implementation

The physical constraints of the Calypso-V6 are centered on the ADC's sampling rate of 1.5 GHz. The filter equation that the FPGA needs to perform was shown in Eq. (6.1).

$$z_{out,k} = \sum_{m=-M}^{M} c_m z_{in,k-m} + \sum_{m=-M}^{M} d_m z^*_{in,k+m}\,, \tag{6.1}$$

This equation presents multiple issues for hardware implementation. One may note that the tap coefficients, c_m and d_m, both have real and imaginary components. Along with the real and imaginary components of the input and the coefficients, this filter will have eight multiplications for each filter tap. The final amount of operations for one output calculation is then *2M+1* multiplied by 8. For example, if there are 10 filter taps, there will be 80 multiplications to calculate. This amount of computations would not allow for the Virtex-6 to meet timing and hardware requirements. Due to this, the incoming signal had to be broken into parallel processing paths. Downsampling the input signal effectively reduces the sampling rate. For example, with the sampling rate of this system at 1.5 GHz, downsampling by two would reduce

the sampling rate to 750 MHz. For this hardware implementation, the input signal was split into 4 parallel processing paths and the downsampling is conducted in each path

This difference between theory and actual implementation is very critical; the compensation method was developed based on the filter being performed on consecutive samples and because of this, a straight-forward implementation into the FPGA would not be possible. To meet all timing constraints for the Calypso-V6, the signal had to be split into four parallel paths [25]. For this to work, a polyphase filter implementation was utilized.

The polyphase filter method is designed to have the parallel data paths to be filtered by filter taps broken into the similar layout as the parallel data paths. For the actual implementation, a filter tap length of 11 was utilized. Figures 6-1 and 6-2 below show how the tap coefficients, C_X , was split to match the incoming data [25].

C_1	C_2	C_3	C_4	C_5	.	.	.	C_{11}

Figure 6-1: Filter tap coefficients.

Branch 1	C_1	C_5	C_9
Branch 2	C_2	C_6	C_{10}
Branch 3	C_3	C_7	C_{11}
Branch4	C_4	C_8	**0**

Figure 6-2: Filter tap coefficients in parallel form.

Once both the filter taps and incoming signal are in similar, parallel forms, 4 branches are created. The overall system is shown in Fig. 6-3. Each branch will receive the same data and process it with a three-tap FIR filter. For example, on four clock cycles, the system will receive 4 samples. These 4 samples will be fed into the 4 parallel paths and each branch will get the same data. The only difference between branches is the coefficients that it will utilize for the FIR filter taps. By viewing Fig. 6-3, each branch's main outcome is to produce four outputs, $out_{m,n}$, where *m* is the branch number and *n* is the sample number. Once each branch produces their different outputs, they are combined and reconstructed to yield a result that would mirror that of a straight forward implementation.

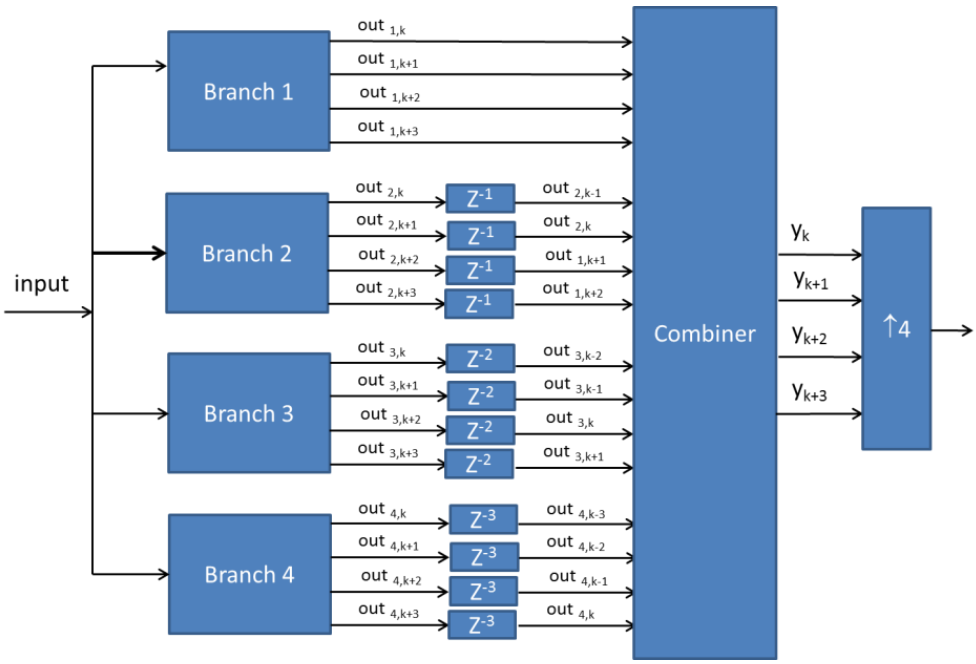


Figure 6-3: The overall system of the hardware implementation

To explain the functionality of each branch, let us consider branch 1. Figure 6-4 demonstrates the path the data takes to produce the four outputs for this specific branch. For branch 1, the three filter taps that would be used in path 1 consists of the coefficients: c_1, c_5, and c_9. This can be seen in Fig. 6-4. The input signal will be split to four paths, experience different delays, be downsampled, and be processed by a three-tap FIR filter. Again the only difference between the four branches will be the filter coefficients used. Therefore, as an example, Branch 2 will look the same as Fig. 6-4 except with filter coefficients from path 2 (shown in Fig. 6-2).

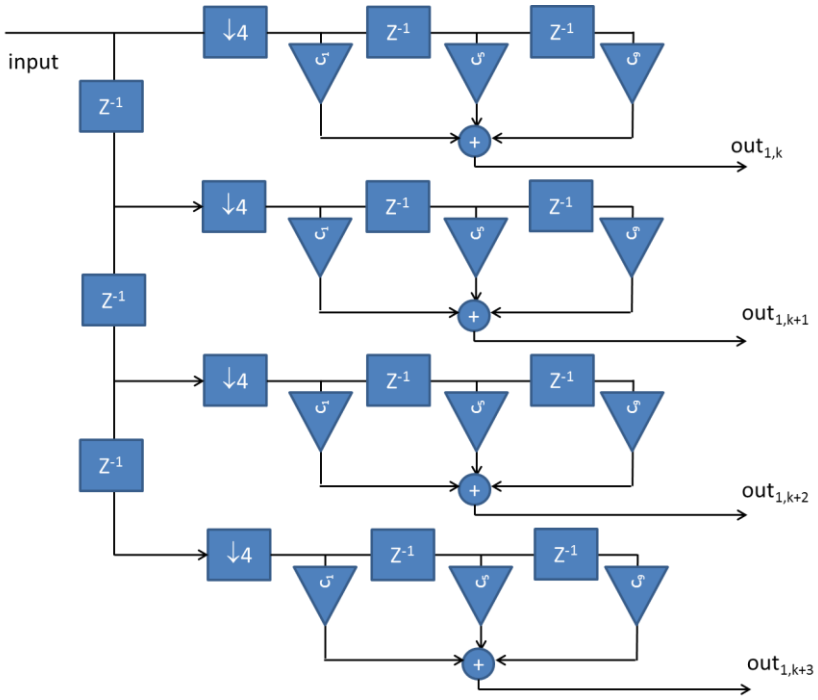


Figure 6-4: Data path to produce outputs for branch 1.

The four outputs produced by each branch (Fig. 6-4) will then go through the combiner, as shown in Fig. 6-3. The final outputs are found by summing the appropriate samples of the four different filter branches and output of each branch experience different delays before the summation. Therefore, the summation groups are a combination of the current cycle of input samples and the previous samples [25]. Due to this, the hardware will need to store not only the current set of samples, but the previous sets of samples as well. The final output, y_k, is calculated following these six steps:

1) Take the sample of the current set of branch 1matching the position of the output being computed (e.g. y_1 *corresponds to* $out_{1,1}$, y_2 *corresponds to* $out_{1,2}$, etc.,)

2) Take the previous sample (delayed by 1 from the first sample) of branch 2

3) Take the previous sample (delayed by 2 from the first sample) of branch 3

4) Take the pervious sample (delayed by 3 from the first sample) of branch 4

5) *Sum all four appropriate samples to achieve the final output as shown in Fig. 6-5)*

6) Repeat these steps for all four outputs (y_k, y_{k+1}, y_{k+2}, and y_{k+3})

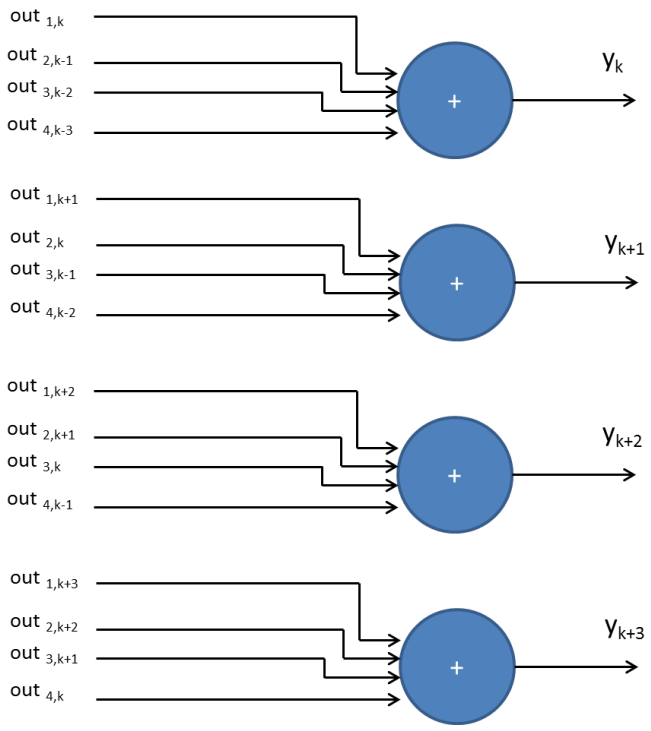


Figure 6-5: Summation groups to produce final outputs.

Once all the outputs of four clock cycles are complete, they are upsampled as illustrated in Fig. 6-3 and reconstructed to form a constant stream of output samples [25]. This method will then repeat itself with the introduction of the new set of 4 data samples; the current set of data will be saved as the previous set of data and the new data will be viewed as the current set of data.

For a more general form, let's focus on the outputs y_k, where k is greater than 11; *k* values of 12 and greater consists of outputs with all non-zero components. The general equation to calculate the final outputs (Fig. 6-5) is shown below:

$$y_k = \sum_{a=k}^{k-4} out_{k-(a-1),a}\,, \tag{6.2}$$

And the four equations for the different outputs for each branch (Fig. 6-4) are shown below:

$$out_{1,k} = s_k * c_1 + s_{k-4} * c_5 + s_{k-8} * c_9 , \quad (6.3)$$

$$out_{2,k} = s_k * c_2 + s_{k-4} * c_6 + s_{k-8} * c_{10}, \quad (6.4)$$

$$out_{3,k} = s_k * c_3 + s_{k-4} * c_7 + s_{k-8} * c_{11}, \quad (6.5)$$

$$out_{4,k} = s_k * c_4 + s_{k-4} * c_8 + s_{k-8} * 0, \quad (6.6)$$

where s_k is the k^{th} input sample.

To show how this system would work, consider the output y_{13}; the four outputs from the four different branches that would be used to produce y_{13} is shown below in Eqs. (6.7 – 6.10) and follows the form described in Fig. 6-5 and Eq. (6.2).

$$out_{1,13} = s_{13} * c_1 + s_9 * c_5 + s_5 * c_9 \quad (6.7)$$

$$out_{2,12} = s_{12} * c_2 + s_8 * c_6 + s_4 * c_{10} \quad (6.8)$$

$$out_{3,11} = s_{11} * c_3 + s_7 * c_7 + s_3 * c_{11} \quad (6.9)$$

$$out_{4,10} = s_{10} * c_4 + s_6 * c_8 + s_2 * 0 \quad (6.10)$$

It is clear to see that the summation of these 4 outputs would produce the following result:

$$y_{13} = out_{1,13} + out_{2,12} + out_{3,11} + out_{4,10} \quad (6.11)$$

$$y_{13} = s_{13} * c_1 + s_{12} * c_2 + s_{11} * c_3 + \cdots + s_3 * c_{11} \quad (6.12)$$

This result mirrors what a straight forward implementation of an 11-tap FIR filter would produce.

6.3 Equipment and Setup

Along with the Calypso-V6, the other main piece of equipment was the 90 degree coupler. The particular coupler used for this test has a working bandwidth from 500 – 2000 MHz; this left a zone between DC and 500 MHz that was not specified by the specific coupler. Before testing the

compensation scheme as the sum of multiple polyphase FIR filters, the filter tap coefficients had to be produced.

The first step to obtain the filter tap coefficients was to calibrate the coupler over the input range of the ADC (DC to 1500 MHz). The calibration results for this specific coupler are shown below and were found by sweeping the frequency range of the working bandwidth with a step frequency of 50 MHz. This data was followed by using an interpolation routine to obtain a set of characterization curves with a fine resolution which was ready for frequency-time transfer. The characterization results of the phase and amplitude imbalance are shown in Fig. 6-6.

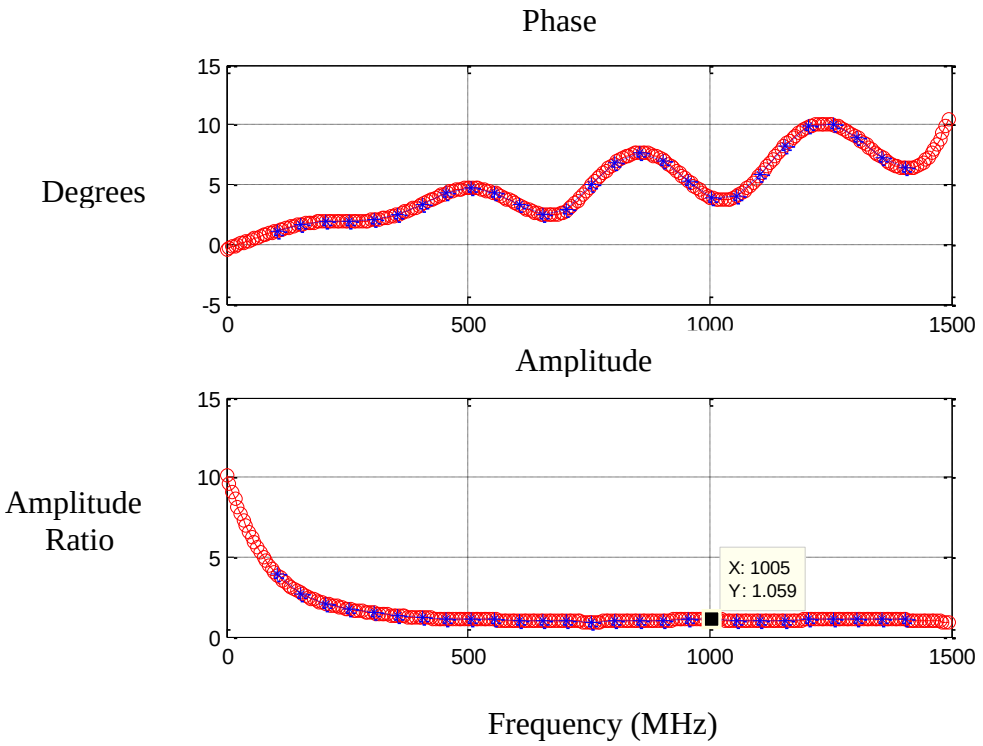


Figure 6-6: Calibration results for coupler: (upper) phase imbalance (bottom) amplitude imbalance.

Once the calibration results for the coupler were obtained, the filter tap coefficients could then be created and the full experiment with all of its equipment could be tested; the setup diagram for this experiment is shown below in Fig. 6-7.

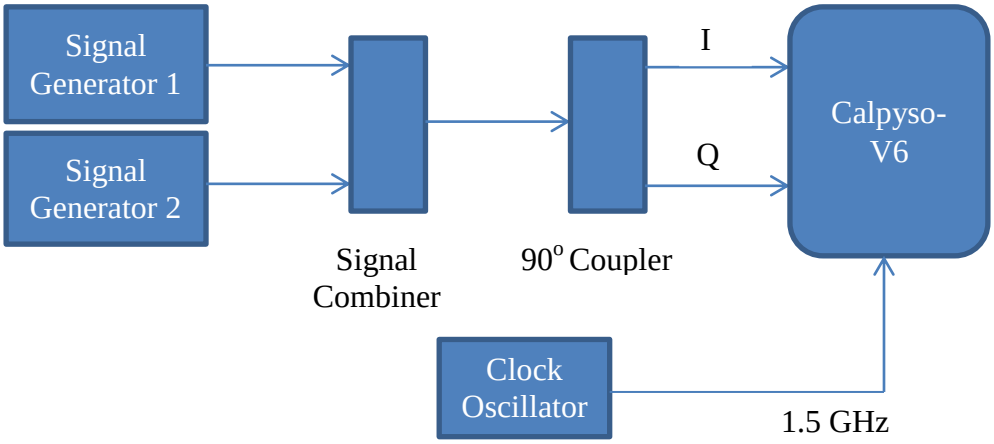


Figure 6-7: Equipment setup for testing.

6.4 Results

Using the equipment setup diagramed in Section 6.3, multiple tests were carried out. There were two main investigations: the feasibility of this method with a real-time hardware implementation and determination of performance under very poor conditions (additional phase and amplitude imbalance added). The first test was to prove the feasibility of the method. The signal frequencies used in experiment are outlined in Fig. 6-8 shown below:

One Tone Test In MHz

RF1	105	205	305	405	505	605	705	805	905	1005	1105	1205	1305	1405

Two Tone Test In MHz

RF1	250	250	250	250	250	250	250	500	500	500	500	500	500	500
RF2	300	500	700	900	1100	1250	1500	200	400	600	800	1000	1200	1400

Two Tone Test (Random Combination) In MHz

RF1	606	1461	504	1274	1317	1317	1368	584	899	759	1300
RF2	931	1410	681	763	763	645	636	1369	1079	1049	644

Fig. 6-8: Signal frequencies used in one-tone and two-tone tests

This initial verification test was split into three separate tests. The first test (one tone test) was to see the performance over the entire working bandwidth. The second verification was the two tone test; the key for the two tone test was that at any given time, there would be two signals in the first Nyquist zone, two signals in the second Nyquist zone, or a signal in both of the two different Nyquist zones. The final verification involved a two tone, random signal; this was to create a more realistic possibility with a receiver detecting a signal of any frequency.

The results of the first part of this test (one tone test) were the first indication that this method was feasible in real-time hardware. These results are shown below in Fig. 6-8.

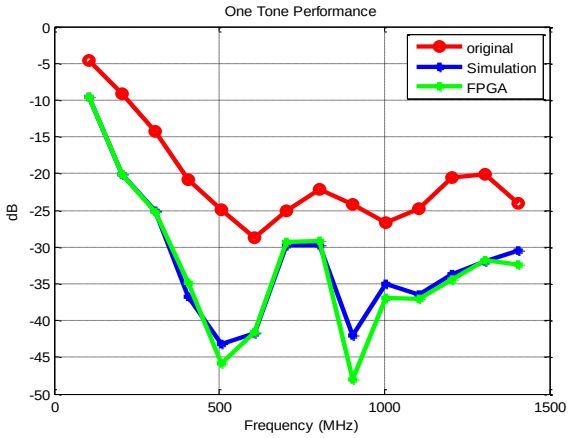
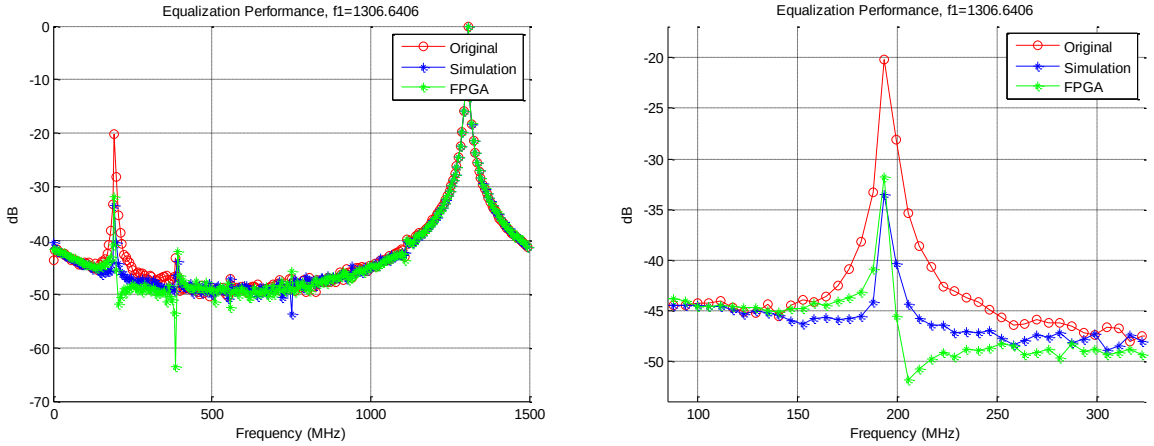


Figure 6-9: One tone equalization performance.



(a) (b)

Figure 6-10: (a) Frequency 1305 equalization performance and (b) the close up of the image peak

As shown in Fig. 6-9, the FPGA results match up very favorably to the simulated results by following a similar pattern of image reduction. The difference between measurement and simulation is probably due the fact that the equalization is implemented in a floating point operation and actual implementation is conducted in fixed-point. Figure 6-10 shows one specific example to visualize how the false peak is lowered after being filtered with the compensation scheme. Figure 6-10 (a) shows one example with the input signal having a frequency of 1305 MHz and Fig. 6-10 (b) shows a zoomed up version of the image peak demonstrating how the specific false image drops.

The next test was the two tone test. Three different sets of data were collected for the two tone test: (1) A signal with a set frequency of 250 MHz and a variable, sweeping frequency, (2) a signal with a set frequency of 500 MHz, and (3) a different set of variable, sweeping frequencies (as shown in Fig. 6-8). One note is with the sampling rate at 1.5 GHz, the Nyquist frequency is located at 750 MHz. Another important note was when the two signals were located where the other signal's image peaks were located; the power of those two signal peaks should not have any reduction in power. The spectra of 2 of the two tone test are shown below in Figs. 6-11 and 6-12 and the complete results are shown in Tables 6-1 and 6-2:

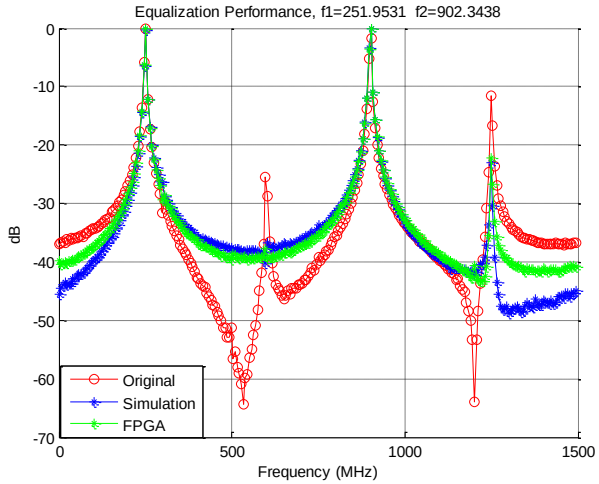


Figure 6-11: Two tone performance (250 MHz and variable frequency of 905 MHz)

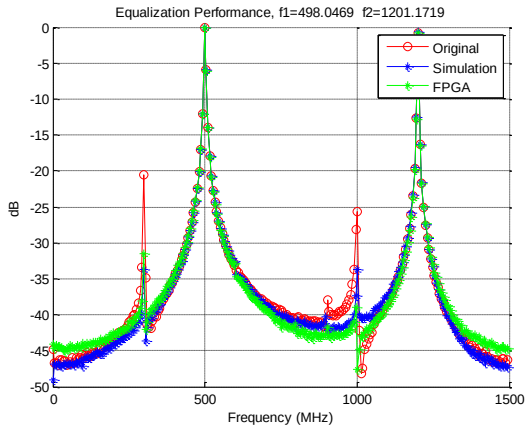


Figure 6-12: Two tone performance (500 MHz and variable frequency of 1205 MHz)

	Case1	Case2	Case 3	Case 4	Case 5	Case 6*	Case 7
RF1 (MHz)	250	250	250	250	250	250	250
Original (dB)	-14	-12	-12	-12	-12	-1	-12
Compensated (dB)	-24.5	-24	-23	-22	-22	0	-22
Difference	10.5	12	11	10	10	-1	10
RF2 (MHz)	300	500	700	900	1100	1250	1500
Original (dB)	-11	-25	-25.5	-25	-25	0	-40
Compensated (dB)	-23	-50	-33	-38	-33	0	-40.5
Difference	12	25	17.5	13	12	0	.5

Table 6-1: 250 MHz and sweeping frequency case performance.

	Case1	Case2	Case 3	Case 4	Case 5*	Case 6	Case 7
RF1 (MHz)	500	500	500	500	500	500	500
Original (dB)	-9	-20	-25	-26	0	-25	-24
Compensated (dB)	-21	-39	-39	-48	0	-47	-35
Difference	11	19	14	22	0	22	11
RF2 (MHz)	200	400	600	800	1000	1200	1400
Original (dB)	26	24	30	24	0	20	24
Compensated (dB)	47	35	39	30	0	31	28.5
Difference	21	11	9	6	0	11	4.5

Table 6-2: 500 MHz and sweeping frequency case performance.

The results from the FPGA again match favorably to the simulation results; all frequencies show an image peak reduction and for the location of the mirroring signal peaks, their respective powers were not reduced. To complete the initial verification test, a two tone, random signal was used as the input signal to test the performance in a more realistic scenario. Figure 6-13 shows the result of one of the cases for the random two tone test and Fig. 6-14 shows the complete performance of all frequencies tested (frequencies are arranged from lowest to highest frequencies for better viewing):

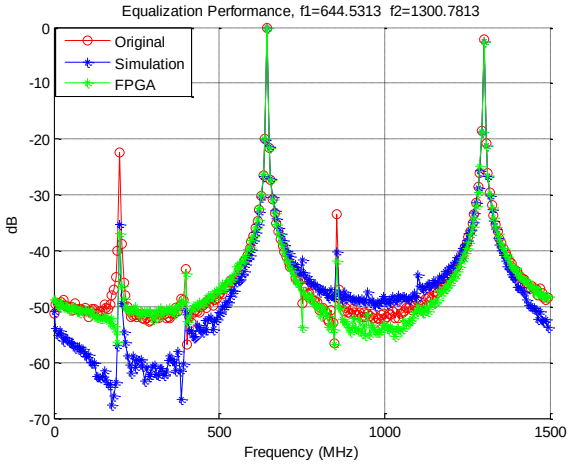
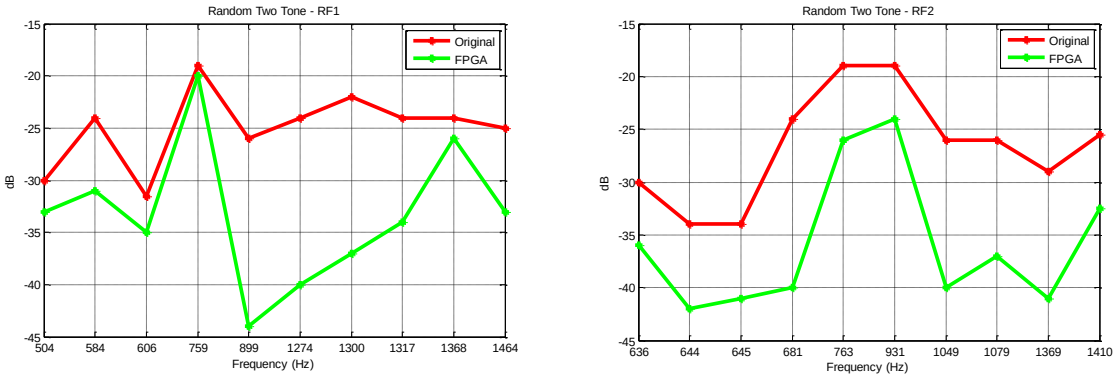


Figure 6-13: Two tone performance (random frequencies)



(a) **(b)**

Figure 6-14: The (a) first and (b) second frequency performance

Again, the FPGA results showing the image drop at all frequencies. In Fig. 6-14 (a), there is no image drop at the frequency of 759 MHz, but this is because that frequency is close to the

folding frequency (750MHz). When the signal is close to the folding frequency, we would like to see the zero dB power drop as seen in Fig. 6-14 (a).

The above figures demonstrate the feasibility of this method with a real-time hardware implementation. However, this is with a coupler that only has a moderate phase and amplitude imbalance; it is of interest to see how well this compensation method holds up when additional amplitude and phase imbalances are introduced. To test this, six more verification tests were completed. The first three tests included adding three different amplitude attenuators: 1, 2.5, and 4 dB. These attenuators were added to the end of the Q channel of the coupler. To offset the phase difference added by the extra length of the attenuator, a 0 dB attenuator was added to the I-channel to compensate that added phase. For the additional phase, connectors were used to lengthen the Q-channel to have a length slightly greater than the I-channel. The summary of the 6 different imbalance profiles is split into two separate tables since the unspecified region of the coupler of DC to 500 MHz has such an extreme imbalance profile. These profiles are shown in Tables 6-3 and 6-4 below:

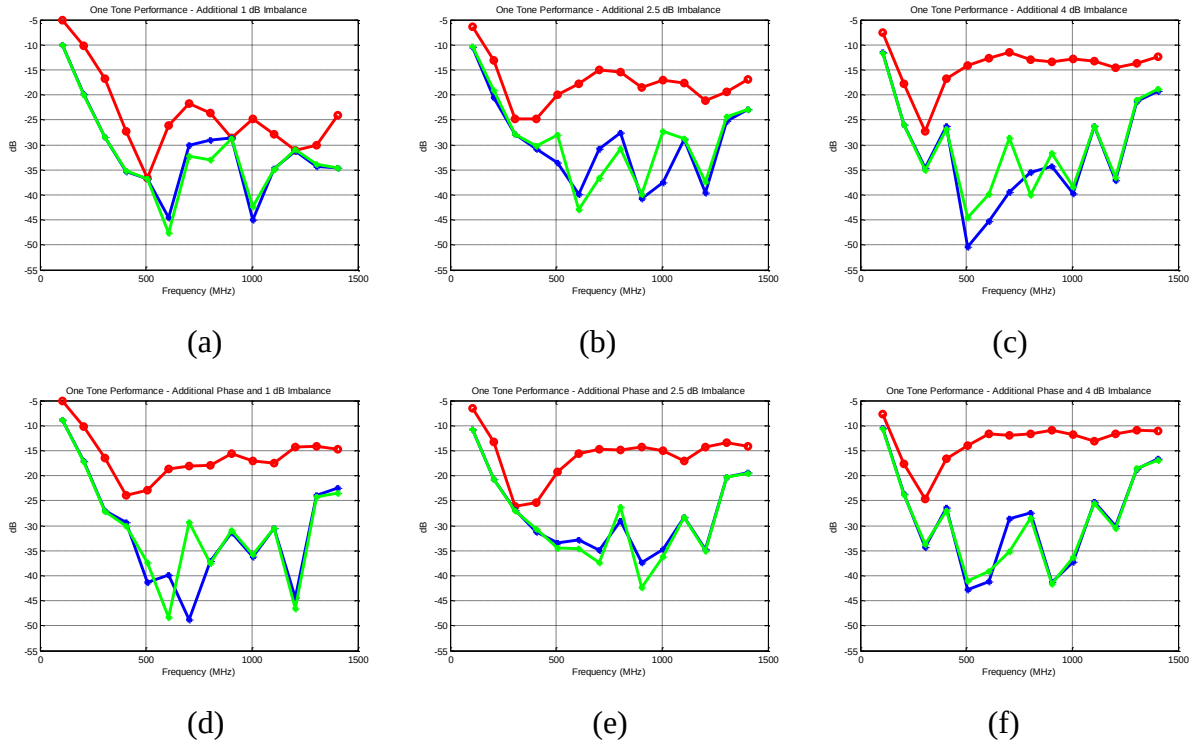
Added Imbalance	Min. Phase	Max Phase	Min. Amp. Ratio	Max Amp. Ratio
1dB	0°	1.5°	10	.92
2.5 dB	0°	1°	8	.74
4 dB	0°	1.5°	6	.64
1 dB + phase	0°	5°	9.5	.89
2.5 dB + phase	0°	8°	8	.72
4 dB + phase	0°	8°	6	.60

Table 6-3: Imbalance profile from DC – 500 MHz

Added Imbalance	Min. Phase	Max Phase	Min. Amp. Ratio	Max Amp. Ratio
1dB	1°	5°	.91	.93
2.5 dB	1°	6°	.74	.75
4 dB	1°	5°	.62	.64
1 dB + phase	8°	25°	.88	.89
2.5 dB + phase	8°	25°	.72	.73
4 dB + phase	8°	25°	.60	.61

Table 6-4: Imbalance profile from 500 MHz – 2000 MHz

Based on these different phase and amplitude imbalance profiles, a new set of coefficients were created for each of the six different cases. Below, Fig. 6-15 shows the performance of the compensation method for the 6 different imbalance profiles:



(a) (b) (c)

(d) (e) (f)

Figure 6-15: One tone performance with added imbalance cases: (a) 1 dB, (b) 2.5 dB, (c) 4 dB, (d) 1 dB plus phase, (e) 2.5 dB plus phase, and (f) 4 dB plus phase.

The outcome of the single tone test, as shown in Fig 6-15, shows that the method does work for couplers with more significant phase and amplitude imbalances. It is interesting to note that the effectiveness increases as the added imbalance increases; the compensation method will lower each of these imbalance cases to around the same level, however when the imbalance is greatly increased, the false image peak is also increased and the image drop will be greater. For example, the average false image power for the added imbalance of 1 dB, 2.5 dB, and 4 dB increase from –23.93 dB, –17.7 dB, and –14.64 dB respectively; this is shown in Fig. 6-14 (a), (b), and (c).

In this chapter, we detailed the implementation of the proposed equalization scheme with a poly-phase filter approach in hardware. The experimental results demonstrate the effectiveness and feasibility of the proposed imbalance compensation method.

Chapter 7

Conclusion and Future Work

In this thesis, two main goals were reached. First, a PDW generator was created. This PDW generator included capabilities to detect multiple types of incoming signals. These signals could consist of CW, chirping, or BPSK signals. Once one or more of these signals were detected, the characteristics of the signal were accurately summarized in form of a PDW. The second goal that was reached was the hardware implementation of a compensation method to reduce the false image produced by the imperfect 90 degree coupler. The method was verified to be feasible in hardware and multiple tests were conducted to show its effectiveness. This was the first indication that this method for increasing the working bandwidth from traditional receivers was feasible.

In the future, there are two possible goals. For the PDW generator, a method to determine the chip time for BPSK signals needs to be devised. This would include determining a more precise way (other than the double phase difference method) to determine the existence of a phase change within a frame of data. This would allow for a more complete PDW generator.
The second objective includes work on the hardware implementation of the IQ compensation scheme. Presently, to obtain the coefficients to reduce the false image, the filter coefficients must be pre-processed manually. It would be advantageous to create these coefficients automatically within hardware by performing the coupler calibration on FPGA board. This would alleviate the issue when the phase and amplitude imbalance changes, causing the results to sometimes be erroneous.

References

[1] J. Tsui, “Introduction”, in *Digital Techniques for Wideband Receivers*, Artech House, 1995, ch. 1, pp. 1-20.

[2] R. Wiley, “Characteristics of ELINT Interception Systems” in *ELINT The Interception and Analysis of Radar Signals*, Artech House, 2006, ch. 3, pp. 29-38.

[3] J. Tsui, “Discrete Fourier Transform”, in *Digital Techniques for Wideband Receivers*, Artech House, 1995, ch. 4, pp. 77-102

[4] F.J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proceedings of the IEEE* , vol.66, no.1, pp. 51- 83, Jan. 1978

[5] D. Maurer, R. Chamlou, and K. Genovese, “Signal Processing Algorithms for Electronic Combat Receiver Applications,” Johns Hopkins APL Technical Digest, vol. 18, no. 1, pp. 69-78, 1997.

[6] M. Sanchez, M. Garrido, M. Lopez-Vallejo, and J. Grajal, “Implementing FFT-based digital channelized receivers on FPGA platforms,” IEEE Transactions on Aerospace and Electronic Systems, vol.44, no.4, pp.1567-1585, Oct. 2008.

[7] J.-M. Zhang, S.-F Cao, and S.-X. Gao, "Long-data DFT and its application in radar reconnaissance receiver," the Proc. of 8th International Symposium on Antennas, Propagation and EM Theory (ISAPE 2008), pp.1450-1453, Nov. 2-5, 2008.

[8] P. Li, Y. Zhang, and Z. Chen, "Design and implementation of a wideband Digital reconnaissance receiver,” the Proc. of 2009 IET International Radar Conference, April 20-22, 2009.

[9] J. Tsui, “Signal Detection From Fast Fourier Transform (FFT) Outputs”, in *Special Design Topics in Digital Wideband Receivers,* Artech House, 2010, ch.6, pp. 109 - 120.

[10] J. Tsui, “Frequency Modulated (FM) Signals”, in *Special Design Topics in Digital WidebandReceivers,* Artech House, 2010, ch.15, pp. 349 - 352.

[11] L. Hang, K. Ho., "BPSK and QPSK modulation classification with unknown signal level ," *MILCOM 2000. 21st Century Military Communications Conference Proceedings*, vol.2, no., pp.976-980 vol.2, 2000

[12] J. Tsui, “Detection of Biphase Shift Keying (BPSK) Signals”, in *Special Design Topics in Digital Wideband Receivers,* Artech House, 2010, ch.14, pp. 343 - 347.

[13] J. Pennington, “Radar Signal Characteristic Extraction with FFT-Based Techniques*”*, Master thesis, Miami University, 2011, Retrieved from https://etd.ohiolink.edu/.

[14] M. Valkama., M. Renfors, V. Koivunen, “Advanced methods for I/Q Imbalance Compensation in Communication Receivers”, Signal Processing Filter for FPGA Implementation, IEEE, vol. 49, no.10, pp. 2335-2344, Oct 2001

[15] J.P.F. Gloas, “Digital I/Q Imbalance compensation in a Low-IF Reciever”, Global Telecommunications Conference, 1998. GLOBECOM 1998. The Bridge to Global Integration. IEEE, vol.3, no.,pp. 1461-1466 vol. 3, 1998

[16] J. Tubbax, B. Come, L. Van der Perre, L. Deneire, S. Donnay, and M. Engels, “Compensation of IQ imbalance in OFDM Systems”, Proc. IEEE Intl. Communications, 2003, pp. 3403-3407.

[17] M. Windisch, G. Fettweis, “Standard-Independent I/Q Imbalance Compensation in OFDM Direct-Conversion Receivers”, Proc. 9th Intl. OFDM Workshop, 2004, pp. 57-61.

[18] K.-H. Lin, H.-L. Lin, S.-M. Wang and R. C., Chang, “Implementation of Digital IQ Imbalance Compensation in OFDM WLAN Receivers”, Proc. IEEE ISCAS, 2006, pp. 3534-3537.

[19] J.-Y Yu, M-F Sun, T-Y Hsu, and C-Y Lee, “A Novel Technique for I/Q Imbalance and CFO Compensation in OFDM Systems”, Proc. IEEE ISCAS, 2005, pp. 6030-6033.

[20] M. Mailand, R. Richter, and H-J Jentschel, “IQ-Imbalance and Its Compensation for Non-Ideal Analog Receivers Comprising Frequency-Selective Components”, Adv. Radio Science, 2006, 4, pp. 189-195.

[21] Y. Wang, Z. Chen, C. Van Meersbergen, S. Heinen, "System simulation of adaptive I/Q mismatch compensation method using SystemC-AMS," *Ph.D. Research in Microelectronics and Electronics (PRIME), 2010 Conference on* , vol., no., pp.1,4, 18-21 July 2010

[22] H. Wang, Y. Lu, X. Wang, and C. Wang, “Digital I/Q Imbalance Compensation in Quadrature Receivers”, Proc. CIE Conf. Radar, 2006 pp. 1-4.

[23] K.-P. Pun, J. Franca, and C. Azeredo-Leme, “Wideband Digital Correction of I and Q Mismatch in Quadrature Radio Receivers”, Proc. IEEE ISCAS, 2000, pp.661-664.

[24] C. Cheng, L. Liou, D. Lin, J. Tsui, and H. Tai, "Wideband I/Q Imbalance Compensation Using FIR Filter", submitted for publication in IET Radar, Sonar & Navigation.

[25] M. Longbrake, "Derivation of Parallel Polyphase Filter for FPGA Implementation", unpublished.