2015

Modern Digital Chirp Receiver: Theory, Design and System Integration

Stephen Ray Benson
Wright State University

Modern Digital Chirp Receiver: Theory, Design and System Integration

A dissertation submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

By

Stephen Ray Benson

B.S. Computer Engineering, Wright State University, 2008

M.S. Electrical Engineering, Wright State University, 2010

2015

Wright State University

WRIGHT STATE UNIVERSITY

GRADUATE SCHOOL

December 16, 2015

I HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER MY SUPERVISION BY Stephen Ray Benson ENTITLED Digital Chirp Receiver: Theory, Design and System Integration BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Doctor of Philosophy.

_____
Chien-In Henry Chen, Ph.D.
Dissertation Director

_____
Dr. Ramana V. Grandhi, Ph.D.
Director, Ph.D. in Engineering Program

_____
Robert E. W. Fyffe, Ph.D.
Vice President for Research and
Dean of the Graduate School

Committee on
Final Examination

_____
Chien-In Henry Chen, Ph.D.

_____
Marty Emmert, Ph.D.

_____
Marian Kazimierczuk, Ph.D.

_____
Saiyu Ren, Ph.D.

_____
Wen-Ben Jones, Ph.D.

ABSTRACT

Benson, Stephen Ray. Ph.D. in Engineering Program, Department of Electrical Engineering, Wright State University, 2015. Modern Digital Chirp Receiver: Theory, Design and System Integration.

Chirp signals can achieve a high range resolution without sacrificing SNR or maximum range, making them a strong candidate for use in radar and sonar applications. Chirp signals are also power efficient and resistant to interference, making them well suited for communication applications as well.

The proposed digital high chirp rate receivers will showcase the use of digital instantaneous frequency measurement (IFM) devices for high chirp rate measurement. The receivers are paired with a high resolution time-of-arrival algorithm, capable of detecting the TOA and TOD of a pulse with an average error of less than 2ns. The high resolution pulse detector is vital for the measurement of high chirp rate, short pulse duration chirp signals when no a priori knowledge of the signals or operating environment is available. Three different receivers were designed and implemented in order to target three different applications: linear chirp signals, nonlinear chirp signals, and linear chirp signals with varying pulse widths. In addition to the digital IFM and TOA algorithm, a high rate 43-tap Hilbert Transform was implemented via an FIR filter in order to convert incoming real data from the ADC into its complex signal representation.

All three designs were synthesized and successfully tested on a Xilinx Virtex 6 SX475 FPGA which is paired with a Calypso 12-bit ADC sampling at 2.56GHz. All three receivers run

at a rate of 320MHz and can measure chirp rates up to 1180MHz in 400ns. The designs boast an overall detection rate of greater than 97% with a false alarm rate of 10^{-7} and achieve a frequency measurement error of less than 1% for both chirp rates and carrier frequencies. The receivers can successfully detect and measure chirp signals and stationary carrier frequencies with SNRs 5dB and higher. The largest design, the digital nonlinear chirp receiver only utilizes 13% of the Virtex 6 SX475 FPGA board.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ADC	Analog to Digital Converter
ASIC	Application-Specific Integrated Circuit
CW	Continuous Waves
DSP	Digital Signal Processing
FFT	Fast Fourier Transform
FIFO	First In First Out
FPGA	Field Programmable Gate Array
FT	Fourier Transform
HDL	Hardware Description Language
LIFO	Last In First Out
MIMO	Multiple In Multiple Out
PRF	Pulse Repetition Frequency
PRI	Pulse Repetition Interval
PW	Pulsed Waves
RF	Radio Frequency
SNR	Signal-to-Noise Ratio
VHDL	Very-High-Speed Integrated Circuit Hardware Description Language
XSG	Xilinx System Generator

Acknowledgements

I would like to thank Dr. Henry Chen for his unwavering support throughout this research. He has been my greatest motivator and a wonderful mentor. His vision throughout this process helped keep the research on track and overcome all of the adversity that was faced. Most importantly, Dr. Chen has always been a great friend and has affected so many aspects of my life. Not only has he made me a better researcher, but he has made me a better person.

I would also like to thank David Lin for all of his guidance as well as the creation of this research topic. We spent numerous hours discussing the functionality of digital chirp receivers, his experience and wisdom was invaluable.

This research could not have been possible without the support from DAGSI and Dr. Elizabeth Downie. I would personally like to thank Dr. Downie for her support and flexibility, allowing this research to continue while overcoming many hurdles. I would also like to express my gratitude to the team at AFRL, including Thomas Pemberton, Dr. Lee Liou, Peter Buxa, Dr. Steve Hary, Luis Concha, Mathew Longbrake, and Tom Dalrymple. Beyond allowing me the use of their facilities and equipment, they graciously donated their time to aid my research.

I would also like to take the time to thank my parents, for their constant encouragement. They have made countless sacrifices to ensure that I could further my education, and for that, I will be eternally grateful. Without their love and support, I would not be the person that I am today.

Additionally, I would like to thank Ethan Lin and George Lee who have been wonderful to work beside. They are both wonderful researchers from whom I learned a great deal.

Furthermore I would like to recognize and thank Dr. Marty Emmert, Dr. Marian Kazimierczuk and Dr. Saiyu Ren, and Dr. Wen-Ben Jones for their willingness to serve on my dissertation committee.

I. INTRODUCTION

1.1 Motivation

The range resolution of radar is defined as the minimum distance between two objects required for the radar to be able to distinguish between the two objects. If the distance between two objects is smaller than the radar's range resolution, the two objects will appear as a single object to the radar. Before the discovery of chirp signals, the length of the transmitted pulse determined the range resolution capabilities of radar systems. If a very small range resolution was desired, a very short pulse was necessary. Reducing the pulse length has a major drawback, however, as it effectively reduces the SNR of the signal being transmitted if additional power is not supplied to the transmitter. Reducing the power lowers the effective range that the radar can operate. This relationship creates a tradeoff between range resolution and the maximum range of the radar. The range resolution of an unmodulated signal is given by the following equation:

$$\text{Unmodulated Signal Range Resolution} = \frac{c_0 * \tau}{2} \quad (1)$$

where c_0 represents the speed of the wave, and τ represents the length of the transmitted pulse [1]. This range resolution limitation can be overcome, however, by increasing the bandwidth of the transmitted signal. The range

resolution of a frequency modulated signal is given by the following equation:

$$\text{Range Resolution Frequency Modulated Signal} = \frac{c_0}{2 * B} \quad (2)$$

We can see from the equation that the range resolution is no longer dependent on the length of the signal, but by the bandwidth of the transmitted pulse [1]. This allows the use of long duration pulses, which can maximize SNR while still achieving a very fine range resolution. Fig. 1 plot (a) shows an unmodulated 50MHz signal with a pulse duration of 400ns while Fig. 1 plot (b) shows the output of the signal correlated with its matched filter. As expected, the signal has a peak, relative to the maximum correlation between the signal and the filter. We can see that the output from the matched filter is very wide, alluding to its inefficiency at detecting closely spaced objects.

Fig. 1 plot (c) shows a linear chirp signal with a chirp rate of 50MHz in 400ns and a starting frequency of 50MHz, with a pulse duration of 400ns. A chirp signal is defined as a signal whose frequency changes over a period of time. We can see in Fig. 1 plot (d) that the peak output of the matched filter still occurs at the same point, however the width of the output has been significantly decreased. The width of the matched filter can be further reduced if the bandwidth of the signal is increased. Fig. 1 plot (e) shows a linear chirp signal with a chirp rate of 1180MHz in 400ns and a starting frequency of 50MHz with a pulse duration of 400ns. Fig. 1 plot (f) shows the output from the matched filter. We can see the output from the matched filter has a very narrow peak, allowing for the detection of objects which are very close to one another.

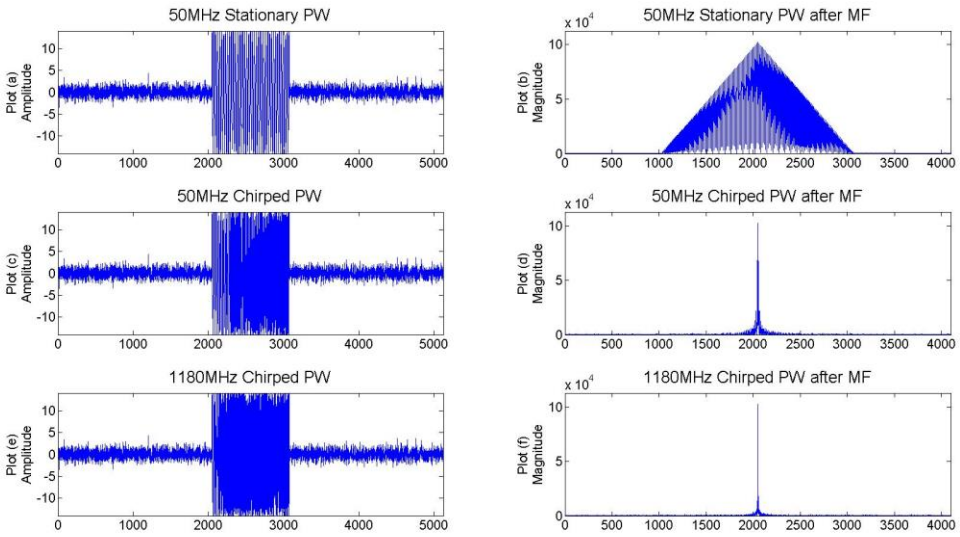

Figure 1 - Chirp rate vs. matched filter output.

A simple example can be used to visualize the range resolution performance of these three signals. If a radar pulse is transmitted and reflected off of two objects, which are very close to one another spatially, the reflected two pulses from the two objects will be received by the receiver separated by a very short delay. Depending on the spatial distance between the objects, it is possible for the reflected pulses to overlap one another. Fig. 2 plot (a) shows the two unmodulated received pulses, while plot (b) shows the output from the matched filter (MF) correlated with these two signals. It can be seen from the plot that the two objects become ambiguous and cannot be distinguished between one another. Plot (c) shows two frequency modulated signals with a bandwidth of 50MHz and plot (d) shows the output of these signals correlated with their matched filter. The two objects are now clearly distinguishable from one another with no ambiguities. As expected, plot (e) shows two frequency modulated signals with a bandwidth of 1180MHz and plot (f) shows the resulting output from the signals correlated with their matched

filter. The further increase in signal bandwidth allows for a much finer range resolution as well as a suppression of the unmatched correlations.

In addition to typical radar applications, chirp signals have also been used for measuring the thickness of snow ice [2], in automotive ranging applications [3], multiple input multiple output (MIMO) radars [4], short-distance detecting and imaging systems [5], and multiple target localization [6]. In fact, the use of chirp signals has spread beyond just radar systems and are becoming widely used in communication systems [7, 8, 9].

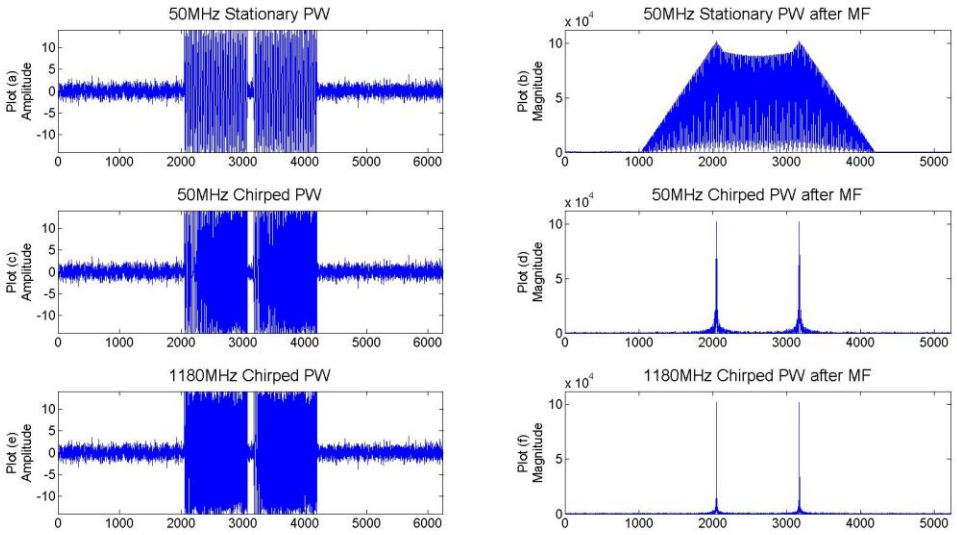

Figure 2 - Chirp rate vs. matched filter output for closely spaced targets.

1.2 Chirp Signal Measurement

Chirp signals can have linear or nonlinear changes in frequency. If the chirp rate is small relative to the measurement period, a traditional fast Fourier transform (FFT) or instantaneous frequency measurement (IFM) device can be used over multiple measurement periods to estimate the chirp rate through slope calculation [10]. However, this method is impractical for use with high chirp rates, which change significantly during the measurement period.

Of course other methods have been used to measure chirp rates, including a Wigner distribution, which has been used to detect non-stationary phase-modulated signals [11, 12]. By computing the line integral of the Wigner distribution of a linear frequency modulated signal along all lines in the time–frequency plane, it is found that the line that produces the maximum value corresponds to the maximum likelihood of the linear instantaneous frequency of the chirp. The Radon–Wigner transform [13] converts the problem of tracking straight lines in the time–frequency plane into locating maxima in an initial frequency versus chirp rate two-dimensional plane. Radon-ambiguity transform [14] combines Radon-Wigner transform and the ambiguity function to improve the detection of linear frequency modulated signals by searching the location of maxima over chirp rates only. Another approach using fractional autocorrelation for parameter estimation of linear frequency modulated signals and chirp rates was proposed [15]. Hough transform was proposed to reduce computational load to detect long and very slow chirp signal, i.e., 1-Hz drop in a 10-hour interval [16]. A state dependent differential Riccati equation (SDDRE) based estimator was proposed in [17]. Mathematical models proposed by these approaches require large computation. No hardware implementation or chirp rate range was discussed or tested.

Chirp rates can be classified as up or down chirps, i.e. the frequency increases or decreases over time. The research presented in this dissertation will focus on up chirps; however, the techniques shown are also useable with down chirps.

Because the frequency parameters of chirp signals do not remain constant during the length of the pulse, parameter estimation can be a difficult task and is reliant on multiple factors. The speed at which the signal changes relative to the sampling

frequency has the biggest effect on measurement performance. As the chirp rate increases, the sampling frequency must also be increased to ensure a proper Nyquist sampling rate as well as provide an adequate number of samples for measurement.

Chirp signal measurement is not limited to the use of a single type of measurement device; however, the type of device chosen will dictate the temporal and frequency resolution limits. The FFT is widely used in the signal processing community and is a powerful tool due to its computational efficiency [18]. The FFT is used to convert time domain data into the frequency domain. The FFT excels at detecting multiple simultaneous signals as it provides a snapshot of the entire spectrum. The size of the FFT determines its computational complexity, latency, frequency resolution and temporal resolution. As the size of the FFT increases, the frequency resolution is improved. However, the trade-off is a reduction in temporal resolution, with an increase in complexity and latency.

The computational complexity of the FFT grows at the rate of O(N$log_2 N$) while the frequency resolution grows at a rate of $\frac{F_s}{N}$, where N is the number of data samples collected (size of the FFT) and F_s is the sampling rate [19]. If the signal of interest has a short pulse duration, then the sampling frequency must be high in order to obtain enough samples to achieve the desired frequency resolution. This has practical limits as high sample rate ADCs are expensive and are limited by current technology.

For these reasons, it is difficult to achieve a fine frequency resolution when measuring short pulsed waveforms with an FFT. The process used to measure chirp signals further reduces the number of samples available to the FFT. The number of available samples is reduced due to the need to mix the chirp signal with a delayed and

conjugated copy of itself in order to isolate the higher order frequency components. The sensitivity of the chirp measurement is increased as this delay is increased. However, every sample of delay used reduces the number of available samples for measurement. If an FFT is used for measurement, a large delay will increase the sensitivity of the measurement, however a reduction in samples will reduce the frequency resolution of the FFT.

If a chirp signal contains multiple chirp parameters; they can be accurately measured by isolating and removing the highest order parameter present until only the carrier frequency is left. Fig. 3 will be used to visualize the process for isolating each frequency component. Fig. 3 plot (a) shows the FFT plot of a chirp signal which contains a 780MHz in $400ns^2$ nonlinear chirp rate, a 390MHz in 400ns linear chirp rate and a starting frequency of 50MHz.

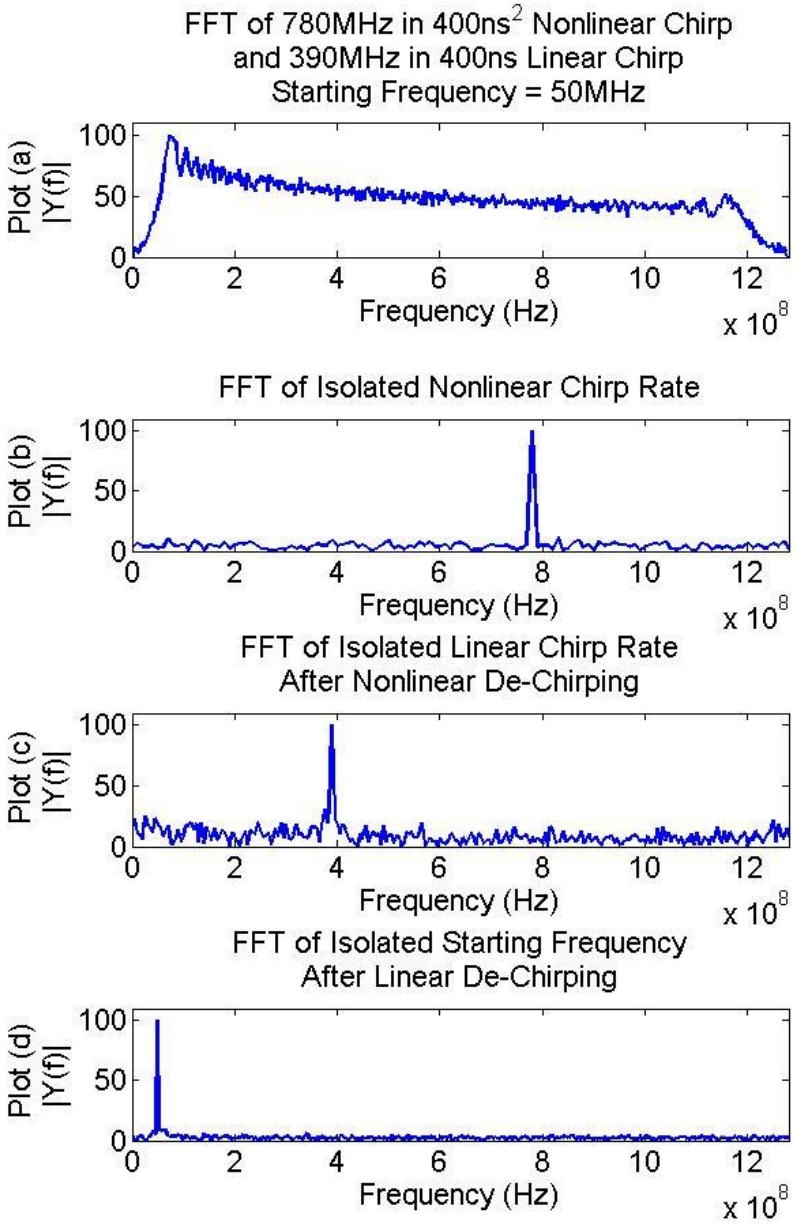

Figure 3 - Nonlinear chirp signal measurement using an FFT.

The nonlinear frequency component of a chirp signal can be isolated by utilizing two separate mixing processes. First, the original signal needs to be mixed with a delayed and conjugated copy of itself. This resulting signal needs to be mixed with a delayed and conjugated copy of itself once more. The optimal delays used for these mixing processes are determined by the pulse width of the signal and the measurement period of the device. For an FFT, the measurement period of the device is chosen based on the desired frequency and temporal resolution. If a 512-point FFT is chosen and a sampling rate of 2.56GHz is used, then the measurement period will be 200ns. The target pulse width of chirp signals in this research is 400ns, therefore the optimal delay will be (400ns – 200ns = 200ns). Since two separate mixing processes are used, the delay must be split between the two processes. The optimal ratio for this split is a 1:2 ratio. The delay for the second mixing process should be twice as long as the delay used for the first mixing process. Therefore, the delay for the first mixing process becomes $\frac{1}{3} * 200ns = 66.66ns$ and the delay for the second mixing process is $\frac{2}{3} * 200ns = 133.33ns$. Fig. 3 plot (b) shows the isolation of the nonlinear chirp component after the signal has undergone both mixing processes.

Once the nonlinear chirp rate has been determined, it is possible to remove the nonlinear chirp rate by mixing the signal with a de-chirping signal. The de-chirping signal should have the same ramp rate, but in the opposing direction. The de-chirping signal is mixed with the original signal to remove the nonlinear chirp rate. After nonlinear de-chirping, the linear chirp component can be measured. The linear chirp component is isolated by mixing the de-chirped signal with a delayed and conjugated copy of itself. The delay value used for this process will be the same as the one used in

the nonlinear chirp isolation process (200ns). Fig. 3 plot (c) shows the isolation of the linear chirp rate after the nonlinear chirp rate has been removed.

Once the linear chirp rate has been measured, it can be removed from the signal by mixing it with an appropriate linear de-chirping signal. The linear de-chirping signal will have the same ramp rate, but in the opposing direction. The de-chirping signal will be mixed with the nonlinear de-chirped signal to remove the linear chirp rate. After all chirp components have been removed from the signal, only the carrier frequency (starting frequency) will remain. The FFT plot of the carrier frequency is shown in Fig. 3 plot (d).

Chirp parameter isolation and measurement can be shown through mathematical analysis. A digitized nonlinear chirp signal can be represented by

$$x(n\Delta t) = ae^{j(\omega n\Delta t+\left(\frac{1}{2}\right)\alpha n^2\Delta t^2+\left(\frac{1}{3}\right)\beta n^3\Delta t^3)} \tag{3}$$

where ω represents the carrier frequency (starting frequency), α represents the linear chirp rate, and β represents the nonlinear chirp rate. The sample number will be represented by n and the sample period will be represented by Δt.A delayed and conjugated copy of the signal can be represented by

$$x((n-m)\Delta t)' = a'e^{-j(\omega(n-m)\Delta t+\left(\frac{1}{2}\right)\alpha(n-m)^2\Delta t^2+\left(\frac{1}{3}\right)\beta(n-m)^3\Delta t^3)} \tag{4}$$

where m represents the number of sample delays used. If the two sequences are mixed they become

$$S1(n\Delta t) = x(n\Delta t) * x\big((n-m)\Delta t\big)'$$

$$= a * a' * e^{\begin{pmatrix} j(m\omega\Delta t+\ \alpha nm\Delta t^2-0.5am^2\Delta t^2+\ \beta n^2m\Delta t^3- \\ \beta nm^2\Delta t^3+\left(\frac{1}{3}\right)\beta m^3\Delta t^3) \end{pmatrix}} \tag{5}$$

Each parameter in Eq. (5) can be grouped according to its power, and the following substitutions can be made

$$\theta'_1 = \omega m\Delta t - 0.5\alpha m^2\Delta t^2 + \left(\frac{1}{3}\right)\beta m^3\Delta t^3$$

$$\omega'_1 = \alpha m\Delta t - \beta m^2\Delta t^2$$

$$\alpha'_1 = \beta m\Delta t$$

After substitutions, Eq. (5) becomes

$$S1(n\Delta t) = |a^2|\, e^{j(\theta'_1 + \omega'_1 n\Delta t + \alpha'_1 n^2\Delta t^2)} \tag{6}$$

Eq. (6) no longer contains a 3rd order term. We can therefore mix this equation with a delayed and conjugated copy once more, allowing for the removal of the second order term. The resulting signal will contain only the original nonlinear chirp rate, but it will be represented as a first order term. It is at this stage that a typical FFT or IFM device can be used to measure the nonlinear chirp rate. Eq. (7) shows the 2nd mixing process required to isolate the nonlinear chirp rate.

$$S2(n\Delta t) = S1(n\Delta t) * S1\big((n-m)\Delta t\big)'$$

$$= |a^2| * |a^2|' * e^{\left(j(m\Delta t\omega'_1 + \alpha'_1 nm\Delta t^2 - 0.5\alpha'_1 m^2\Delta t^2\right)} \tag{7}$$

Once again grouping all parameters of the same power, the following substitutions can then be made

$$\theta_2 = m\Delta t\omega'_1 - 0.5\alpha'_1 m^2\Delta t^2 \tag{8}$$

$$\omega_2 = \alpha'_1 m\Delta t \tag{9}$$

So Eq. (7) then becomes

$$S2(n\Delta t) = |a|^4 e^{j(\theta_2 + \omega_2 n\Delta t)}$$

$$= |a|^4 e^{j(\theta_2 + \beta m\Delta t * m\Delta t * n\Delta t)} \quad (10)$$

Since θ_2 is a constant, Eq. (10) only contains the nonlinear chirp parameter, represented as a first order term. This makes it easily measured by an FFT or IFM. Once the chirp rate has been determined, a de-chirping signal can be generated and mixed with the original signal of Eq. (3)

$$D1(n\Delta t) = a\, e^{j\left(\omega n\Delta t + \frac{1}{2}\alpha n^2\Delta t^2 + \frac{1}{3}\beta n^3\Delta t^3\right)} * e^{-j\left(\frac{1}{3}\beta n^3\Delta t^3\right)}$$

$$= a\, e^{j\left(\omega n\Delta t + \frac{1}{2}\alpha n^2\Delta t^2\right)} \quad (11)$$

Eq. (11) is now in the same format as Eq. (6) and therefore the same procedure used for Eq. (6) can be utilized again. Eq. (11) will be mixed with a delayed and conjugated copy of itself to yield

$$S3(n\Delta t) = D1(n\Delta t) * D1\big((n-m)\Delta t\big)'$$

$$= a * a' * e^{\left(m\Delta t\omega + \alpha n m\Delta t^2 - 0.5\alpha m^2\Delta t^2\right)} \quad (12)$$

$$\theta_3 = m\Delta t\omega - 0.5\alpha m^2\Delta t^2 \quad (13)$$

$$\omega_3 = \alpha m\Delta t \quad (14)$$

$$S3 = |a|^2 e^{j(\theta_3 + \omega_3 n\Delta t)} \quad (15)$$

Since θ_3 is a constant, S3 only contains the linear chirp rate as a first order term and can be easily measured with an FFT or IFM. Once the linear chirp rate is measured, the

appropriate linear de-chirping signal can be generated to remove the linear chirp rate from Eq. (11). The de-chirping process is as follows.

$$D2(n\Delta t) = ae^{j\left(\omega n\Delta t+\frac{1}{2}\alpha n^2\Delta t^2\right)} * e^{-j\left(\frac{1}{2}\alpha n^2\Delta t^2\right)}$$

$$= ae^{j(\omega n\Delta t)} \quad (16)$$

Eq. (16) no longer contains any chirp parameters and the carrier frequency can now be easily measured using an FFT or IFM.

1.3 Contribution

The research made possible by DAGSI's sponsorship has resulted in this dissertation as well as multiple models, which simulate the detection and measurement of high chirp rate signals. Along with the simulation models, three VHDL models and FPGA programming files have been created and tested on a Virtex 6 FPGA, allowing for the real-time operation of the digital chirp receivers. The three designs which have been created are as follows:

1. Digital Linear Chirp Receiver: Capable of measuring linear chirp rates ranging from 50MHz in 400ns up to 1180MHz in 400ns and carrier frequencies ranging from 50MHz up to 1230MHz.
2. Digital Nonlinear Chirp Receiver: Capable of measuring nonlinear chirp rates ranging from 50MHz in $400ns^2$ up to 1180MHz in $400ns^2$ as well as linear chirp rates ranging from 50MHz in 400ns up to 1180MHz in 400ns and carrier frequencies ranging from 50MHz up to 1230MHz.

3. Digital Variable Linear Chirp Receiver: This receiver is not limited to a single chirp period, but rather is bound by frequency across a wide range of chirp periods. The fastest chirp rate measureable is 1180MHz in 400ns, while the slowest chirp rate measurable is roughly 50MHz in 1 hour. This receiver is also capable of measuring carrier frequencies ranging from 50MHz up to 1230MHz.

1.4 Overview

This dissertation will begin by describing the development design flow and prototyping hardware for each of the receivers in Section II. The development and performance of the time-of-arrival algorithm used by all three of the chirp receivers will be described in Section III. The design and implementation of the Hilbert Transform will be covered in Section IV. The digital instantaneous frequency measurement (IFM) device and its use for chirp rate measurement will be discussed in Section V. The digital linear chirp receiver design will be presented in Section VI. Section VII will cover the design and implementation of the nonlinear digital chirp receiver. The operation of the variable chirp receiver will be presented in Section VIII. In depth performance evaluations for all three chirp receiver designs will be provided in Section IX. The final chapter will summarize the research presented in this dissertation as well as give insight into how this research could be furthered.

II. Design Flow and Prototyping Hardware

2.1 Design Flow

The flow of design for developing, testing and verifying all the designs in this research followed an identical path. Ideas were developed and tested in Matlab first in order to prove their feasibility. The code was enhanced and optimized using Matlab's built in code profiler. The profiler was able to record the amount of time spent in each function and each line of code, as well as the number of times each function was called throughout a simulation. This provided a thorough analysis on the efficiency, or lack thereof, of the code. The simulation run times were significantly reduced by vectorizing the code. Matlab has been highly optimized for vector and matrix operations, meaning it can perform a function on a vector almost as quickly as it can perform a function on a single value. Vectorizing the code is the process of modifying the code to perform all operations on as many simultaneous elements as possible. This greatly reduces the number of times a function needs to be called and greatly reduces the simulation run times.

Once the Matlab code has been completely optimized and verified, the next step in the development process was transitioning the Matlab code into a design based in Simulink/Xilinx System Generator (XSG). The flow between the two tools was not always a simple conversion. It is possible for Simulink to

implement small portions of Matlab code using a translator known as the “Mcode” block. Not all Matlab functions are useable within a Mcode block, and extra precautions must be taken to ensure proper data types and bit widths are used.

The use of Mcode blocks allows for higher levels of abstraction and easier design implementation, while reducing the amount of control over the system that the user has. Extra care is needed when using Mcode blocks as the designer has minimal control over implementation styles and a limited ability to alleviate timing issues. An additional downside to using Mcode blocks is they are limited to using a single clock rate, which is set by the System Generator token.

In addition to the numerous pre-fabricated blocks, which are provided from Xilinx for use in Simulink, the user can also write his own VHDL/Verilog code and insert it directly into the design through the use of the “Black Box” tool. In comparison with the Mcode block, the Black Box provides a much greater level of control to the designer. The Black Box is capable of handling multiple clock rates, though the implementation of the clock rates is not necessarily straightforward or intuitive. The Simulink/XSG tool does not provide clock input ports for any of the design blocks and this is true for Black Boxes as well. In order for a designer to utilize multiple clock rates within a Black Box, matching ‘clock’ and ‘clock_enable’ input ports must be included in the entity definition in their VHDL code. The ‘clock’ and ‘clock_enable’ ports will not be visible when viewing the block in Simulink, so no physical connection within the design needs to be made. In the configuration file for the Black Box, the user has the ability to choose the rate of for the ‘clock_enable’, which will effectively determine the rate of the clock paired with that ‘clock_enable’.

The majority of the logic for the digital chirp receivers was generated using the pre-fabricated blocks provided within the Xilinx block set, in which there is no simple conversion from Matlab code. Therefore most of the designs were developed by hand using the GUI.

The Simulink/XSG model has access to all the variables in Matlab's workspace, and can use the workspace variables as either inputs or outputs. There are currently no tools that can estimate the hardware usage or check whether the design will meet timing constraints for the target device within Simulink/XSG. To accomplish this, the Simulink/XSG model must first be translated into VHDL/Verilog code. The VHDL/Verilog code can then be synthesized by the Xilinx ISE tool. It is at this stage that the design can be analyzed in regards to hardware usage as well as determine if it will meet the timing constraints. If a design fails to meet the constraints, design changes will need to be made to the original Simulink/XSG model, which will need to be re-translated into VHDL/Verilog and synthesized again. This process can be extremely time consuming as certain timing constraints can be very difficult to meet.

2.2 Prototyping Hardware

The target device for this research is a Virtex 6 SX 475 FPGA paired with a Calypso 12-bit ADC, which was developed by Tekmicro. The ADC is capable of sampling at a rate of up to 3.6GSPS, but was configured to use a sampling rate of 2.56GSPS for all the digital chirp receivers in this dissertation. The Tekmicro device can be seen in Fig. 4.

In order to generate short duration, high chirp rate signals, an arbitrary waveform generator was needed. For this, a Keysight M8190A 12GSa/s arbitrary waveform

generator was used to generate the needed signals and is shown in Fig. 5. The signal generator was supplied with a Matlab based GUI, greatly simplifying the process of signal generation. The large memory size of the signal generator allowed for the storage of up to 1 second's worth of samples to be stored at a time. This was more than sufficient storage to allow the entire frequency range of all the signal types to be tested consecutively.

In order to retrieve the frequency measurement information from digital chirp receivers on the FPGA, the results were written to a set of FIFOs. Once the FIFO's were filled, their data was sent over Ethernet to a local PC for analysis.

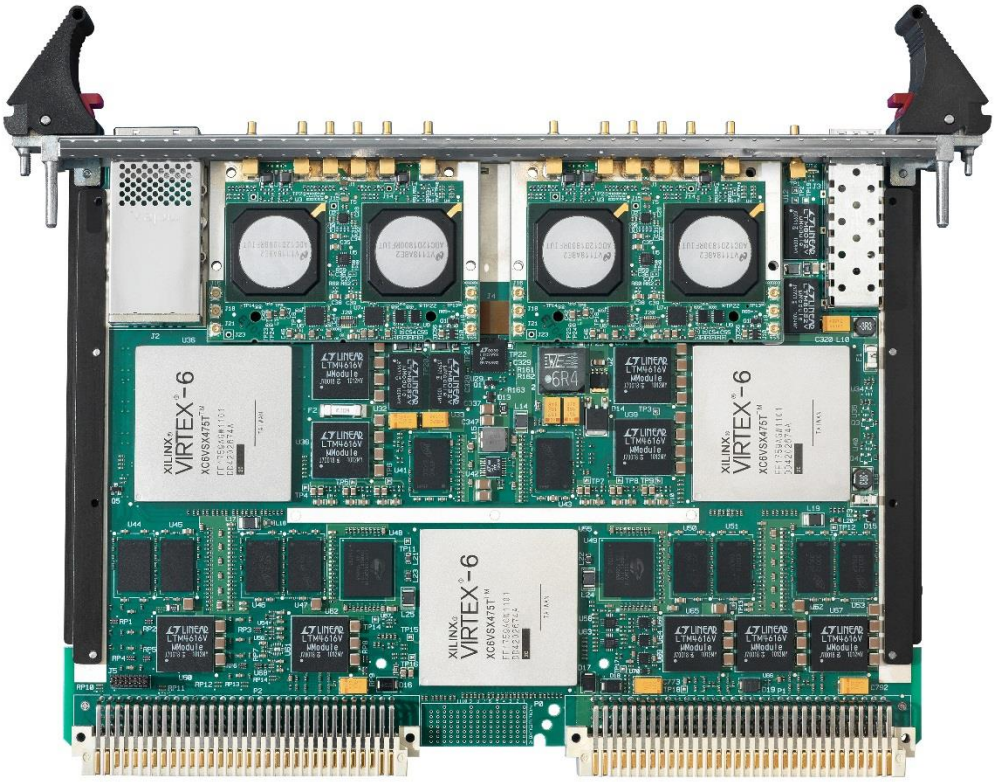

Figure 4 – Calpyso Virtex 6 SX475 FPGA paired with 3.6 GSPS 12-bit ADC.

Figure 5 – Keysight M8190A 12GSPS arbitrary waveform generator.

III. Time-of-Arrival Detection of Chirp Signals

3.1 Motivation

Accurate time of arrival estimation for chirp signals with very high chirp rates is immensely important. The effect of time-of-arrival estimation error on the measurement of chirp signal parameters depends greatly on the type of chirp signal. An exponential nonlinear chirp rate is the most forgiving of TOA estimation error. This is because the slowest change in frequency occurs at the beginning of the signal. However, linear chirp signals are greatly affected by TOA estimation error since the frequency change in the beginning of the signal is substantially higher than its nonlinear counterpart. Fig. 6 plots the instantaneous frequency for a nonlinear chirp signal in plot (a) and a linear chirp signal in plot (b).

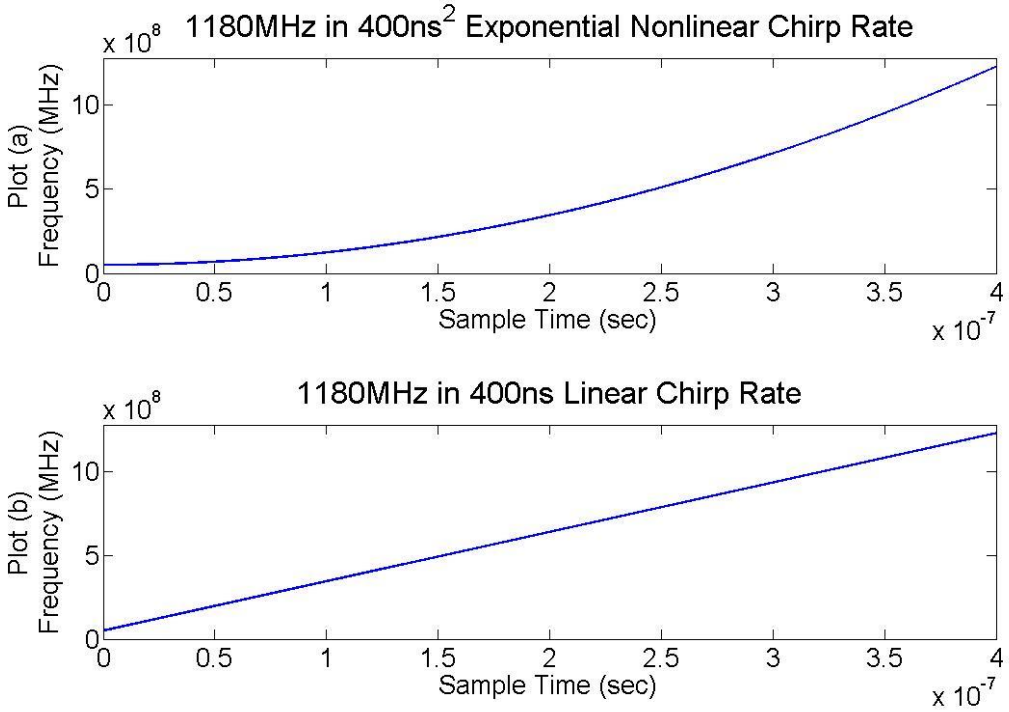

Figure 6 – Instantaneous frequency vs. time for nonlinear and linear chirp signals.

A simple example can be used to highlight the differences between measuring nonlinear chirp signals and linear chirp signals. If very fast chirp rates are considered for both signal types, say a nonlinear signal with a chirp rate of 1180MHz in $400ns^2$ and a linear chirp signal with a chirp rate of 1180MHz in 400ns, we can calculate the expected starting frequency error based on the average expected TOA error. If the detection system being utilized achieves an average TOA error of 5ns, then measuring the starting frequency of the nonlinear chirp rate will encounter an average error of at least 184 KHz. However, when measuring the starting frequency of the linear chirp signal, the system will encounter an average error of at least 14.75MHz. Accurate TOA estimation is one of the greatest challenges when trying to measure high chirp rate signals when no a priori signal or environmental knowledge is available. TOA estimation is not a great concern

when measuring stationary, non-chirp signals because there is no reduction in measurement performance if the signal is detected late. For these reasons, many of the widely used energy detectors and TOA estimators used in the signal processing community are not suitable for use with high chirp rate, short pulse duration signals.

Therefore, a new TOA methodology was studied and developed for use with short duration, high chirp rate signals. All three digital chirp receivers were targeted for real-time implementations on a Virtex 6 SX 475 FPGA. Utilizing a sampling rate of 2.56GHz, the FPGA operating frequency is set at 320MHz since the incoming samples are de-multiplexed at a ratio of 1:8. The requirements for TOA detection were established based on the following principles:

1) Minimum operating frequency of 320MHz
2) Accuracy and resolution capabilities better than ~10ns
3) Detection rate higher than 90%
4) False alarm rate less than 10^{-7} (1 in 10,000,000)
5) Function for SNRs from 5dB up to 20dB
6) Signal detection based on no a priori knowledge

The following sections will cover the investigation and development of the TOA algorithm, which is suitable for use with high chirp rate signals.

3.2 Time-of-Arrival Study for Chirp Signals

Data is sampled at 2.56GHz and is de-multiplexed at a 1:8 ratio, allowing for 8 samples to be provided every clock cycle. Every sample from the ADC is represented by 4 bits, setting an amplitude range of -7 to 7. In order to keep computational complexity at a minimum to allow for the 320MHz operational frequency, simple time-domain amplitude based criteria were studied for TOA detection.

Two of the simplest criteria to consider are mean value and maximum amplitude

based thresholds. Since data is provided in 8-sample chunks, all evaluations are based on evaluating 8 samples in parallel. One of the most difficult tasks for TOA detection is determining the optimal thresholds. The difficulty of this task is further increased by the extremely long simulation times needed to determine the detection rates and false alarm rates for each threshold setting. In order to overcome this hurdle, a statistical model was developed to predict the performance of various threshold settings without the need for lengthy simulations.

In order to develop a useful statistical model, a very large data set was needed, which accurately represents both signals and noise. Remember, no a priori knowledge is available on the incoming signals, so all variables must be accounted for in the data set. The variables which can have a significant impact on signal detection include SNR, starting frequency, chirp rate, and chirp signal type. A data set was created which fully represents all 4 variables and contains the following boundaries:

- Stationary Signals:
    - SNRs: 1dB to 20dB in 1dB increments
    - Carrier Frequencies: 50MHz to 1230MHz in 0.25MHz increments
    - Chirp Rate: N/A
    - Pulse Length: 200ns
    - 200ns * 20 SNRs * 4,720 Frequencies = 18.8ms of samples

- Linear Chirp Signals:
    - SNRs: 1dB to 20dB in 1dB increments
    - Carrier Frequencies: 50MHz to 1180MHz in 0.25MHz increments
    - Chirp Rate: 1180MHz in 400ns down to 50MHz in 400ns in 1MHz decrements
    - Pulse Length: 200ns
    - 200ns * 20 SNRs * 4,520 Frequencies = 18ms of samples

- Nonlinear Chirp Signals:
  - SNRs: 1dB to 20dB in 1dB increments
  - Carrier Frequencies: 50MHz to 1180MHz in 0.25MHz increments
  - Chirp Rate: 1180MHz in 400ns^2 down to 50MHz in 400ns^2 in 1MHz decrements
  - Pulse Length: 200ns
  - 200ns * 20 SNRs * 4,520 Frequencies = 18ms of samples

- Nonlinear and Linear Chirp Signals:
  - SNRs: 1dB to 20dB in 1dB increments
  - Carrier Frequencies: 50MHz to 1130MHz in 0.25MHz increments
  - Chirp Rate: Nonlinear swept from 1130MHz in 400ns^2 down to 50MHz in 400ns^2 in 1MHz decrements and linear swept from 50MHz in 400ns up to 1130MHz in 400ns.
  - Pulse Length: 200ns
  - 200ns * 20 SNRs * 4,320 Frequencies = 17.2ms of samples

The detection rate of signals was estimated by utilizing signals embedded with additive white Gaussian noise (AWGN). A pulse length of 200ns was utilized instead of 400ns in order to reduce the simulation time. The initial target for TOA accuracy was ±10ns, and it was therefore determined unnecessary to simulate the last 200ns of the different signal types. Utilizing these parameters resulted in a total of 72ms of sampled data, or roughly 184 million data samples.

False alarm rates were estimated by creating a data set in which only AWGN was present. Noise was generated in Matlab with zero mean and a standard deviation of 1. Similar to the first data set, 72ms of AWG noise was generated in Matlab.

3.3 Algorithm Development and Threshold Optimization

Once the data sets were created, they were analyzed in 8-sample blocks in which the average of the absolute values as well as the number of occurrences for each magnitude value were recorded for each 8-sample block. This allowed for the creation of cumulative distribution functions at each SNR for the following criteria:

- Criteria 1
    - Average of the absolute values of 8-sample blocks

- Criteria 2
    - Magnitude values which occur at least 1 time in an 8-sample window
    - Magnitude values which occur at least 2 times in an 8-sample window
    - Magnitude values which occur at least 3 times in an 8-sample window
    - Magnitude values which occur at least 4 times in an 8-sample window
    - Magnitude values which occur at least 5 times in an 8-sample window

Since the number of iterations providing the best performance was unknown, numerous versions were tested. Fig. 7 shows the estimated detection rates in plot (a) and false alarm rates in plot (b) for Criteria 1 at an SNR of 7dB for different threshold settings. The dotted vertical lines represent the lowest threshold setting, which meets the false alarm requirement of 10^{-7}. However, the graph shows that this threshold is not able to meet the 90% detection rate at an SNR of 7dB as it only obtains a detection rate of 67.9%.

Fig. 8 shows the estimated detection rates in plot (a) and false alarm rates in plot (b) for Criteria 1 at an SNR of 11dB. The same vertical dotted lines show the lowest threshold setting, which achieves the desired false alarm rate. The higher SNR changes the expected detection rate dramatically, as nearly a 100% detection rate is predicted.

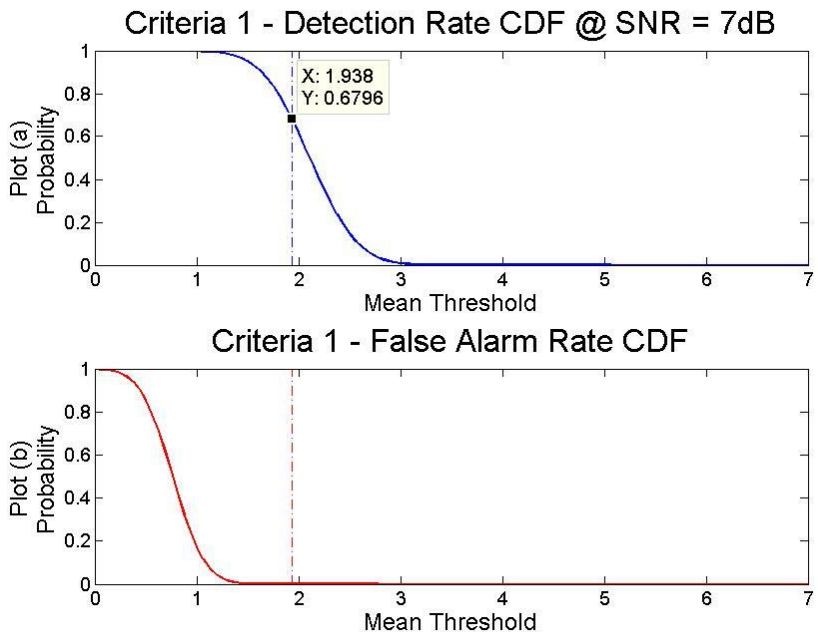

Figure 7 – Cumulative Distribution Function for TOA criteria 1 @ SNR = 7dB.

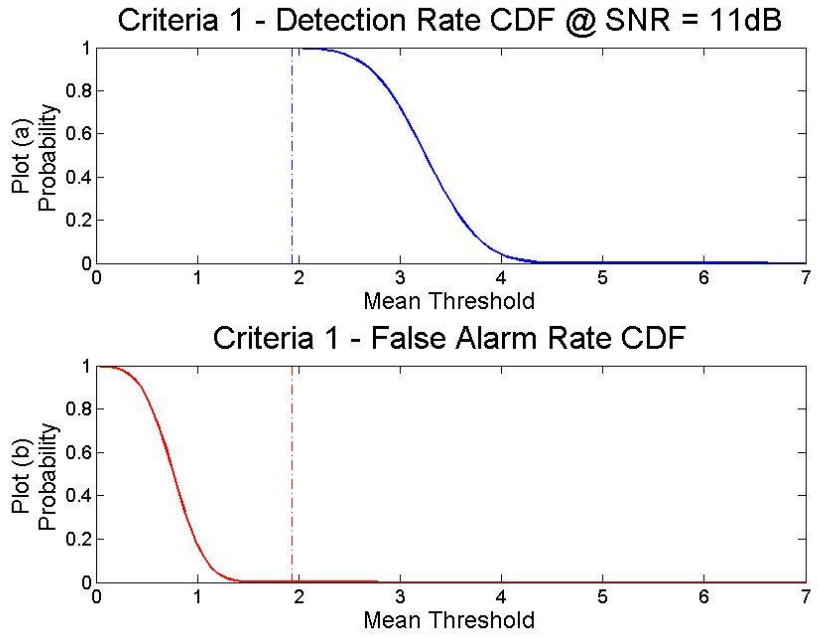

Figure 8 - Cumulative Distribution Function for TOA criteria 1 @ SNR = 11dB.

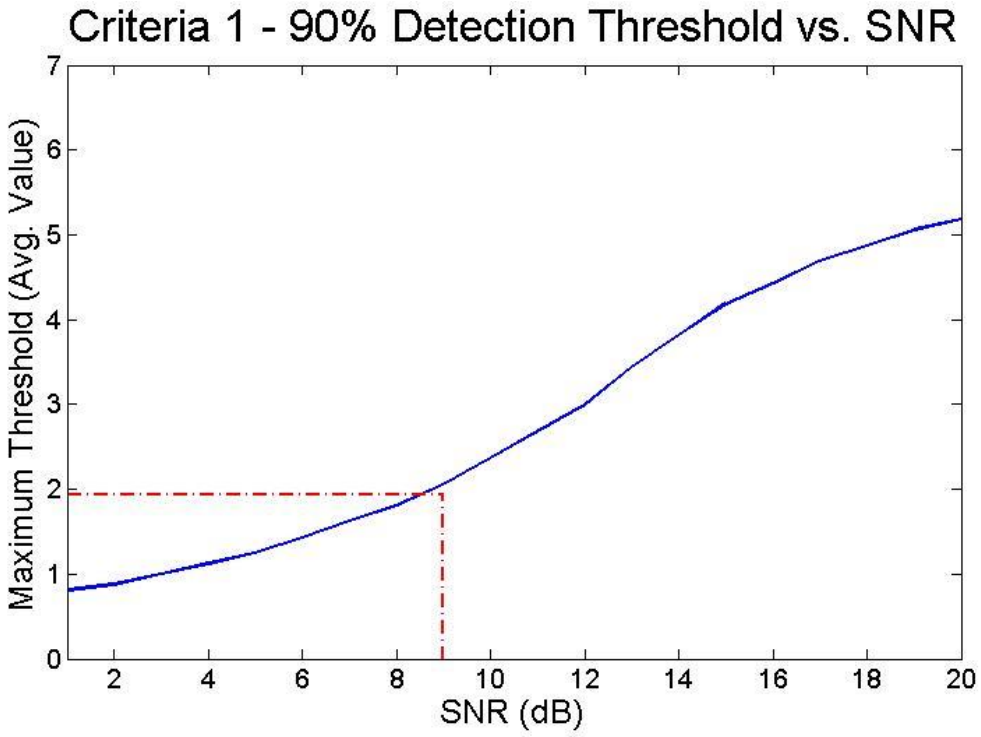

Figure 9 - Criteria 1 plot of 90% detection rate and 10^{-7} false alarm rate thresholds.

Fig. 9 shows the SNR and threshold combinations, which are able to simultaneously meet the detection and false alarm rate requirements. Any threshold value above the horizontal dotted red line, or any SNR to the right of the vertical dotted red line represents an acceptable threshold/SNR combination, which simultaneously meets the 90% detection rate and 10^{-7} false alarm rate requirements.

The same plots were generated to measure the performance of criteria 2, which operates using the maximum magnitude value within an 8-sample block. An additional constraint was added to criteria 2, which sets the minimum number of occurrences required within an 8-sample block. Fig. 10 shows the 90% detection rate threshold vs. SNR for criteria 2 when only 1 magnitude occurrence is required. The lowest SNR, which meets both detection and false alarm requirements when utilizing a single

magnitude occurrence, is 10dB. The performance of criteria 2 is slightly improved when two occurrences of a magnitude value are required and a plot of the 90% detection threshold vs. SNR is shown in Fig. 11. The use of three occurrences achieved the best performance with an SNR of 8dB, and is plotted in Fig. 12. Figures 13 and 14 show the 90% detection threshold vs. SNR when 4 and 5 magnitude occurrences are required, but the performance is degraded when compared to using 3 occurrences.

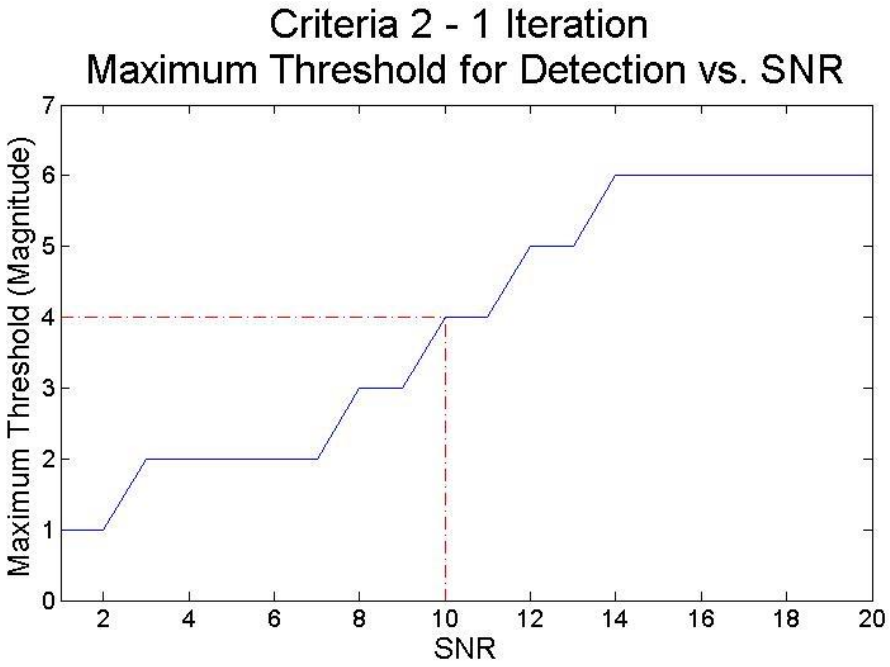

Figure 10 – Criteria 2 – 1 iteration plot of 90% detection rate and 10^{-7} false alarm rate thresholds.

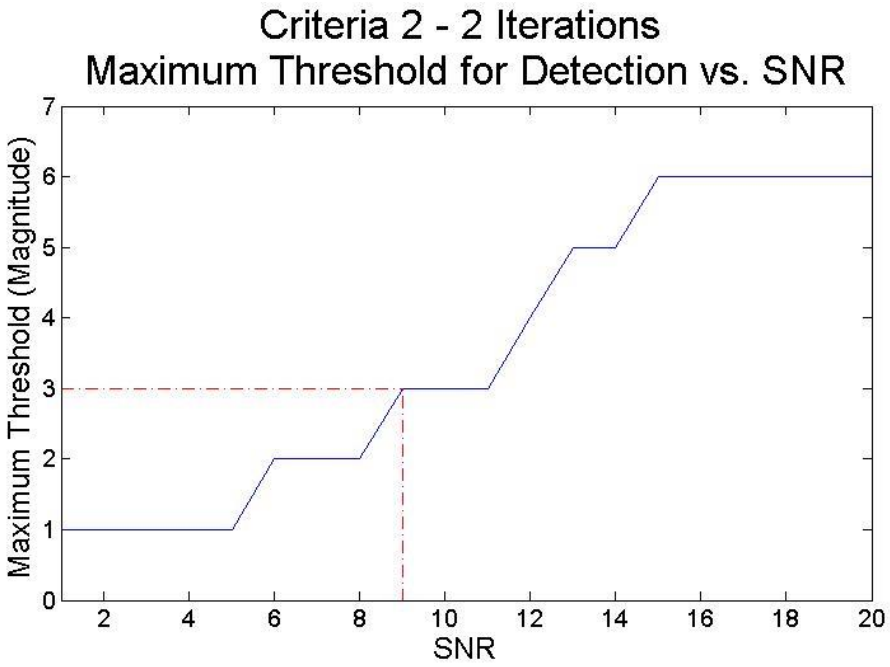

Figure 11 - Criteria 2 – 2 iterations plot of 90% detection rate and 10^{-7} false alarm rate thresholds.

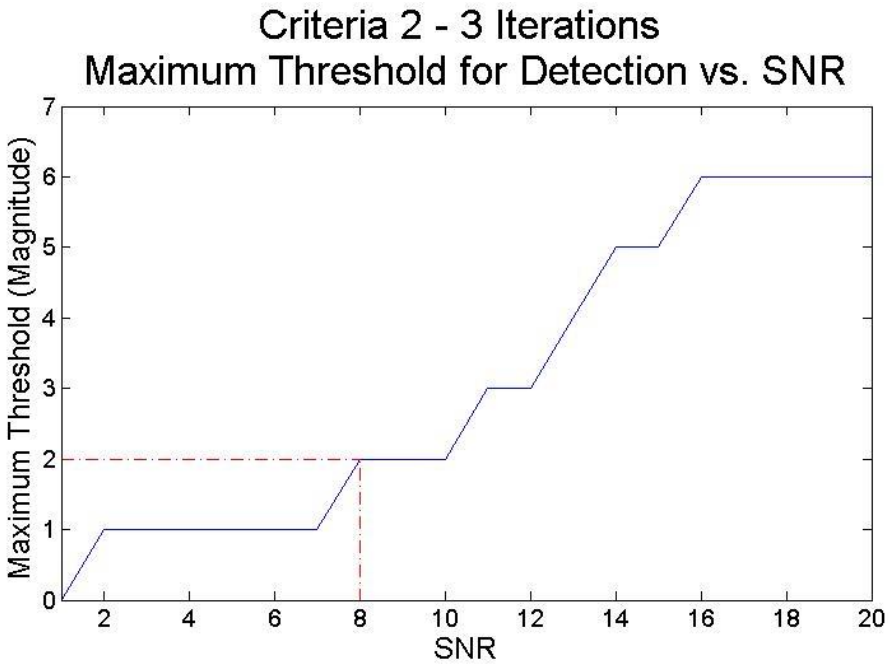

Figure 12 - Criteria 2 – 3 iterations plot of 90% detection rate and 10^{-7} false alarm rate thresholds.

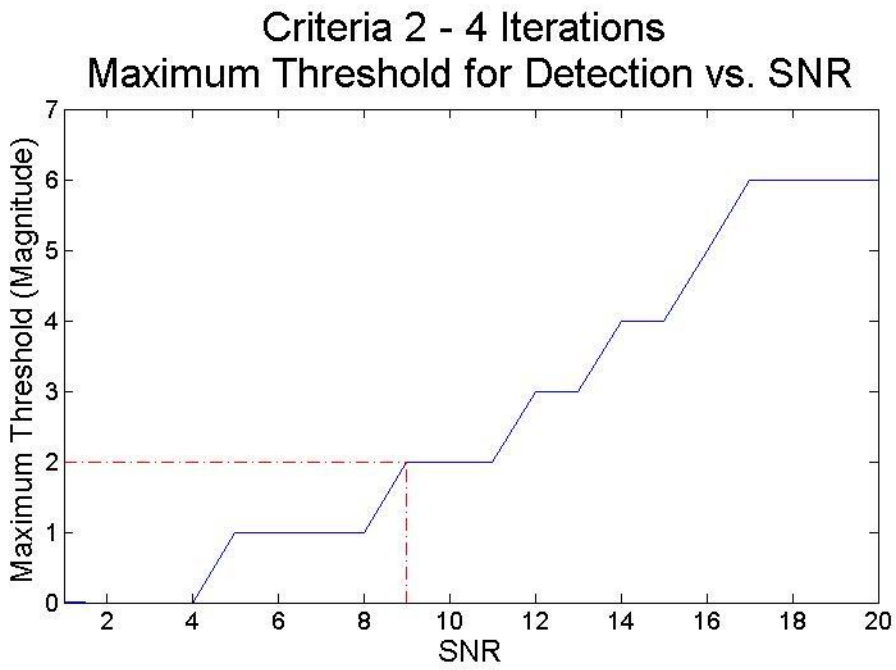

Figure 13 - Criteria 2 – 4 iterations plot of 90% detection rate and 10^{-7} false alarm rate thresholds.

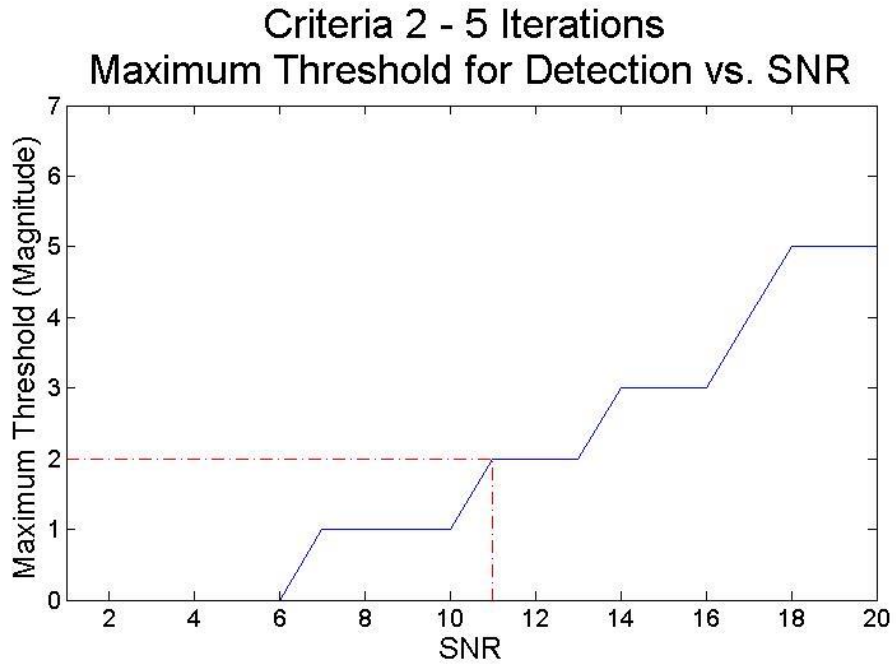

Figure 14 - Criteria 2 – 5 iterations plot of 90% detection rate and 10^{-7} false alarm rate thresholds.

The distributions for all criteria are not assumed to be normally distributed, such

is the case with criteria 2, which uses the maximum value within an 8-sample window. However, the distribution for this criteria is assumed to be normal-like, as a very large number of samples are utilized, and useful estimations are derived from the use of the CDF.

Detection rates and false alarm rates were also estimated when both criteria are utilized simultaneously. After analyzing the data, it was found that no combination of thresholds could meet the detection rate, false alarm rate and SNR requirements simultaneously. An additional study was done to evaluate the detection performance when multiple threshold crossings were considered over a series of 8-sample blocks. Instead of limiting signal detection to a single 8- sample block, multiple blocks were considered, which required further threshold optimization. Investigation was necessary to determine the optimal thresholds, the optimal number of sample blocks to consider as well as the number of threshold crossings which would be required. To find the optimal configuration a Matlab simulation was used to test the following conditions:

- SNRs from 1dB up to 10dB
- Criteria 1 thresholds between 1 and 2
- Number of sample windows to consider for detection between 2 windows and 8 windows

During the study, it was determined that TOA variability could be greatly reduced if the number of threshold crossings required was equal to the number of sample windows being considered minus 1. For example, if 6 consecutive sample windows are evaluated and only 2 threshold crossings are required for signal detection, the system will

encounter an inherent variability in TOA estimation of 4 sample windows (32 samples). This is because the signal TOA could occur in the first 2 sample windows, or the last 2 sample windows, the difference between the two points is 4 samples windows. This variability can be reduced by requiring the number of threshold crossings to be one less than the number of sample windows being evaluated, reducing the variability to only 1 sample window. This also reduced the number of combinations which needed to be considered during the simulations. The simulations found that only 2 threshold/sample window combinations were able to simultaneously meet the detection rate and false alarm rate requirements at an SNR of 5dB.

The first combination that met the requirements used a threshold setting of 1.125 for criteria 1 while requiring the thresholds to be crossed 7 sample windows out of 8 and is shown in Fig. 15. The second combination that met the requirements used a threshold setting of 1.25 for criteria 1 while requiring the thresholds to be crossed for at least 5 sample windows out of 6 and is shown in Fig 16.

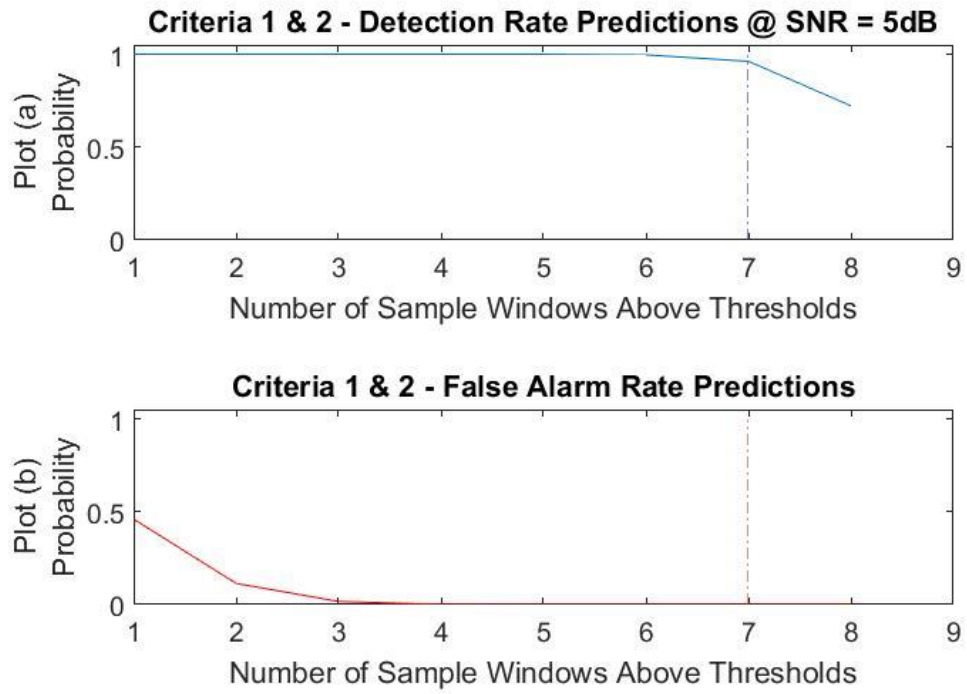

Figure 15 – Criteria 1 & 2 predicted detection and false alarms rates utilizing 8-sample windows.

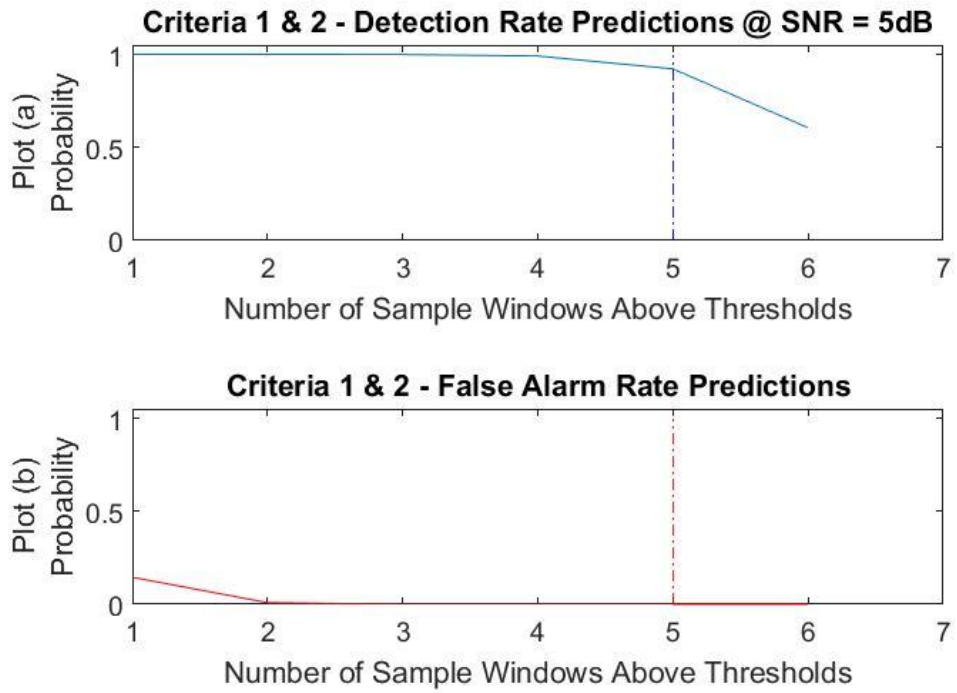

Figure 16 – Criteria 1 & 2 predicted detection and false alarms rates utilizing 6 sample windows.

The second combination was chosen for use as it requires 2 sample windows (16 samples) less to make a detection decision. Even though the use of criteria 1 and 2 meet all the initial requirements for signal detection, the threshold setting does not provide any adjustability or fault tolerance. In order to increase the robustness and flexibility of the TOA algorithm, a third criteria was added.

Criteria 3 operates by monitoring the average of the absolute values of each 8-sample window over an extended period of time. During this evaluation period, the number of threshold crossings is recorded. If the total number of threshold crossings is higher than the criteria, then a signal is considered detected. The purpose of criteria 3 is to verify the correct detection of a signal and is therefore only activated after criteria 1 or 2 have their thresholds crossed for at least 5 sample windows out of 6 consecutive sample

windows. Since criteria 1 and 2 utilize a much shorter evaluation period, they achieve a finer temporal resolution. The optimal evaluation period for criteria 3 was studied and modeled in Matlab. Lengths of 50ns (16 sample windows), 100ns (32 sample windows), and 200ns (64 sample windows) were investigated for use.

Statistical data for criteria 3 was extracted from the large data sets originally created for TOA evaluation. All signals and noise samples were analyzed in 50ns, 100ns, and 200ns blocks, while recording the average number of threshold crossings for each threshold possibility as well as calculating standard deviations. New CDFs were generated to represent the detection rates and false alarm rates across the various threshold settings.

Figures 17 through 19 show the thresholds which meet the 90% detection rate and 10^{-7} false alarm rate requirements based on the statistical data. The red line in the figures represents the lowest threshold combinations which meet the 10^{-7} false alarm rate. The blue line in the figures represents the largest threshold combinations which meet the 90% detection rate requirement. The shaded blue area between the two lines represents all threshold combinations which meet both requirements simultaneously.

It can be seen from the figures that all evaluation lengths are capable of simultaneously meeting the false alarm and detection requirements, however choosing an evaluation period of 200ns allows for a much greater range and flexibility in threshold settings. Since the minimum pulse width for the signals in this research is 400ns, criteria 3 is able to use 200ns of the pulse length for evaluation without any increase in latency or throughput.

Since the TOA algorithm operates over a span of 70 sample windows, 6 sample windows for criteria 1 and 2 followed by 64 sample windows for criteria 3, the definition of the false alarm rate requirement needs to be redefined. A false alarm rate of 1 in 10 conveys that for every 10 signals detected, it is likely that one of the detected signals is actually a false alarm. From this definition, if 70 sample windows are required for signal detection, then our false alarm rate should be based on 70 sample windows. A false alarm rate of 10^{-7} insinuates that a false alarm should occur no more frequently than once every 7.0e7 sample windows. In order to meet this new false alarm rate, criteria 3's final average threshold was set to 1, with a minimum number of crossings equal to 42 windows out of 64 windows.

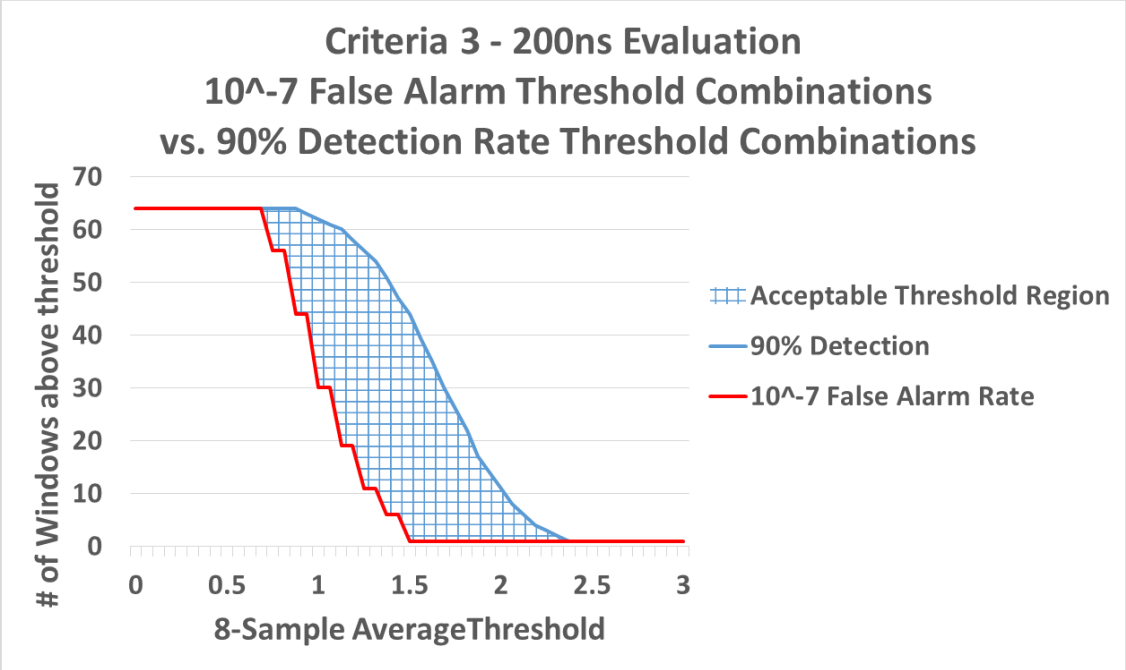

Figure 17 – Criteria 3 acceptable threshold region utilizing 200ns evaluation period.

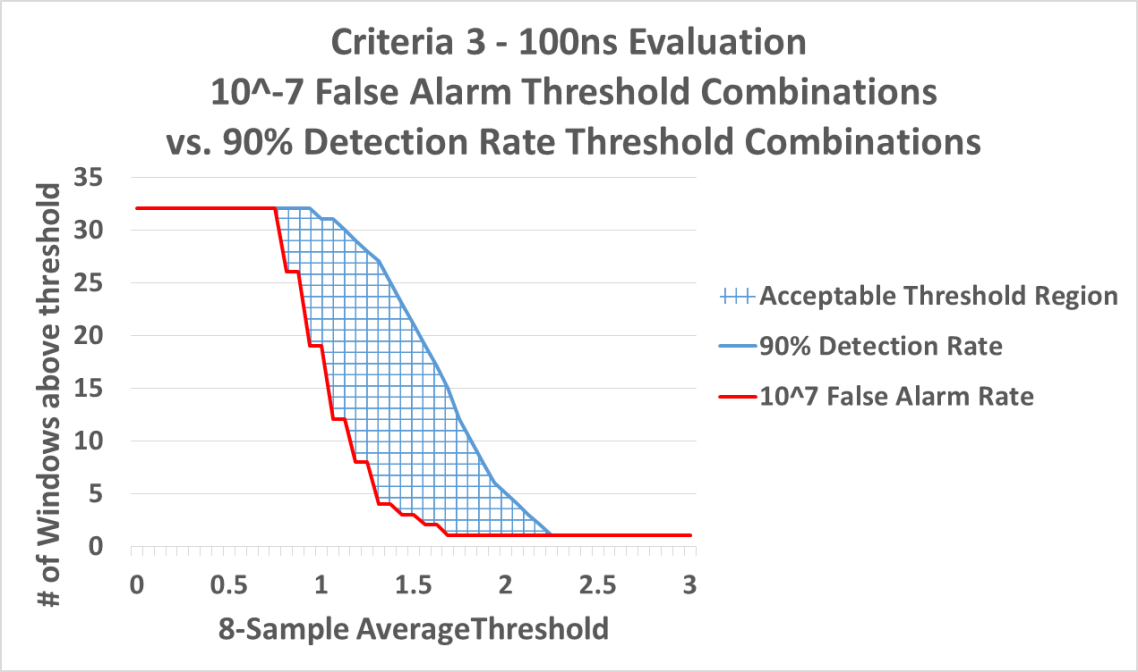

Figure 18 - Criteria 3 acceptable threshold region utilizing 100ns evaluation period.

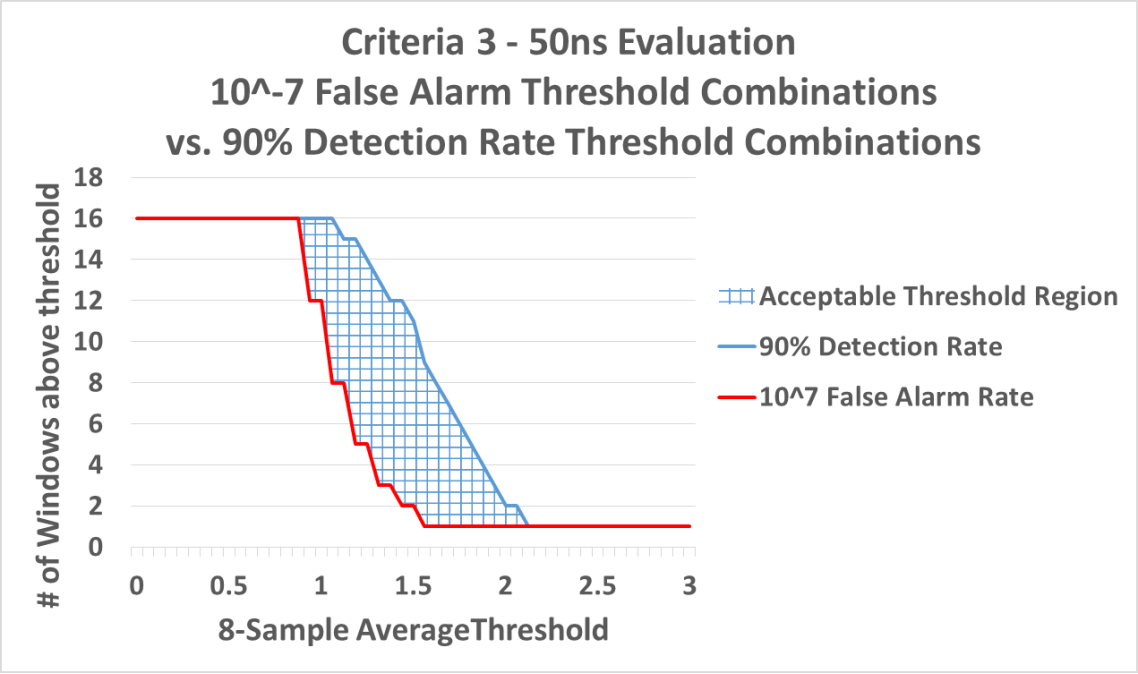

Figure 19 - Criteria 3 acceptable threshold region utilizing 50ns evaluation period.

An example pulsed-wave (PW) signal activating all 3 criteria has been provided in Fig. 20 to allow the reader to visualize the operation of the TOA algorithm. The dotted red horizontal lines represent the thresholds for each criteria. The green vertical dotted lines for criteria 1 and 2 encapsulate 6 sample windows and the green vertical dotted lines in criteria 3 encapsulate 64 sample windows. The blue line represents the average value or maximum magnitude value for each sample window. The actual signal begins at sample window 32. The figure shows that criteria 1 and 2 operate simultaneously and the signal causes both criteria's thresholds to be crossed for at least 5 out of 6 continuous sample windows. Once this occurs, criteria 3 is activated and evaluated over 64 sample windows (200ns). Since all of the sample windows are above the threshold set for criteria 3, the signal will be considered detected at the end of the 200ns evaluation period.

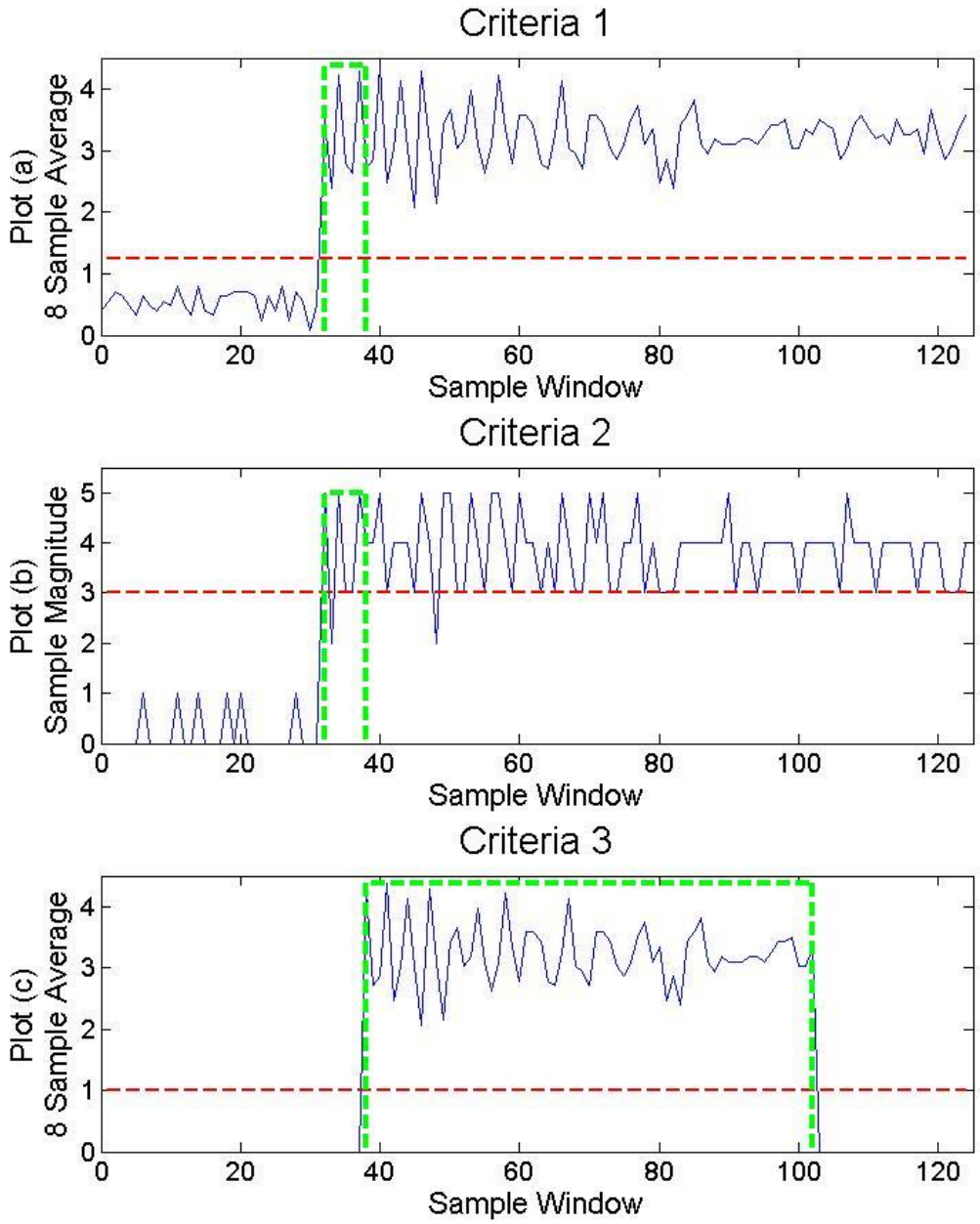

Figure 20 – TOA Algorithm activation example.

3.4 Time-of-Departure for Chirp Signals

TOD detection utilizes the same criteria and thresholds as the TOA detector in order to simplify the implementation. However, instead of counting the number of

threshold crossings, the TOD detector tracks the number of sample windows in which none of the thresholds are crossed.

If the TOA of a signal was previously detected, then criteria 1 and criteria 2's threshold crossings are monitored. If they are not crossed for at least 5 sample windows out of 6, then a TOD is considered detected. After TOD detection, the algorithm is reset and will attempt to detect the next TOA of an incoming signal.

If no signal was previously detected and neither criteria 1 nor criteria 2's thresholds are crossed for at least 5 sample windows out of 6, then the algorithm is reset and will begin waiting for the next TOA of an incoming signal.

The accuracy of the TOD detector is particularly important for the variable chirp receiver, which will be discussed in detail in Section VIII Variable Chirp Receiver. The variable chirp receiver specifically requires the collection of the last 384 samples of a signal, which is reliant on an accurate TOD measurement.

Additionally, the TOD detector also plays an important role in the proper operation of all three digital chirp receivers. If the TOD detector's thresholds are set too low, then it can cause the algorithm to be unable to reset before the arrival of the next signal. This can cause multiple signals to be considered as a single signal, or it can inject substantial TOA measurement bias into the receiver. If the TOD detector's thresholds are set too high, then the algorithm may reset too frequently, causing a single signal to be detected multiple times which can lead to incorrect frequency measurements. The thresholds for the TOD detector and TOA are therefore carefully balanced to ensure the proper operation of the algorithm and to minimize false alarms.

3.5 Performance Evaluations

The statistical data cultivated from the data sets allowed for the analysis and optimization of the thresholds for the TOA algorithm. However simulations were needed to verify that the TOA algorithm operated as the model predicted and to quantify the temporal performance of the algorithm. To accomplish this, the TOA algorithm was implemented in Matlab using the optimized thresholds. All signal types were tested and swept across their entire frequency range. Actual detection rates and false alarms were recorded for SNRs from 5dB up to 16dB. Fig. 21 shows the predicted estimates for signal detection and false alarm rates as well as the simulated detection rates and false alarms from the implemented TOA algorithm. The statistical data predicted a detection rate of 99.88% while the TOA algorithm achieved an actual detection rate of 99.95%. A total of $1.9x10^9$ noise samples were used to test the accuracy of the false alarm rate predictions, It was found no false alarms were encountered.

After verifying the operation of the TOA algorithm, further simulations were needed to test the TOA temporal accuracy. This was accomplished by simulating all signal types across their entire frequency range while TOA, TOD, and pulse width measurements were recorded for SNRs from 5dB up to 20dB.

The TOA model achieves an average TOA error of 2.76 samples (1.07ns), an average TOD error of 3.45 samples (1.35ns), and an average pulse width error of 5.32 samples (2.08ns) for SNRs from 5dB to 20dB. The temporal resolution provided from the TOA model is well below the 10ns requirement and is acceptable for use with high chirp rate signals with short pulse durations. The plot of TOA/TOD and pulse width measurement error can be seen in Fig. 22.

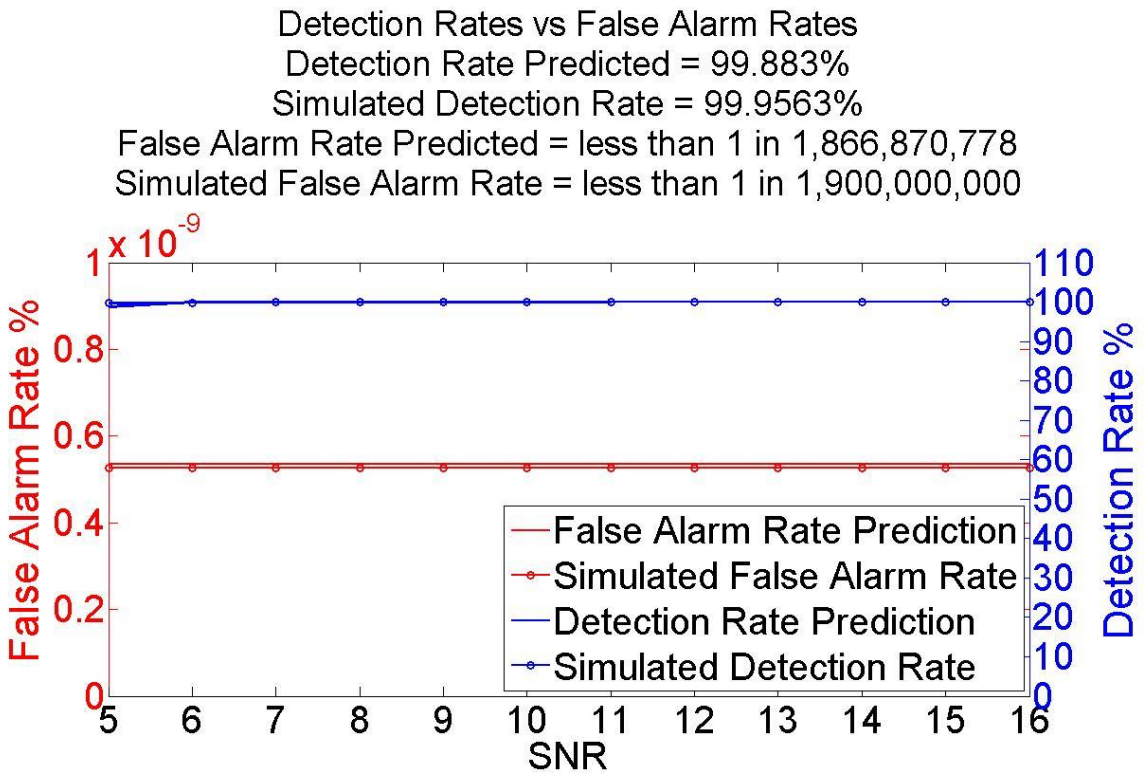

Figure 21 – TOA Algorithm predicted detection and false alarm rates for SNRs 5dB to 20dB.

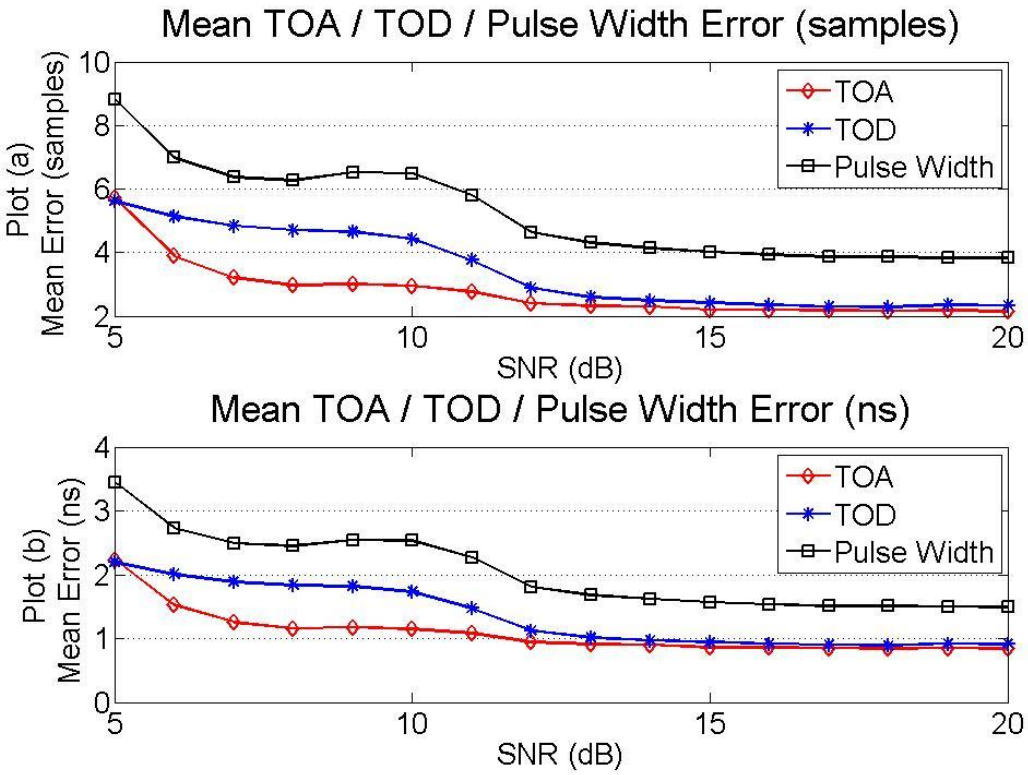

Figure 22 – TOA algorithm TOA/TOD/Pulse Width performance for SNRs from 5dB to 20dB.

IV. Digital Implementation of Hilbert Transform for Wide Bandwidth Signals

4.1 Introduction

In signal processing, Hilbert Transforms are commonly used to transform single channel real data into their complex signal representation. The Hilbert Transform is a linear operator which applies a 90° phase shift to an incoming real valued signal. Complex valued signals are often desired, as is the case with the digital IFM. The digital IFM requires a complex valued signal as it measures the instantaneous phase of incoming signals, effectively calculating $\tanh^{-1}\left(\frac{\mathrm{Q}}{\mathrm{I}}\right)$ where *Q* represents the imaginary component of a signal and *I* represents the real component of a signal. It is for this reason that the Hilbert Transform is a necessary component of the digital chirp receivers and will be discussed in detail in the following sections.

4.2 Theory

The equation for the Hilbert transform is given by:

$$\mathrm{g(y)} = \mathrm{H(f(x))} = \frac{1}{\pi}\,\mathrm{PV}\int_{-\infty}^{\infty}\frac{f(x)dx}{x-y} \tag{17}$$

and its inverse is given by:

$$\mathrm{f}(x) = H^{-1}\big(\mathrm{g(y)}\big) = -\frac{1}{\pi}\,\mathrm{PV}\int_{-\infty}^{\infty}\frac{g(y)dy}{y-x} \tag{18}$$

Some of the most useful transformations [20] from the Hilbert Transform can be seen in the following table:

f(x)	g(y)
cos x	sin y
sin x	-cos y

If the input signal to the Hilbert Transform is of the form cos(x), then the output from the Hilbert transform is of the form sin(y). This transformation is widely used within the signal processing community, as it creates a 90° phase shift between the input and output signals. This is important as it allows for the generation of a complex valued signal when only real data is provided. A complex signal can be defined as follows using Euler's formula:

$$e^{ix} = \cos(x) + isin(x) \tag{19}$$

which can be visually represented using the unit circle, shown in Fig. 23.

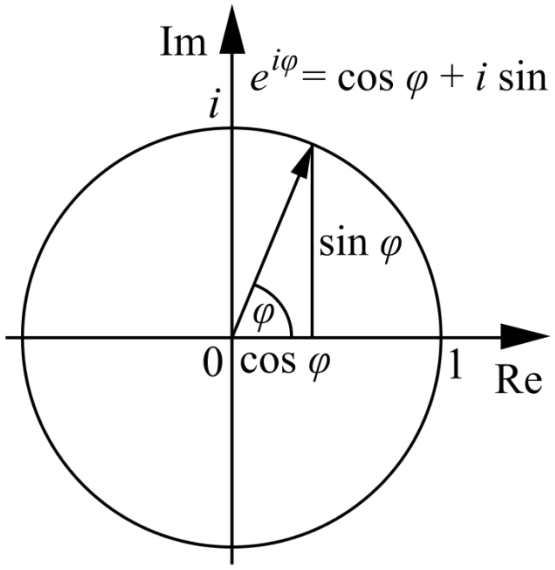

Figure 23 – Unit circle showcasing real and imaginary components.

4.3 Digital Implementation

The discrete Hilbert Transform can be modeled by the following equation:

$$\mathrm{h}(n) = \frac{1}{2\pi}\left[\int_{-\pi}^{0} ie^{i\omega n}d\omega - \int_{0}^{\pi} ie^{i\omega n}d\omega\right] \tag{20}$$

which can be modeled as an IIR filter using the following:

$$\mathrm{h}(n) \stackrel{\text{def}}{=} \begin{pmatrix} 0, & \text{for } n \text{ even} \\ \dfrac{2}{\pi n}, & \text{for } n \text{ odd} \end{pmatrix} \tag{21}$$

This filter can also be implemented by an FIR approximation. If an odd number of anti-symmetric coefficients are used, the filter will be a Type III, which will function as a band pass filter. If an even number of anti-symmetric, non-zero coefficients are used, the filter will be a Type IV and will function as a high pass filter [21].

A Hilbert transform was originally developed for use with the digital IFM in [22]. The original IFM was based on a mono-bit design and the Hilbert transform was designed accordingly to accept 1-bit samples. The original HT used in the design was implemented by a set of LUTs which effectively implemented an 11-tap FIR filter with the coefficients [6 0 10 0 30 0 -30 0 -10 0 -6]. Since there is an odd number of anti-symmetric coefficients, it operates as a Type III, band pass filter. The coefficients for the FIR filter were determined by scaling the optimal coefficients until all integer coefficients were obtained [-1/5 0 -1/3 0 -1 0 1 0 1/3 0 1/5] x 30 = [6 0 10 0 30 0 -30 0 -10 0 -6].

In order to enhance the performance of the digital IFM to allow for accurate chirp signal measurement, the original IFM implementation was modified to accept 2-bit samples instead of only 1-bit samples. In depth details on the digital IFM and the changes made are provided in Section V. This design change required the original HT to be redesigned since the original implementation would be unusable with multi-bit samples.

One of the biggest advantages to the mono-bit HT implementation is that no multipliers or adders are required, which greatly reduces the complexity and difficulty of meeting timing within the FPGA implementation. Another great benefit is that the mono-bit implementation does not suffer any amplitude imbalances between the real and imaginary components.

Unfortunately the modification of the digital IFM required a more complex implementation for the HT. The choice of FIR filter coefficients plays a huge role in the performance of the HT as well as the ease of physical implementation.

The design of FIR filters typically involves a tradeoff between the desired frequency response and the hardware limitations of the target platform. To achieve a more desirable frequency response, it is common to increase the number of taps in the filter. However, this will require a greater number of adders and multipliers, which can quickly consume the hardware resources.

A better frequency response can also be obtained by approximating the filter coefficient values as closely as possible to their optimal values. This too has drawbacks, as it increases the number of bits necessary for each addition and multiplication operation. In order to reduce the complexity of the design as well as reduce the hardware resource utilization, it is often desirable to avoid the use of multipliers whenever possible.

One of the simplest ways to avoid the use of multipliers in FIR filter design is to choose filter coefficients which are based on powers of 2. In digital design, multiplication by any power of 2 can be accomplished by a shift to the right (division) or a shift to the left (multiply). By utilizing power of 2 coefficients, all multipliers can be replaced with simple shift and add operations, which utilizes far less hardware and incurs less latency.

An optimal HT implementation using an 11 tap FIR filter will have the coefficients [-1/5 0 -1/3 0 -1 0 1 0 1/3 0 1/5]. If coefficients are chosen based on powers of 2, then the values which most closely approximate the optimal values are the following coefficients: [-1/8 0 -1/4 0 -1 0 1 0 1/4 0 1/8].

Fig. 24 plot (a) shows the comparison of the optimal coefficient values vs. coefficients based on powers of 2 for a 35-tap Type III filter. It is easily recognizable that all even numbered coefficients for the Type III filter are zero valued. Fig. 24 plot (b) shows the comparison between the optimal coefficient values vs. coefficients based on powers of 2 for a 34-tap filter Type IV filter. As expected, there are no zero valued coefficients in the Type IV filter.

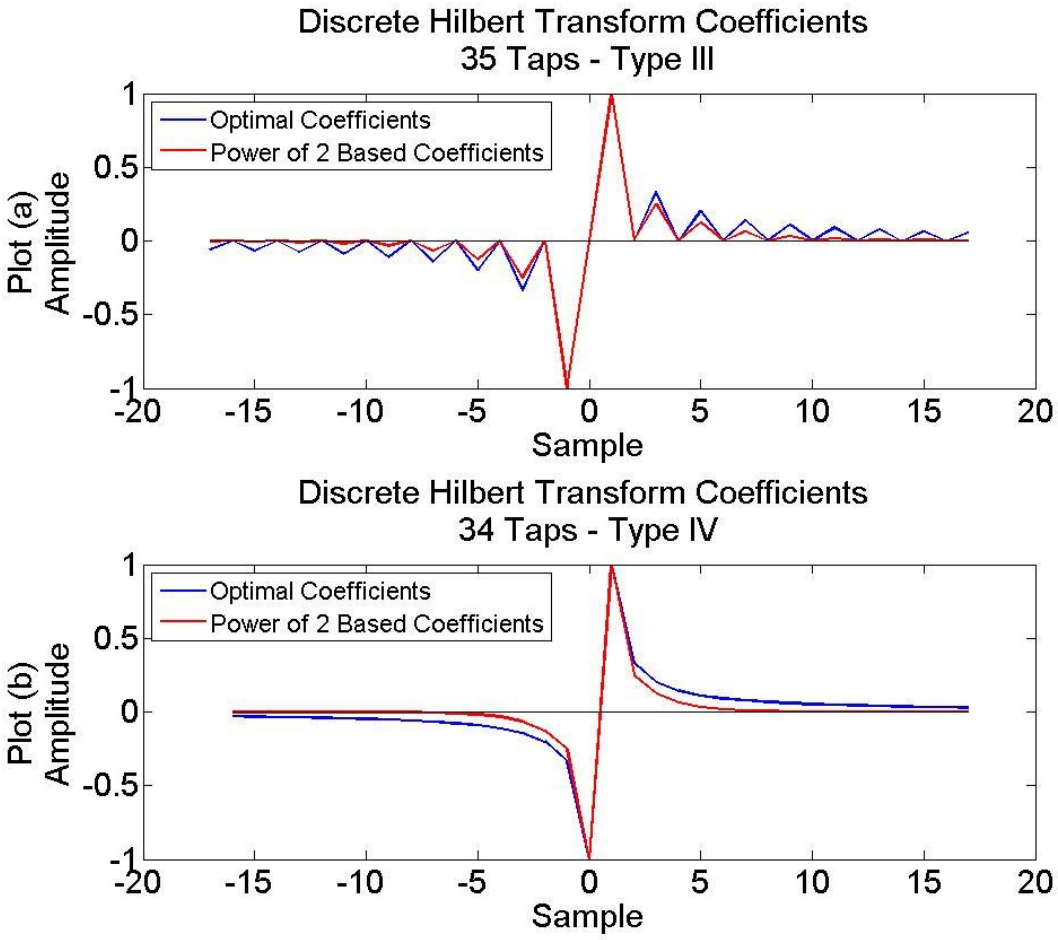

Figure 24 – 34/35 tap Hilbert Transform coefficient plots using power of 2 based coefficients.

We can examine the frequency response of the HT as the number of taps is increased, starting with a simple 3 tap design. The frequency response for a simple 3 tap Hilbert transform can be seen in Fig. 25. This filter uses the coefficients [1 0 -1] and achieves very poor performance for frequencies near zero and near the Nyquist zone. As the number of filter coefficients are increased, a wider passband region can be obtained at the expense of increased filter implementation complexity. Fig. 26 shows the frequency response comparison for a 7 tap HT when the optimal coefficients are used vs. using powers of 2 based coefficients. Since only a few taps are used, there are only minor

differences between the two coefficient choices. The same results can be seen when a 6 tap Type IV filter is implemented and is shown in Fig. 27.

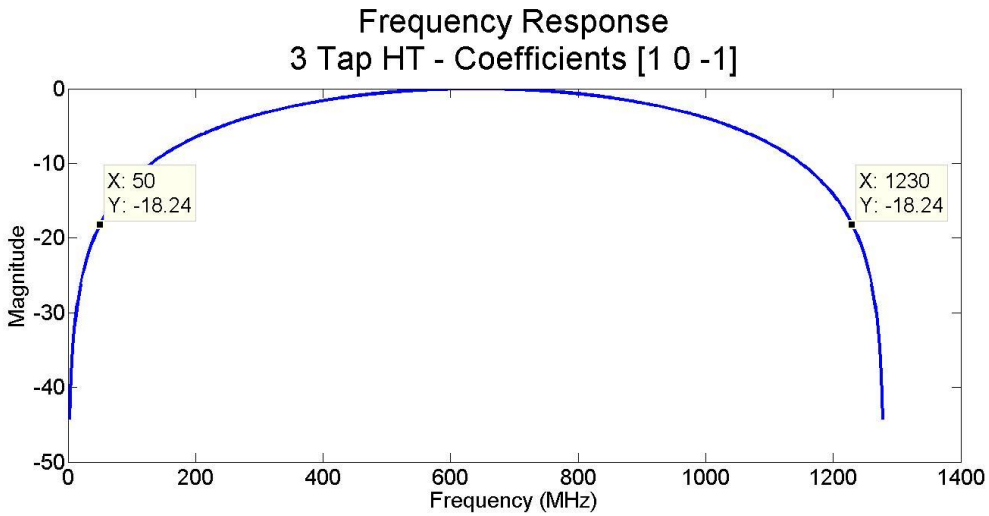

Figure 25 – Frequency response of 3-tap Hilbert Transform.

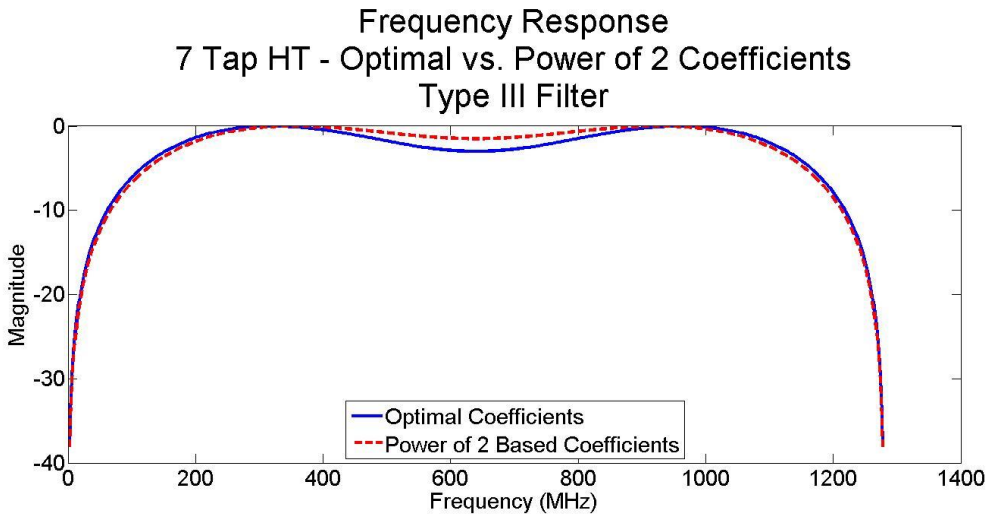

Figure 26 - Frequency response of 7-tap type III Hilbert Transform.

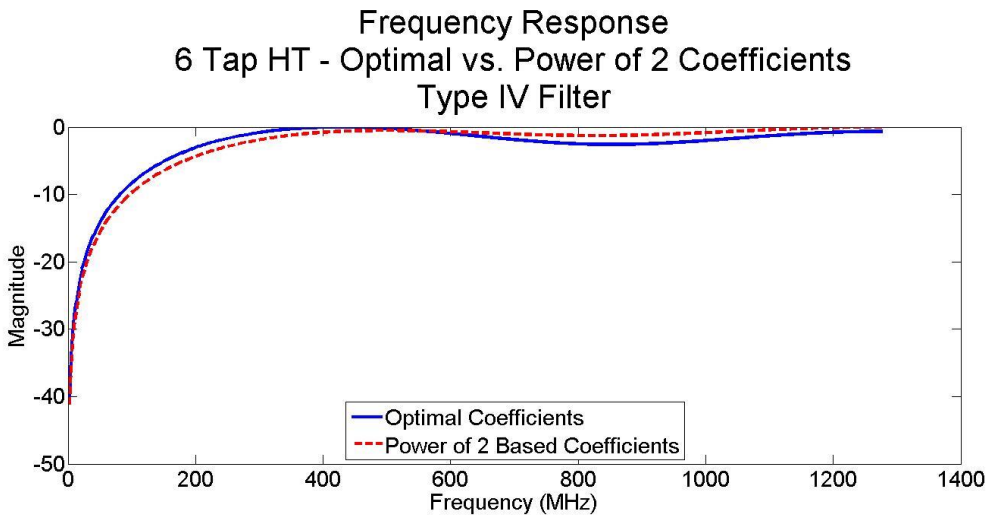

Figure 27 - Frequency response of 6-tap type IV Hilbert Transform.

If the number of filter taps is increased to 11, it is shown that the passband region has been widened, and the difference between the two coefficient choices is more easily noticeable. The optimal coefficients achieve a better performance for frequencies near zero and near the Nyquist zone. The frequency response for the 11-tap Type III can be seen in Fig. 28 and the 10-tap Type IV can be seen in Fig. 29.

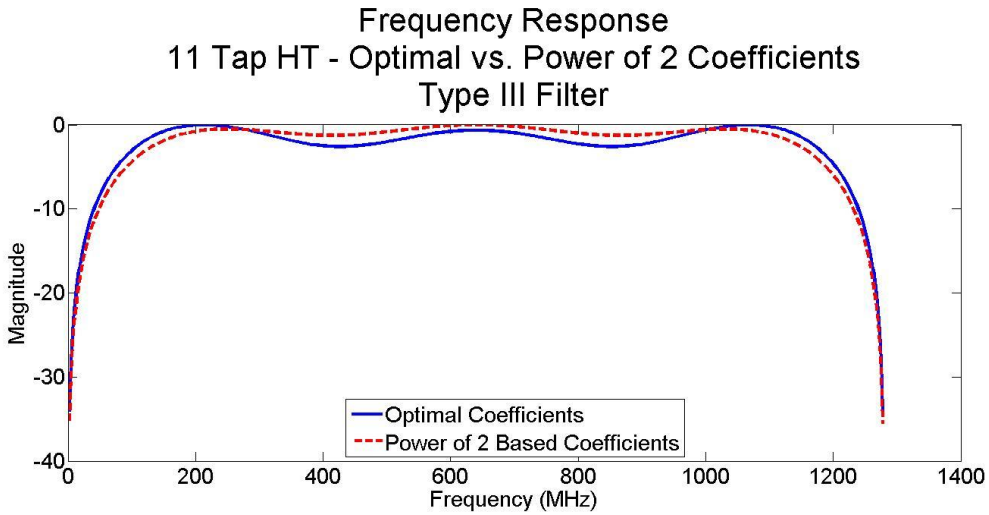

Figure 28 - Frequency response of 11-tap type III Hilbert Transform.

A comparison of the Type IV filters shows many of the same tradeoffs as the Type III. The optimal coefficients perform better for frequencies near zero, however the power of 2 based coefficients encounter less passband ripple effects.

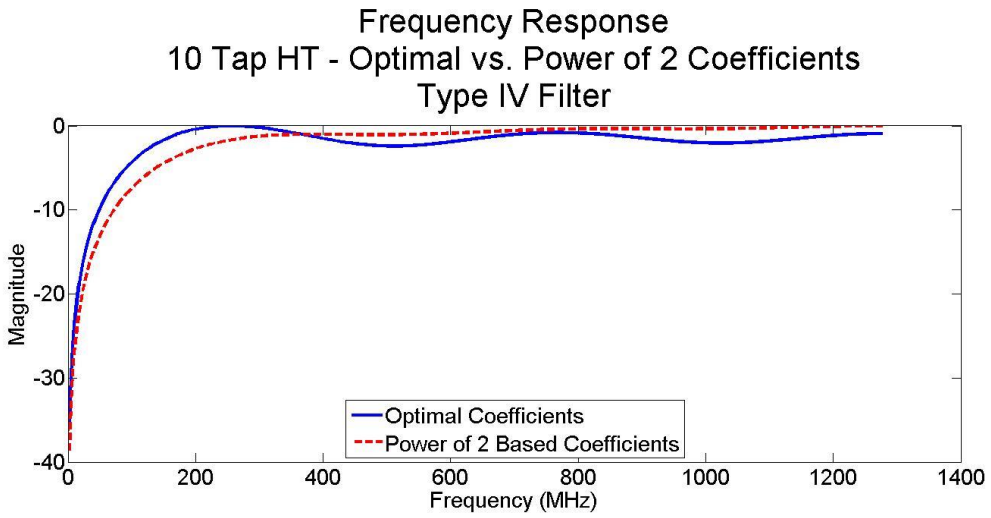

Figure 29 - Frequency response of 10-tap type IV Hilbert Transform.

As the number of taps is increased for both filter types, the differences between the optimal coefficients and the power of 2 based coefficients continue to become more significant. Fig. 30 shows that the optimal coefficients achieve a much better response for frequencies near zero and near the Nyquist zone for a 35 tap Type III filter. The coefficients based on powers of two on the other hand, achieve a much flatter passband region. Once again, the same differences are present in the 34 tap Type IV filter design as shown in Fig. 31.

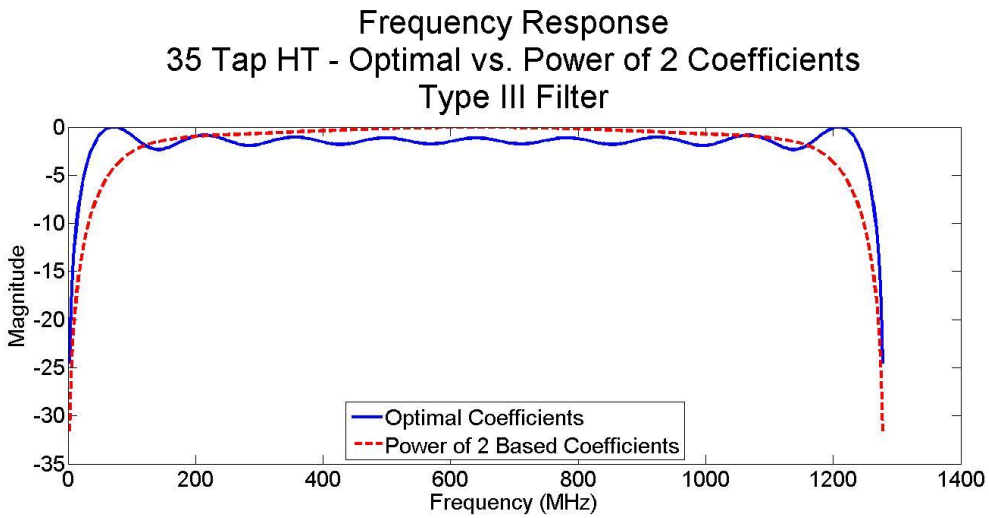

Figure 30 - Frequency response of 35-tap type III Hilbert Transform.

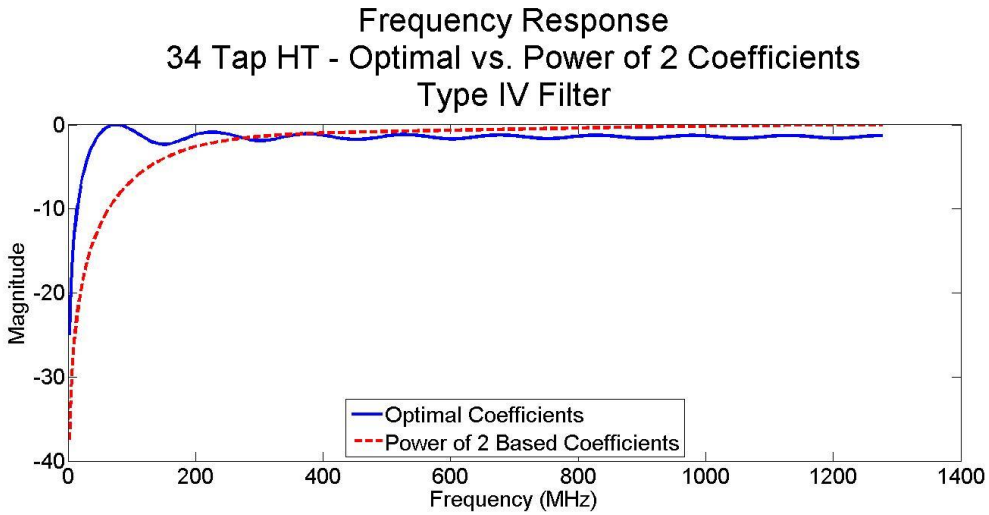

Figure 31 - Frequency response of 34-tap type IV Hilbert Transform.

A better approximation can be achieved if we do not limit ourselves to only coefficients based on powers of 2. The use of multipliers can still be avoided by using shift and add operations to mimic a multiplication in order to approximate filter coefficients. A simple example will be used to show how the shift and add approximation works. If we try to approximate the coefficient $\frac{1}{3}$, we must first choose the maximum allowable bit shift for the approximation, keeping in mind that larger bit shifts will allow

for closer approximations. For this example, we will use a maximum bit shift of 10 bits, giving us a coefficient resolution of $\frac{1}{2\text{^}10} = \frac{1}{1024}$. We can calculate the necessary shift and add operations through the following decomposition: $\frac{1024}{3} = 341.333$. By rounding to the nearest integer, the fraction $\frac{341}{1024}$ will be the closest approximation to $\frac{1}{3}$, assuming that no more than 10 bit shifts are possible. The approximation $\frac{341}{1024}$ can then be separated into a series of the fractions, representing the necessary shift and add operations. The coefficient $\frac{341}{1024}$ can be represented as $(\frac{256}{1024} + \frac{64}{1024} + \frac{16}{1024} + \frac{4}{1024} + \frac{1}{1024})$, which can be simplified to $(\frac{1}{4} + \frac{1}{16} + \frac{1}{64} + \frac{1}{256} + \frac{1}{1024})$. The maximum number of bit shifts and the coefficient being approximated will affect the number of shift and add operations required, certain coefficients will require more shift and add operations than others.

Fig. 32 shows the optimal filter coefficients vs. approximated coefficients using only shift and add operations. The largest shift used to approximate the coefficients in Fig. 32 plot (a) is 6-bits, while a maximum shift of 8-bits was used to approximate the coefficients in Fig. 32 plot (b). Better approximations can be achieved by using a greater number of bit shifts.

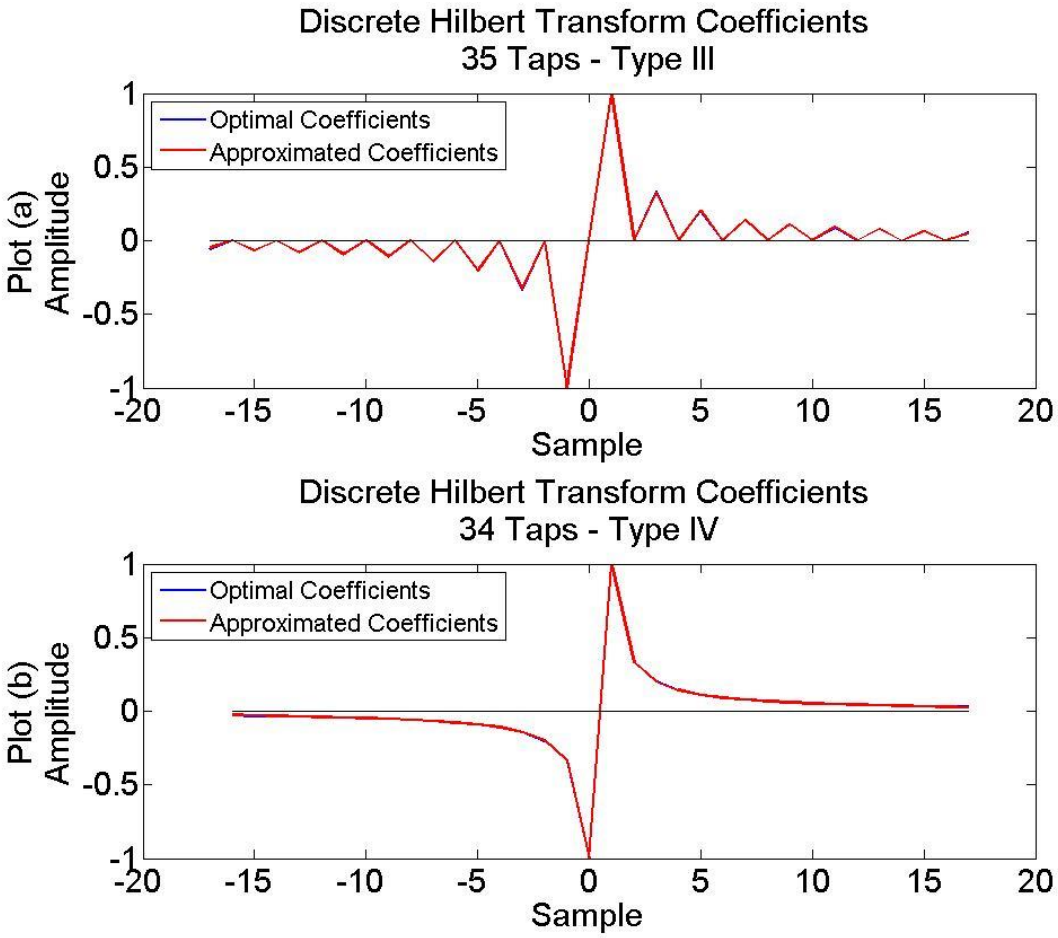

Figure 32 - 34/35 tap Hilbert Transform coefficient plots using approximated coefficients.

Various tables were generated which highlight important characteristics of the optimal filter coefficients vs the approximated filter coefficients, including the number of adders/multipliers required for each implementation as well as the 3dB cutoff frequency. The tables also show the amount of suppression at 50MHz, which is relevant to the chirp receiver research as it was the targeted as the lower frequency bound for the designs.

Table I shows the performance of the optimal coefficients for various filter lengths and the necessary number of adders and multipliers to implement each Type III filter.

Table I – Resource and performance comparison for Type III Hilbert Transform with optimal coefficients.

Type III Filter				
Optimal Coefficients				
# Taps	# Adders	# Multipliers	50MHz Suppr. (dB)	3dB Cutoff Freq (MHz)
3	8	0	-18.24	322.5
7	24	16	-11.80	155
11	40	32	-8.33	102.5
15	56	48	-6.01	77.5
19	72	64	-4.32	62.5
31	120	112	-1.34	40
43	168	160	-0.16	27.5
63	248	240	-0.39	20
87	344	336	-1.93	15
123	488	480	-1.68	10
167	664	656	-1.06	7.5

Table II shows the performance of the approximated filter coefficients when only shift and add operations are utilized for a Type III filter. An additional column is used in Table II, which shows the largest shift utilized and the corresponding maximum number of unique coefficients, which can be generated using the corresponding number of bit shifts. The approximated coefficients require no multipliers, but utilize almost double the number of adders as a result. The 3dB cutoff for the optimal filter and the approximated filter is comparable throughout most filter lengths.

Table II - Resource and performance comparison for Type III Hilbert Transform with approximated coefficients.

Type III Filter					
Approximated Coefficients with only Shift and Add					
Largest Shift	Max # of Taps	# Adders	# Multipliers	50MHz Suppr. (dB)	3dB Cutoff Freq (MHz)
0 bit	3	8	0	-18.24	322.5
1 Bit	7	24	0	-11.02	140
2 Bits	7	24	0	-12.46	167.5
3 Bits	15	72	0	-6.16	77.5
4 Bits	19	104	0	-5.00	67.5
5 Bits	31	184	0	-1.77	42.5
6 Bits	43	280	0	-0.49	32.5
7 Bits	63	424	0	-0.06	22.5
8 bits	87	616	0	-0.92	17.5
9 Bits	123	904	0	-1.90	12.5
10 Bits	167	1272	0	-1.05	10

Table III shows the performance of the filters when only power of 2 based coefficients are used for Type III filters. The number of adders is greatly reduced from the shift and add approximation and no multipliers are required for any of the implementations. However, it can be easily seen from the table that the frequency response of the filters has greatly diminishing returns when the filter length is increased beyond 19 taps.

Table III - Resource and performance comparison for Type III Hilbert Transform with power of 2 based coefficients.

Type III Filter				
Reduced Adder Approximation - Only Powers of 2 Coefficients				
# Taps	# Adders	# Multipliers	50MHz Suppr. (dB)	3dB Cutoff Freq (MHz)
3	8	0	-18.24	322.5
7	24	0	-12.46	167.5
11	40	0	-9.76	125
15	56	0	-8.08	102.5
19	72	0	-7.41	97.5
31	120	0	-6.66	92.5
43	168	0	-6.62	92.5
63	248	0	-6.61	92.5
87	344	0	-6.61	92.5
123	488	0	-6.61	92.5
167	664	0	-6.61	92.5

Tables IV through VI are used to show the optimal coefficients, the approximated coefficients and the power of 2 based coefficients for Type IV filter implementations. The approximated coefficients achieve a similar performance to the optimal coefficients without the requirement of multipliers. The power of 2 based coefficients also show that the frequency response of the filters have diminishing returns when the filter length is increased beyond 16 taps.

Table IV - Resource and performance comparison for Type IV Hilbert Transform with optimal coefficients.

Type IV Filter				
Optimal Coefficients				
# Taps	# Adders	# Multipliers	50MHz Suppr. (dB)	3dB Cutoff Freq (MHz)
2	8	0	-24.25	642.5
4	24	16	-17.74	307.5
4	24	16	-17.74	307.5
8	56	48	-11.69	152.5
10	72	64	-9.80	122.5
16	120	112	-5.98	77.5
22	168	160	-3.62	55
32	248	240	-1.33	40
44	344	336	-0.16	27.5
62	488	480	-0.28	20
84	664	656	-1.71	15

Table V - Resource and performance comparison for Type IV Hilbert Transform with approximated coefficients.

Type IV Filter					
Approximated Coefficients with only Shift and Add					
Largest Shift	Max # of Taps	# Adders	# Multipliers	50MHz Suppr. (dB)	3dB Cutoff Freq (MHz)
0 bit	2	8	0	-24.25	642.5
1 Bit	4	24	0	-16.95	280
2 Bits	4	24	0	-18.40	332.5
3 Bits	8	72	0	-11.85	155
4 Bits	10	104	0	-10.57	135
5 Bits	16	184	0	-6.62	82.5
6 Bits	22	280	0	-4.45	62.5
7 Bits	32	424	0	-2.01	45
8 bits	44	616	0	-0.60	32.5
9 Bits	62	904	0	-0.03	22.5
10 Bits	84	1272	0	-0.99	17.5

Table VI - Resource and performance comparison for Type IV Hilbert Transform with power of 2 based coefficients.

Type IV Filter				
Reduced Adder Approximation - Only Powers of 2 Coefficients				
# Taps	# Adders	# Multipliers	50MHz Suppr. (dB)	3dB Cutoff Freq (MHz)
2	8	0	-24.25	642.5
4	24	0	-18.40	332.5
4	24	0	-18.40	332.5
8	56	0	-13.84	202.5
10	72	0	-13.08	192.5
16	120	0	-12.14	182.5
22	168	0	-12.02	185
32	248	0	-12.00	185
44	344	0	-12.00	185
62	488	0	-12.00	185
84	664	0	-12.00	185

After analyzing the filter performance data, a 43 tap Type III FIR filter was chosen to implement the HT. The 43 tap FIR provided the best tradeoff between frequency response and implementation complexity. Fig. 33 compares the frequency response of the optimal coefficients vs. approximated coefficients for a 43 tap Type III HT. Fig. 34 compares the frequency response of the optimal coefficients vs. approximated coefficients for a 44 tap Type IV HT. It can be seen from the figures that the frequency response of the approximations is very close to the response of the optimal coefficients.

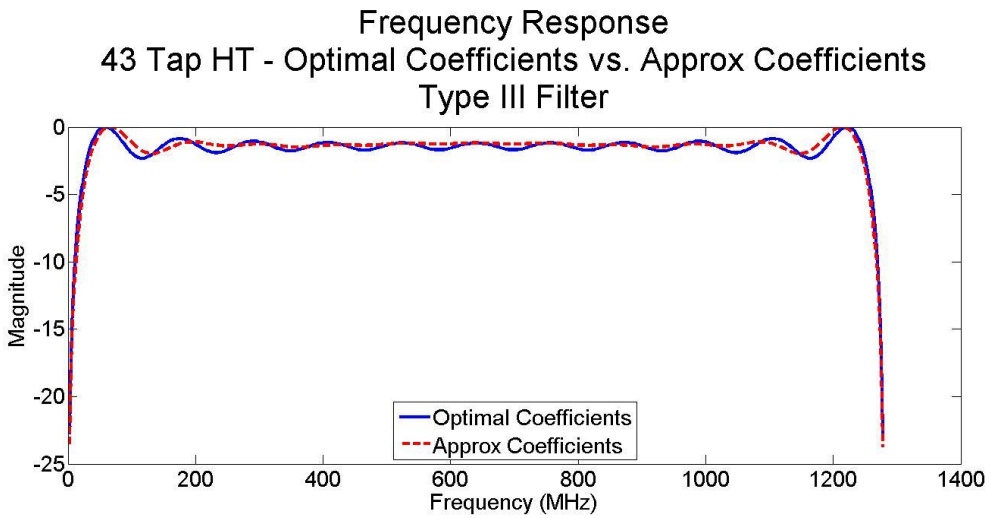

Figure 33 – Frequency response of Type III 43-tap Hilbert Transform filter.

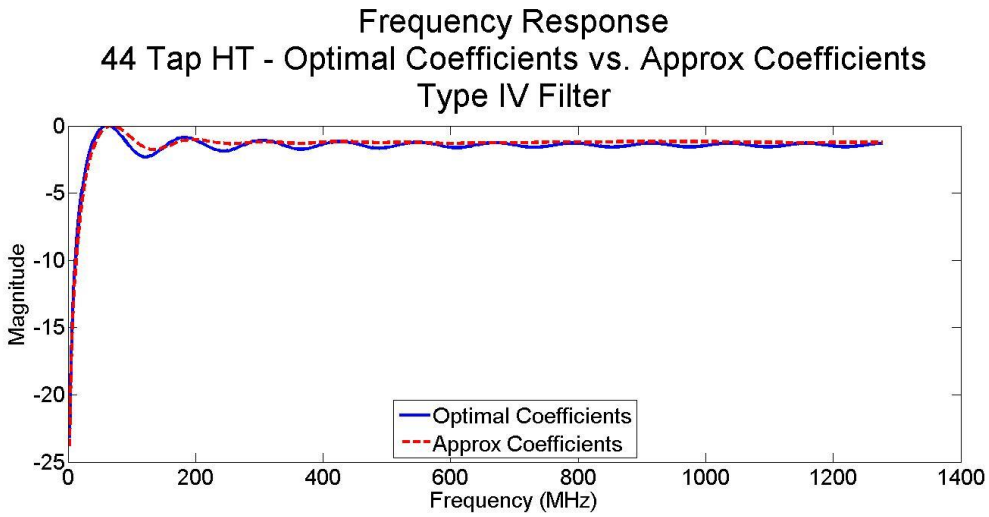

Figure 34 - Frequency response of Type IV 44-tap Hilbert Transform filter.

4.4 Amplitude Imbalance

Amplitude imbalance between the real and imaginary components can cause significant measurement error within the digital IFM. The I/Q imbalance can occur from the inherent scaling caused by the FIR filter or from suppression due to the frequency response of the filter itself. Ideally the filter should be designed to minimize the

variability of the I/Q imbalance across the desired frequency range. Amplitude imbalance in the digital chirp receivers is corrected by utilizing the shift and add operations to approximately correct the imbalance. Since the frequency response of the HT throughout the crucial frequency ranges (50MHz – 1230MHz) is nearly constant, a constant amplitude correction is acceptable. The 43 Tap Hilbert transform suffers a maximum imbalance variability of 2dB for frequencies between 50MHz and 1230MHz. The average scaling for frequencies ranging from 50 MHz up to 1230MHz was simulated to be 1.7115. The correction value of $\frac{1}{1.7115} = 0.5843$ was therefore estimated by using the shift and add approximation (1/2 + 1/16) = 0.5625. This correction factor was used to scale all of the samples outputted from the HT.

The effect of this imbalance on the digital IFM's measurement performance was simulated in Matlab. Carrier frequencies ranging from 50MHz up to 1230MHz were measured using the digital IFM with the imbalance present and were measured again after applying the correction factor previously mentioned. Fig. 35 shows the measurement performance at an SNR of 10dB when the imbalance is present. The digital IFM achieved an average carrier frequency error of 0.146MHz. Fig. 36 shows the measurement performance when the imbalance is corrected; the digital IFM achieved an average carrier frequency error of 0.0848MHz, a difference of 58%.

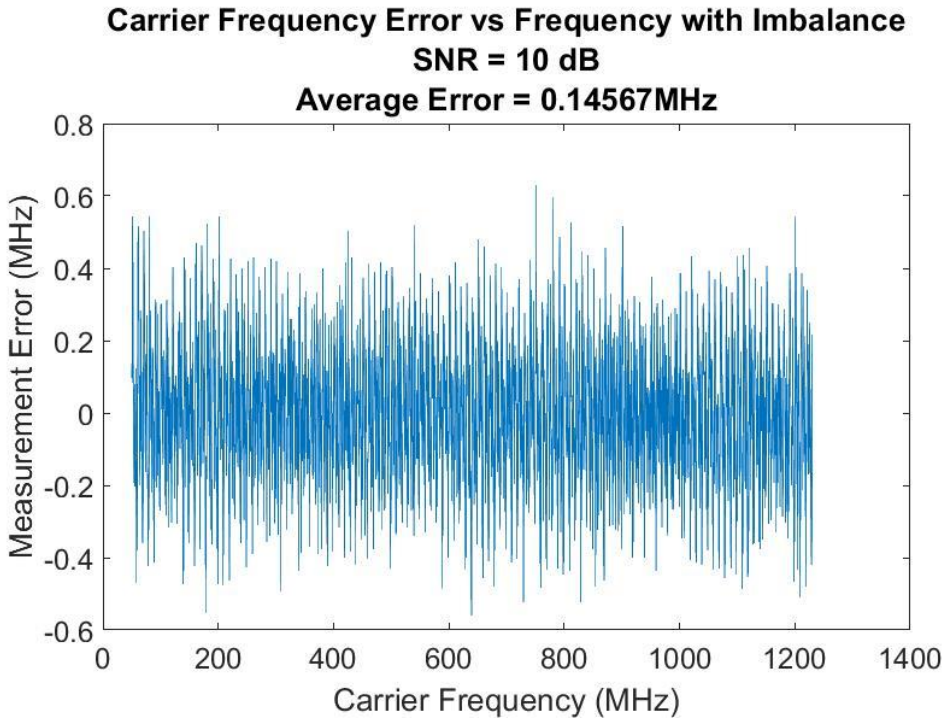

Figure 35 – Digital IFM frequency measurement error when the imbalance from the Hilbert Transform is not corrected.

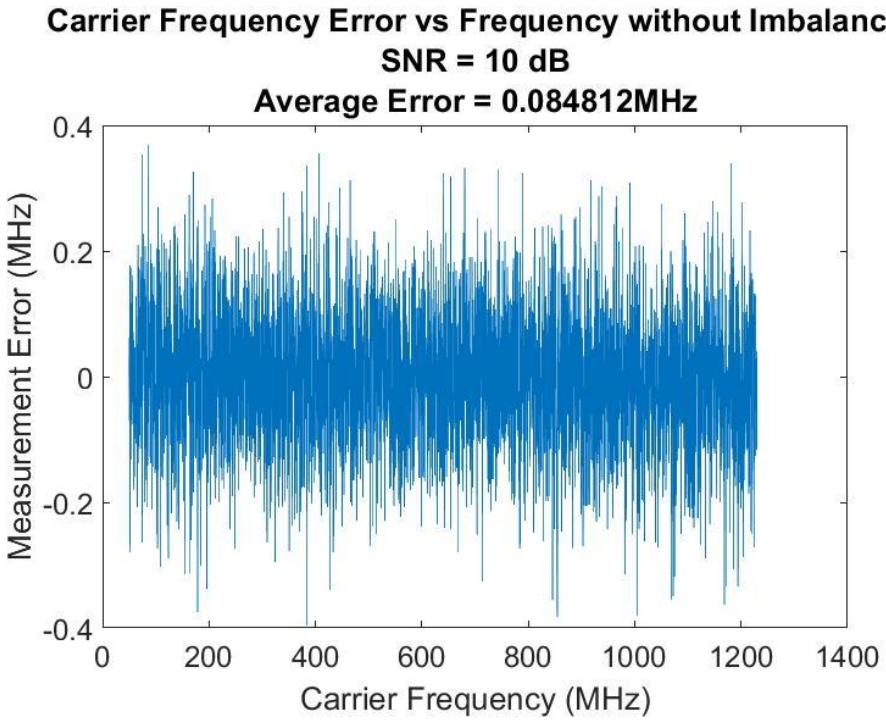

Figure 36 – Digital IFM frequency measurement error when the imbalance from the Hilbert Transform is corrected.

V. Digital Instantaneous Frequency Measurement for Chirp Measurement

5.1 Introduction

The instantaneous frequency measurement (IFM) device was developed in the 1950's as an analog device to allow for "instantaneous" frequency measurement with low system complexity [23]. Since its digital counterpart has been developed, with the advancement of today's FPGAs, it can be easily implemented in digital hardware. The maturing of the IFM is discussed in greater detail in [23, 24]. Recent research has been conducted with utilizes the IFM to tackle an ultra-wideband (UWB) frequency measurement problem [25].

The IFM functions as a phase measurement device, which has a linear relationship to a signal's frequency. IFMs have several desirable traits, which include fine frequency measurement resolution, simple design and implementation as well as easy configurability. One of the biggest downfalls to the IFM is its inability to measure simultaneous signals. Unlike the FFT, the IFM is capable of obtaining fine frequency resolution as well as fine temporal resolution simultaneously. This is due to the fact that the IFM's measurement performance is not solely dependent on the number of samples collected. The hardware complexity of the IFM also does not always scale as its frequency measurement accuracy is improved. Utilizing only 256 samples, the digital IFM

is capable of a frequency resolution of < 1MHz with a bandwidth of over 1GHz if a sampling rate of 2.56GHz is utilized. Comparing this result to a typical FFT, the FFT would require more than 2048 samples at a sampling rate of 2.56GHz to achieve a similar frequency resolution. The temporal resolution of the digital IFM is 100ns compared to the temporal resolution of the FFT which is 800ns or greater. For these reasons, the digital IFM was chosen as the foundation for the digital chirp receiver presented in this research.

5.2 Theory

The digital IFM operates by calculating the phase angle of multiple auto correlators, which are then mapped into their corresponding frequency. The relationship between the phase angle measurement and frequency is linear. A complex signal is required as input into the IFM, which requires the use of the Hilbert Transform for our digital chirp receivers. The incoming complex signal is auto correlated with itself multiple times, with each autocorrelation utilizing a different delay value. The auto correlators used in the digital IFM are *S*m for m = 1, 2, 8, 32, 128, and are shown as follows:

$$S1 = \sum_{n=1}^{256} x(n)'x(n-1) \tag{22a}$$

$$S2 = \sum_{n=1}^{256} x(n)'x(n-2) \tag{22b}$$

$$S8 = \sum_{n=1}^{256} x(n)'x(n-8) \tag{22c}$$

$$S32 = \sum_{n=1}^{256} x(n)'x(n-32) \tag{22d}$$

$$S128 = \sum_{n=1}^{256} x(n)'x(n-128) \tag{22e}$$

The computation of each auto correlator results in a summation of the autocorrelation operation, therefore each auto correlator will be contain a real and imaginary sum. The phase of each correlator can be calculated by the following:

$$\text{Phase}_{Sm} = tan^{-1}\left(\frac{Imaginary(Sm)}{Real(Sm)}\right) \tag{23}$$

The phase of the first auto correlator can be mapped directly into frequency, however, the sensitivity of the measurement to frequency change is very low, making it a poor estimator of the true frequency.

$$f_1 = \theta_1 \tag{24}$$

Auto correlators which utilize longer delays will have an increased sensitivity, allowing for a more accurate frequency approximation. However, as the length of delay is increased, the number of ambiguous frequencies for a given phase angle will be increased. For this reason, the phase of the auto correlator with a smaller delay is used to map the appropriate zone of the auto correlator using a longer delay. This allows frequency ambiguities to be resolved. The calculation of the correct zone for the phase measurement to be mapped to is as follows:

$$z_m = round\left(\frac{(mf_{m-1})}{2\pi}\right) \tag{25}$$

Once the correct zone has been determined, the phase of the auto correlator can be mapped to its appropriate frequency as follows:

$$f_m = \frac{\theta_m}{m} + z_m \frac{2\pi}{m} \tag{26}$$

The final signal frequency can be calculated from the auto correlator with the longest delay by using Eq. 27.

$$\text{Signal\_frequency} = \frac{f_{128}}{2\pi} * f_s \tag{27}$$

Example plots of the phase angle measurement for each correlator and the corresponding zone calculation are shown in Fig. 37 and 38 respectively. It can be seen that the S1 correlator contains no frequency ambiguities, and can therefore directly be mapped into frequency. The phase measurements from S1 and S2 are used to determine the appropriate zone for S8. Once the zone of S8 has been determined, the phase and zone can be determined for S32. Finally the phase of S128 can be measured and mapped to the appropriate zone, which will allow for the final mapping of phase to frequency.

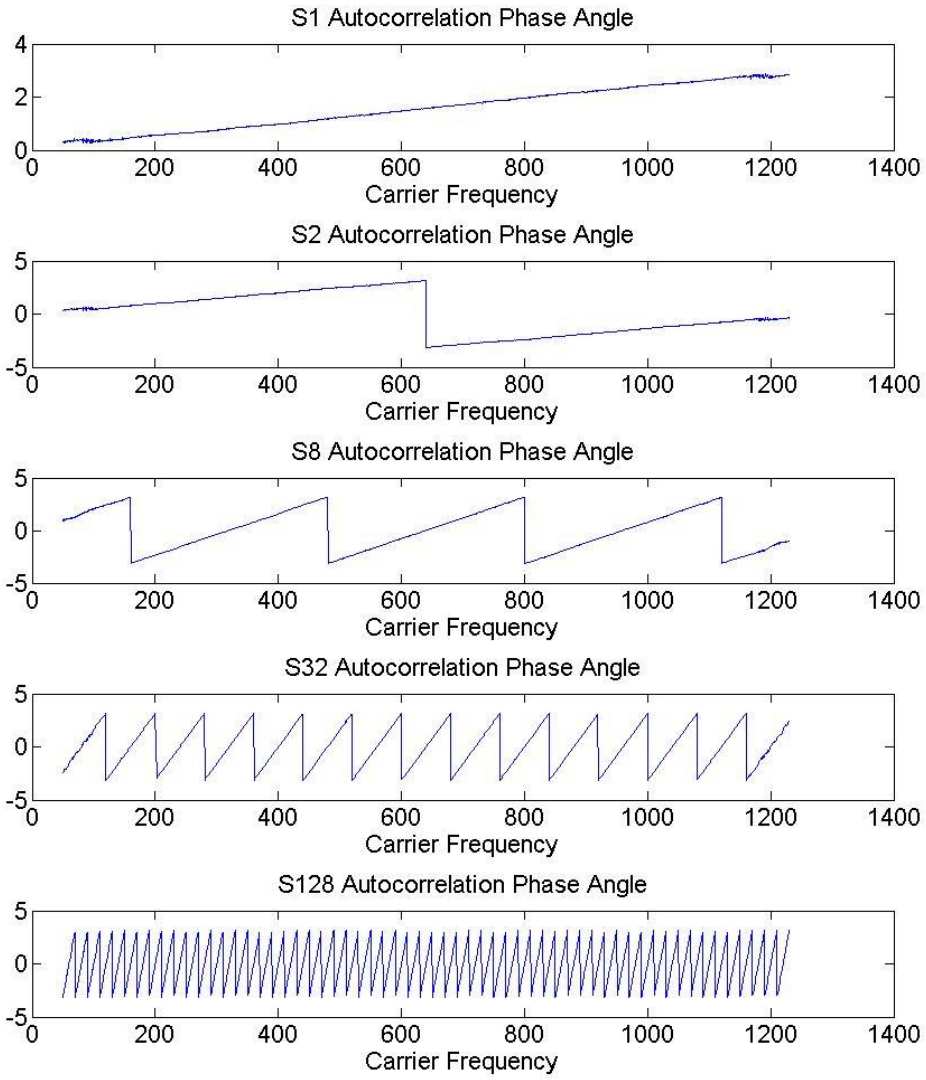

Figure 37 –Phase angle measurement from digital IFM auto correlators.

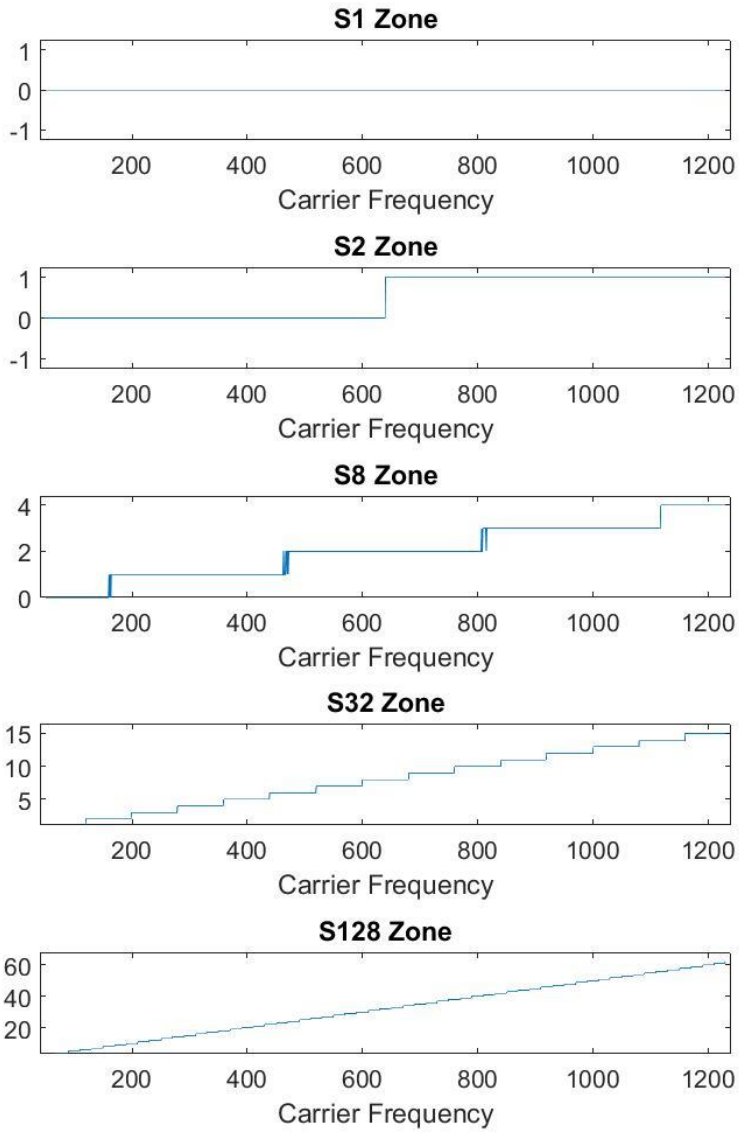

Figure 38 – Zone mapping measurements from digital IFM auto correlators.

5.3 Design Considerations

The digital IFM developed in [26] was designed for wideband stationary carrier frequency measurement. The original design covered a bandwidth of 1.2GHz and completed a frequency measurement every 100ns (based on a sampling frequency of 2.56GHz). Though the IFM will output a frequency measurement every 256 samples (100ns), because of the S128 correlator, the initial frequency measurement will incur an

extra 50ns latency due to the 128- sample delay used. The original design used mono-bit samples in order to minimize the design complexity. The mono-bit approach meant that only the values of [-1, 1] could be represented in the IFM. The digital IFM was later ported into a mono-bit digital chirp receiver in [27].

The frequency measurement performance of the IFM was investigated in order to understand which design criteria have the greatest effect on performance. The study found that the accuracy of the IFM is determined by three criteria, the total number of samples collected, the number of bits used to represent each sample, and the delay values chosen for the auto correlators. Fig. 39 shows the performance of the digital IFM as it was tested across its entire frequency range for each IFM size. The delay values for the correlators are kept constant as 1, 2, 8, 32 and 128, while each sample is represented by 3 bits and the SNR is kept constant at 0dB. The number of samples collected by the IFM was swept from 8 samples up to 512 samples in 4 sample increments.

The plot shows that a significant performance improvement can be had if the number of samples collected is increased. This makes sense intuitively as more samples will result in a higher correlation of the signal, and less correlation of the noise. Additionally a larger sum allows for a finer phase angle to be resolved within the unit circle. Remember, each auto correlator consists of a real and imaginary accumulated total from which the inverse tangent is calculated.

As an example, if only 2 1-bit samples are collected and accumulated, the smallest step size for $\frac{Q}{I}$ will be $\frac{1}{2}$, approximately 26.56°. However, if 16 1-bit samples are collected, the smallest step size for $\frac{Q}{I}$ will be $\frac{1}{16}$, approximately 3.57°. From this example

we can see that collecting a larger number of samples allows for a finer phase measurement to be obtained. This can also be seen graphically in Fig. 40, which represent a unit circle scaled by 2 and 16. The unit circles are meant to represent all possible combinations for the real and imaginary summations if a total of 2 1-bit samples are accumulated for the IFM vs. the accumulation of 16 1-bit samples for the IFM.

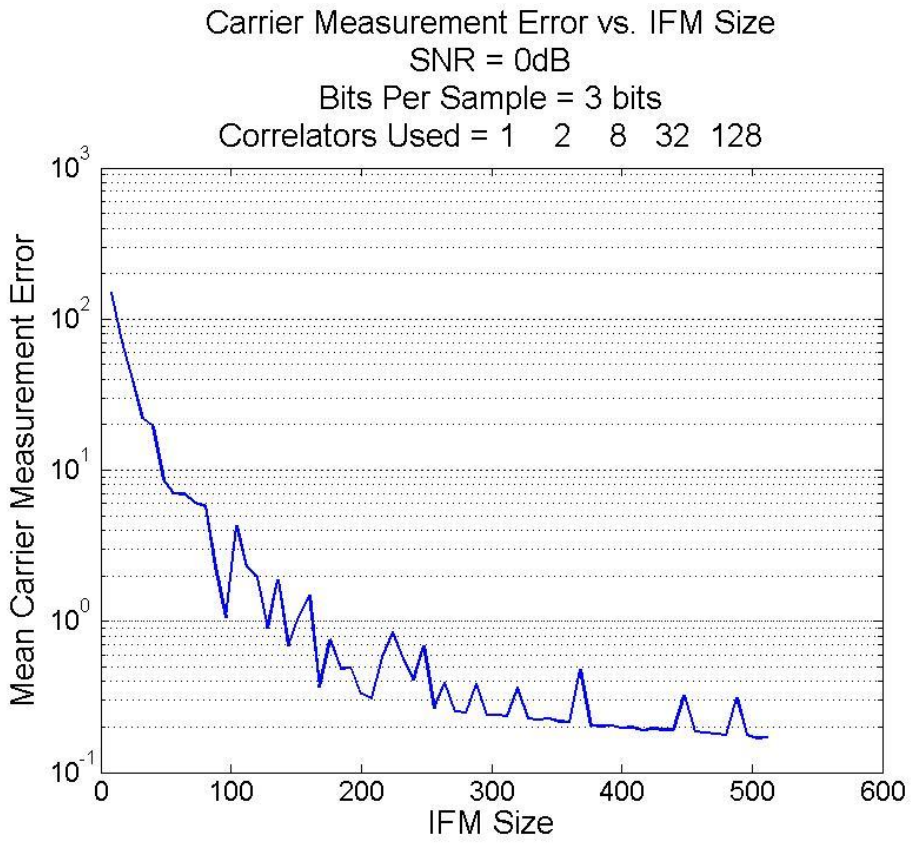

Figure 39 – Digital IFM frequency measurement performance vs. number of samples correlated.

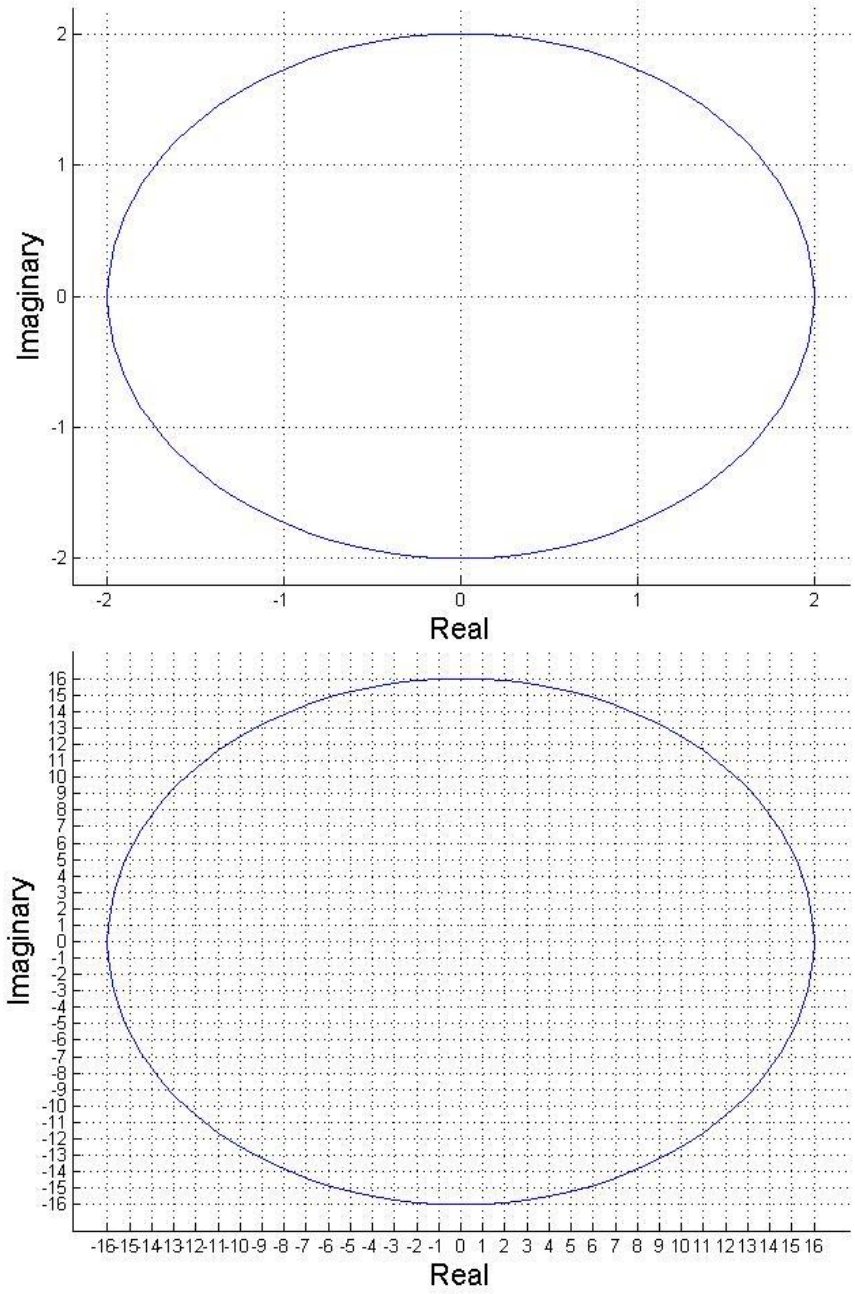

Figure 40 – Unit circle resolution.

The phase measurement resolution of the IFM can also be improved by increasing the amount of data stored within each sample. Fig. 41 shows the average frequency measurement error of the IFM as the number of bits per sample is increased. Again, the entire frequency range from 50MHz up to 1230MHz was tested for each point on the

graph. The SNR for all signals was kept constant at 0dB and the correlators used were [1 2 8 32 128]. The IFM collected a total of 512 samples for each simulation. It can be seen that the largest improvement is gained from increasing the number of bits from 1 to 2. Utilizing a single bit is very restrictive because it does not allow for the representation of the values [-1, 0, 1], instead only the values [1, -1] can be represented. A wider range of values per sample allows for a wider range in the accumulated sums for the auto correlators, allowing for a finer phase measurement to be obtained.

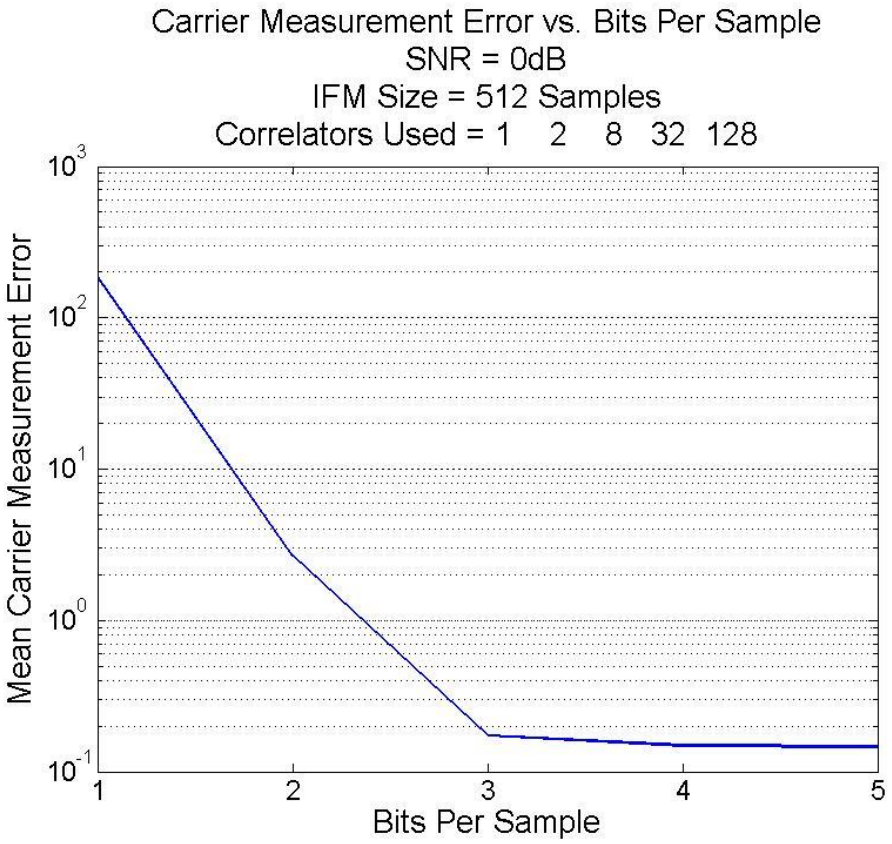

Figure 41 - Digital IFM frequency measurement performance vs. number of bits per sample.

The longest delay chosen for the auto correlators also has a significant impact on the phase measurement capabilities of the IFM, though it is not as intuitively understood

as the other two parameters. Fig. 37 from before was used to show the phase angle measurements from each of the auto correlators as an input signal was swept across the entire frequency range. These plots can also be used to understand why a longer auto correlator allows for a better frequency measurement. The purpose of the shortest delay line is to resolve any frequency ambiguities. Since the entire bandwidth 0MHz – 1.28GHz has a linear relationship with the phase $0 - \pi$, every frequency can be unambiguously converted. However, the sensitivity of the first correlator is very low, meaning a large change in frequency is needed to cause a change in the phase angle measurement. We can see that as longer delay values are chosen, the sensitivity of the auto correlator to a change in frequency is increased. In other words, a very small change in frequency translates to a large change in the phase measurement. This allows for a finer frequency resolution to be obtained. Fig. 42 shows the average measurement error as the longest correlator delay is increased.

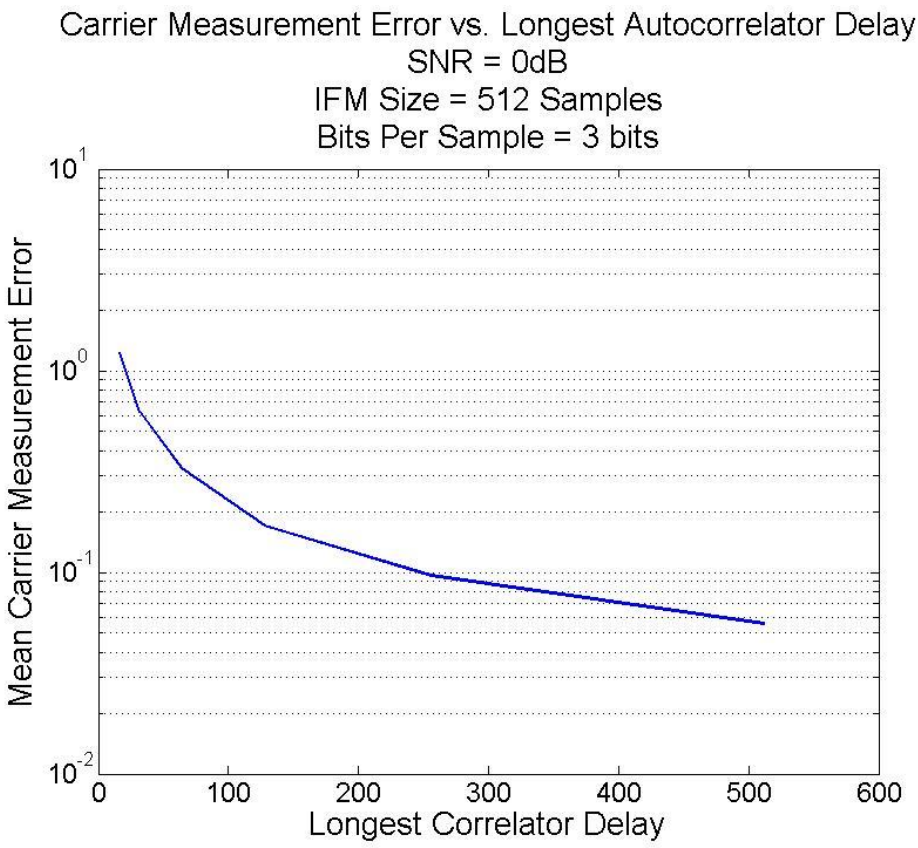

Figure 42 - Digital IFM frequency measurement performance vs. longest correlator delay.

A downfall to this is that the zone determination for auto correlators using longer delays rely on the auto correlators with shorter delays to resolve the frequency ambiguities. This causes the IFM to be susceptible to large 'encoding' errors. If the zone of one of the auto correlators is mapped incorrectly, it will cause the error to propagate forward into the zone mapping for the rest of the auto correlators. The biggest contributor to zone encoding errors is noise. The IFM's resistance to noise interference is directly related to the total number of samples collected. As more samples are collected, it will act as an increase in SNR, as the signal will correlate with itself while the noise will not. The improvement in noise sensitivity can be seen in the following figures. Fig. 43 shows the average IFM measurement error at a low SNR, -5dB when an IFM size of 128 samples is

used with different maximum auto correlator delays. It is apparent that by using only 128 samples, the IFM is too sensitive to noise for the increase in the delay length to cause any measurement improvement.

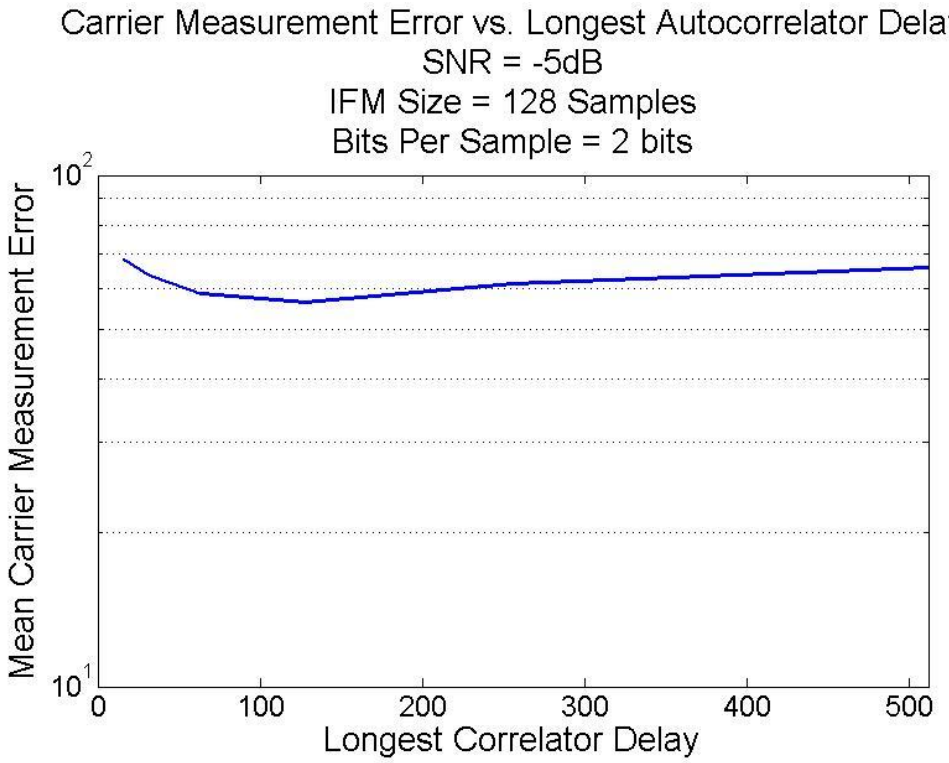

Figure 43 – Effect of longest correlator delay on digital IFM performance at low SNR.

Fig. 44 shows similar results when an increasing number of bits per sample are tested at -5dB SNR with an IFM size of 128 samples. The IFM is not able to function properly in the presence of such strong noise.

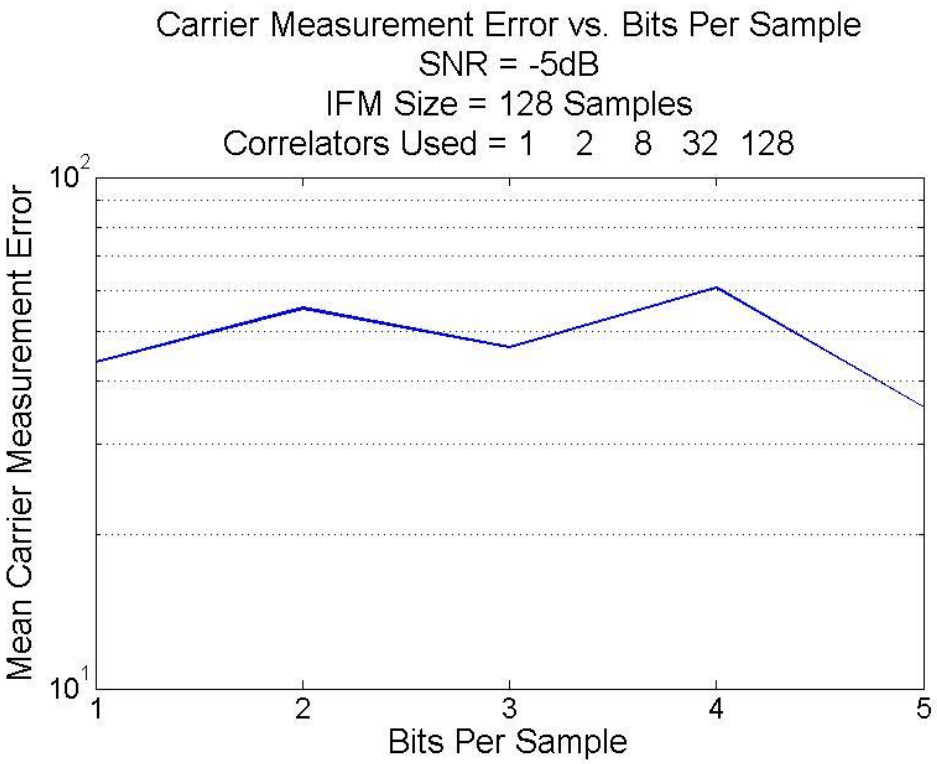

Figure 44 - Effect of bits per sample on digital IFM performance at low SNR.

However, when the number of samples collected by the IFM is increased, we can see a significant improvement in measurement performance, even at an SNR of -5dB. Fig. 45 shows the IFM performance as the size of the IFM is varied from 128 samples up to 512 samples.

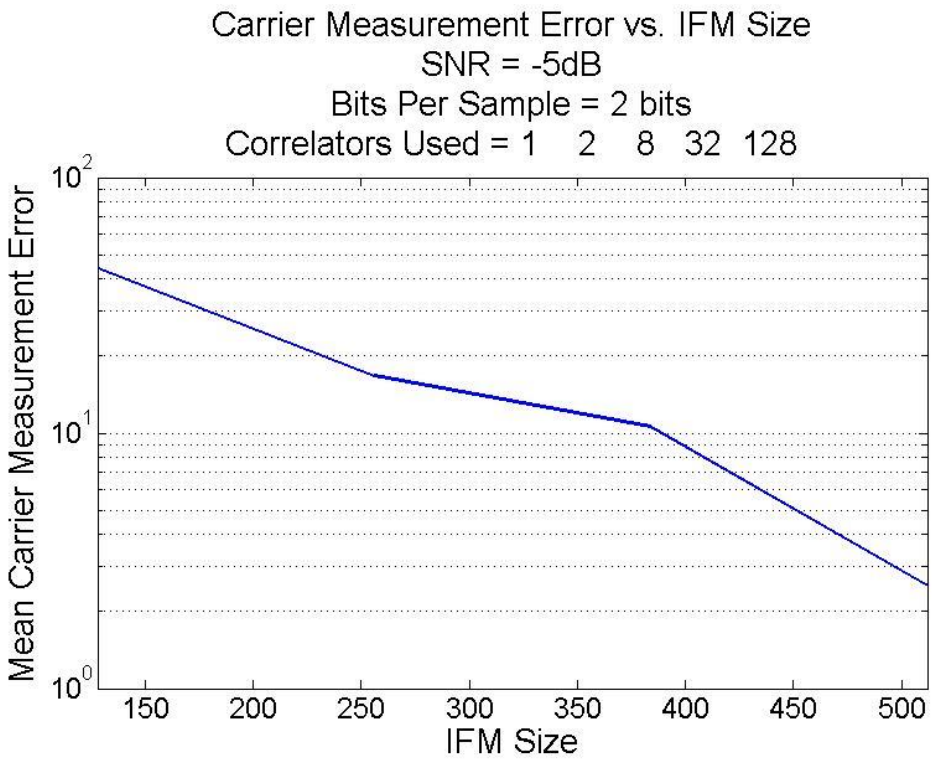

Figure 45 - Effect of number of correlated samples digital IFM performance at low SNR.

The findings from investigating the performance of the IFM resulted in a change of the original mono-bit IFM design. The design was modified to accept two bit samples, allowing for better frequency measurement capabilities. The correlator delays used from the original design we left unchanged. The IFM still collects 256 samples as it provides sufficient noise resistance at an SNR of 5dB, which is the lowest SNR that the TOA algorithm is capable of operating at.

Another important aspect of the digital IFM, which has not been mentioned yet is the detection variable, which is a measurement of the signal power. The detection variable is calculated by summing the squares of the auto correlator values as follows:

Detection Variable = S1*S1’ + S2*S2’ + S8*S8’ + S32*S32’ (28)

+S128*S128’

The detection variable from the original digital IFM design was used as a signal detector. If the detection variable crossed a threshold, a signal was considered to be present and its frequency measurement was outputted. The process to determine and set the threshold for signal detection was relatively simple, as the original design targeted stationary signals. In depth information as to how the detection variable is used for chirp signal measurement is covered in Sections VI and VII.

An important aspect to the detection variable is that its frequency response is directly affected by the frequency response of the Hilbert Transform. For this reason, the determination of any detection thresholds should only be determined when the HT and IFM are operating together. The relationship between the HT response and the detection variable can be seen in Fig. 46 and 47 respectively. Fig. 46 shows the frequency response of the HT when it is implemented by a 7-tap FIR filter with the coefficients [1 0 2 0 -2 0 -1] and the output of the detection variable as the input frequency is swept from 50MHz up to 1230MHz. Fig. 47 shows the frequency response of the HT when it is implemented by an 11-tap FIR filter with the coefficients [0.25 0 0.5 0 2 0 -2 0 -0.5 0 -0.25] and the detection variable once again as the input frequency is swept from 50MHz up to 1230MHz.

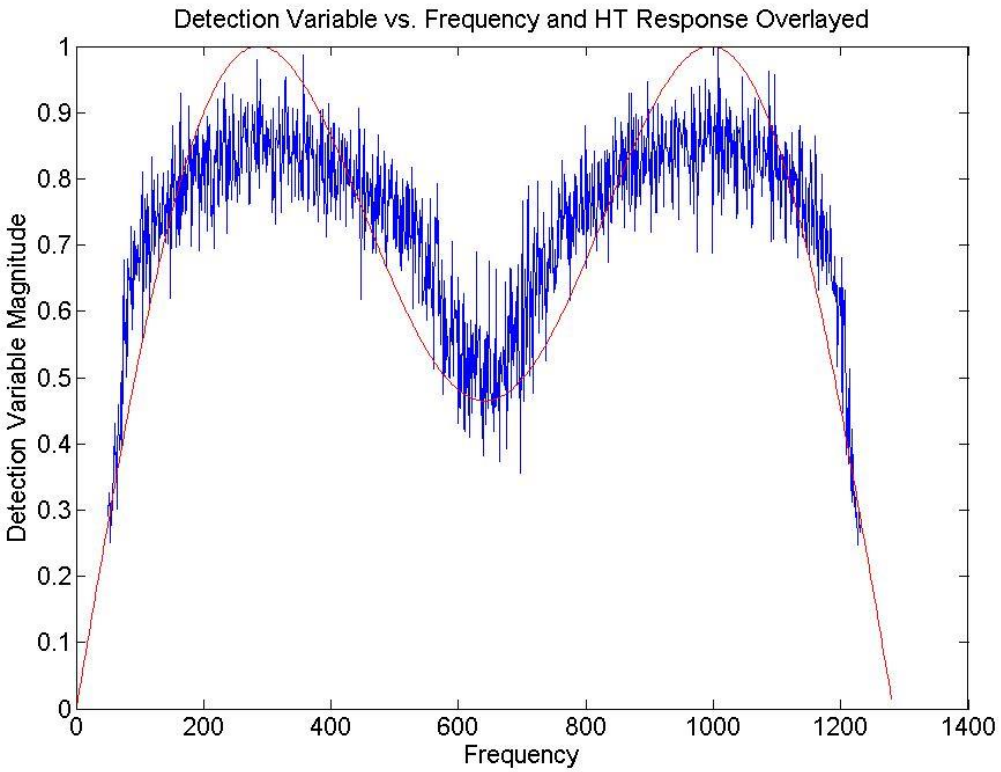

Figure 46 – Frequency response of the digital IFM detection variable when a 7-tap Hilbert Transform is utilized.

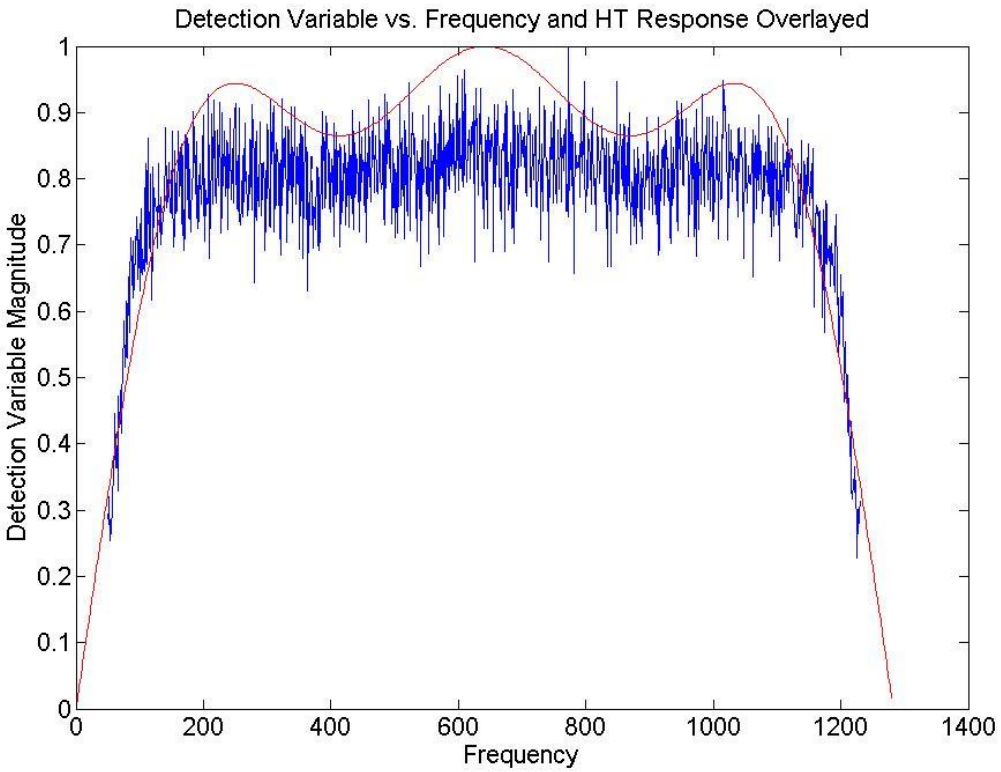

Figure 47 - Frequency response of the digital IFM detection variable when an 11-tap Hilbert Transform is utilized.

VI. Linear Chirp Receiver

6.1 Introduction

The linear chirp receiver utilizes the TOA algorithm, Hilbert Transform, and digital IFM previously discussed to accurately measure short pulse, high linear chirp rate signals. The target platform for the design is a Virtex 6 SX475 FPGA which is paired with a Calypso 12-bit ADC sampling at 2.56GSPS.

Real data is supplied to the FPGA from the Calypso ADC, which first passes through the TOA detection block. After a signal is detected, the data will pass through the HT to generate the complex signal representation. The output from the HT will enter the first digital IFM, which will measure the linear chirp rate, if one is present. Once the chirp rate has been measured, the appropriate de-chirping signal can be selected and output from a de-chirping LUT. The original signal is then mixed with the de-chirping signal, allowing for the removal of the chirp rate. Once the chirp rate has been removed, the 2nd IFM is used to measure the starting frequency. If no chirp rate is present (stationary signal), the data will pass through the first IFM and will not be de-chirped, instead the 2nd IFM will be used to measure the carrier frequency. The following sub-sections will describe each of these stages in detail. Table VII highlights the acceptable ranges for linear chirp rates and carrier frequencies that the linear chirp receiver is capable of measuring.

Table VII – Digital linear chirp receiver acceptable chirp rates and carrier frequency ranges.

Acceptable Frequency Ranges for Each Signal Type		
Signal Type	**Linear Chirp Rate**	**Starting Frequency**
Linear Chirp Signal	50MHz in 400ns - 1180MHz in 400ns	50MHz - 1180MHz
Stationary Signal	N/A	50MHz - 1230MHz

6.2 Data Acquisition

The ADC is configured to sample at 2.56 GSPS, which is a speed at which the FPGA cannot operate. To overcome this, the incoming serial data is de-multiplexed at a ratio of 1:8, allowing 8 samples to be provided in parallel at a rate of 320MHz. Only the 4 most significant bits are used by digital chirp receiver design. The TOA algorithm will utilize 4 bits to detect the TOA and TOD of a signal. The samples will then pass through the HT to form the complex signal representation. The samples will be trimmed to 2-bits after leaving the HT and will be maintained at 2-bits as they pass through the rest of the chirp receiver design. Only after a signal has been detected by the TOA algorithm and passed through the HT will any frequency measurements begin. To aid in the digital linear chirp receiver design discussion, a data flow chart for the linear chirp receiver is shown in Fig. 48.

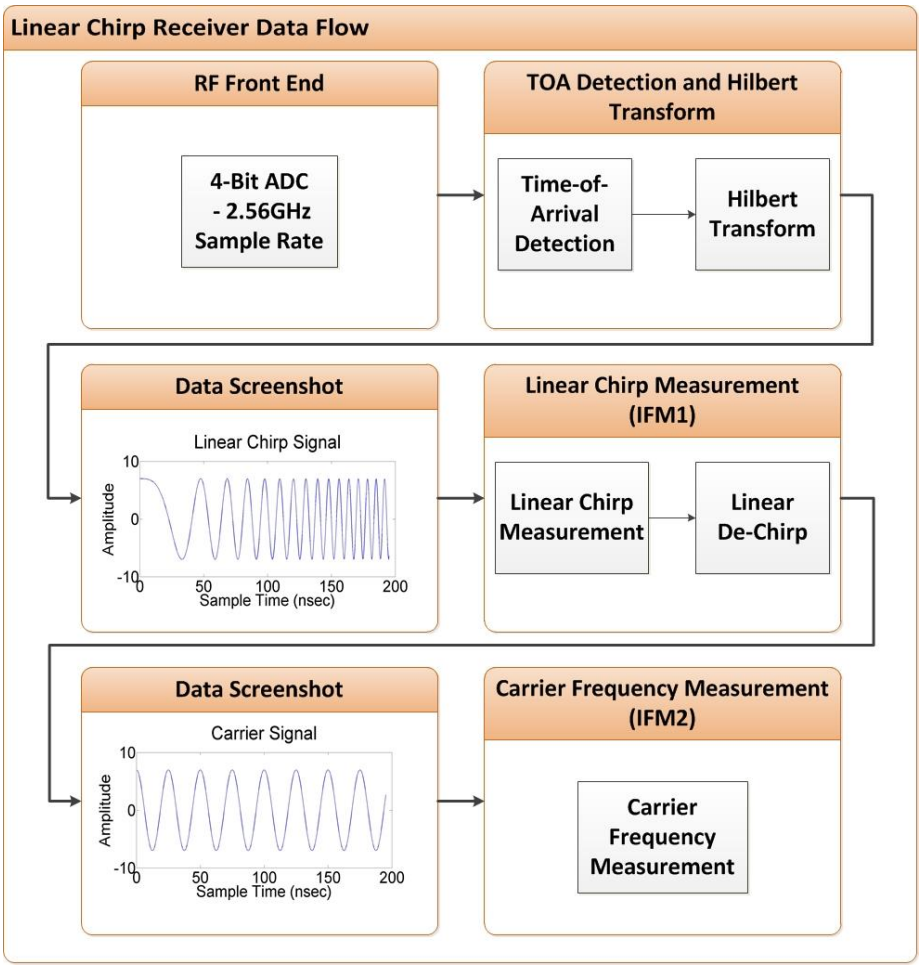

Figure 48 – Digital linear chirp receiver data flow.

6.3 Linear Chirp Rate Measurement

The complex signal generated by the HT is mixed with a delayed and conjugated copy of itself before entering the first digital IFM. The delay value chosen for this operation is vital to the accurate measurement of the linear chirp rate. To optimize chirp rate sensitivity and measurement performance, the delay length should be as long as possible. The appropriate delay value for a design can be calculated by subtracting the total measurement period from the targeted chirp pulse length. For the linear chirp receiver design, the digital IFM has a total measurement period of 150ns, (100ns to

collect 256 samples and 50ns to accommodate the longest delay for the autocorrelator) and is targeted to measure chirp pulse lengths of 400ns. The optimal delay value for mixing is therefore 400ns – 150ns = 250ns. Shorter delay values may be used if necessary, but will cause a reduction in the frequency measurement accuracy of chirp rates.

Once the original signal has been mixed with its delayed and conjugated copy, the first digital IFM is used to measure the linear chirp rate of the signal, if one is present. Along with the frequency measurement, the digital IFM will also take a signal power measurement as well as a DC power measurement. These three values will be evaluated to determine whether a linear chirp rate is present or not.

The signal power is measured by the following equation:

$$Signal\ Power = \sum_{n=1}^{256} |x(n)| \tag{29}$$

while the signal's DC component is measured by the following equation:

$$Signal\ DC\ Power = \sum_{n=1}^{256} x(n) \tag{30}$$

where x(n) is the complex signal entering the IFM. In depth details for how these measurements are used for signal classification will be provided in sub-section 6.6.

6.4 Linear De-chirping

If a linear chirp rate is detected by the first IFM, an appropriate de-chirping signal is chosen from a de-chirping LUT. The LUT contains 1024 linear de-chirping signals, which are uniformly spaced from 40MHz in 400ns up to 1190MHz in 400ns in 1.12MHz increments. Each de-chirping signal is complex signal with a length of 384 samples,

where each sample is represented by a single bit. In order to accurately measure the carrier frequency, the 2nd IFM requires the original signal to be de-chirped for 384 samples (150ns) due to the IFMs initial measurement latency of 150ns.

6.5 Carrier Frequency Measurement

The de-chirping signal is mixed with the original signal to remove the linear chirp rate from the signal. Once the signal has been de-chirped, the 2nd IFM is used to measure the carrier frequency.

If no linear chirp rate is present, the original input signal is allowed to pass through without being de-chirped, this allows stationary PW signals to be measured as well. In addition to the frequency measurement from the 2nd IFM, it will also output a signal power measurement. The frequency measurement and signal power measurement is used with the frequency measurement, signal power measurement and DC power measurement from the first IFM to correctly classify the signal and for error checking. The process for signal classification is discussed in the next sub-section.

6.6 Signal Classification

Signal classification is an important aspect to the linear chirp receiver design since no a priori knowledge is known about the incoming signals. For this reason, the presence of a linear chirp rate must be determined by evaluating the signal power measurements and frequency measurement outputs from the two IFMs. The appropriate thresholds were determined by testing both signal types: linear chirp signals and stationary signal types across their entire frequency ranges and for SNRs from 5dB up to 20dB. Only simulations in which the receiver encountered less than 10MHz in 400ns of

linear chirp rate error and 10MHz of carrier frequency error were considered for threshold determination. This prevents any possible zone mismatch errors within an IFM from skewing the threshold settings. Fig. 49 shows the entire range of values that each power measurement assumed during the simulations for each of the signal types. Linear chirp signals exhibit DC power measurements from the first IFM which are typically very small in magnitude. As a result, the power measurement from the first IFM is typically significant in magnitude. When stationary signals are considered, the opposite findings are typical, i.e., the DC measurement from the first IFM is typically very large and is paired with a signal power measurement that is small in magnitude. From the plot, we can see that the DC measurements and signal power measurements from the first IFM do not overlap for the two signal types. These findings support the notion that the power measurements are capable of distinguishing the difference between the presence of a linear chirp signal and a stationary signal type.

Since each power measurement exhibits a large range of acceptable values, each power measurement will have an upper and lower bound for each signal type. For a signal to be classified all of the power measurements much fall within the acceptable ranges and the frequency measurements must be within acceptable ranges as well. If a signal does not fall within the ranges of a signal types, the receiver will not output any measurements.

After the threshold values were determined from simulations, the linear chirp receiver was simulated once again using the new thresholds. For each signal type the entire frequency range were tested for SNRs ranging from 5dB up to 20dB. Table VIII shows the resulting signal detection rate as recorded from the TOA detection block as

well as the correct classification rates and misclassification rates from using the thresholds. The correct classification rate for linear chirp signals was simulated to be 99.98% and the correct classification rate for stationary chirp signals was simulated to be 99.99%.

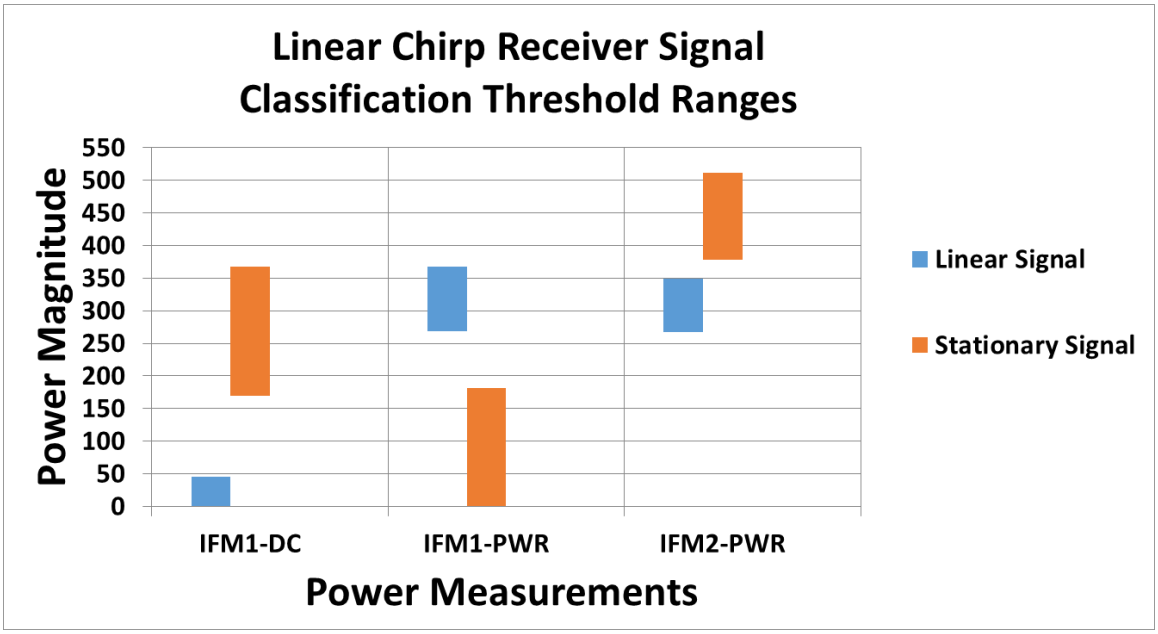

Figure 49 – Classification thresholds for digital linear chirp receiver.

Table VIII - Digital linear chirp receiver detection and correct classification rates.

Signal Type	TOA Detection Rate	Correct Classification	Misclassifications
Linear Chirp Signal	100%	99.98%	0
Stationary Signal	100%	100.00%	0
Overall	100%	99.99%	0

VII. Nonlinear Chirp Receiver

7.1 Introduction

The nonlinear chirp receiver utilizes the TOA algorithm, Hilbert Transform, and digital IFM previously discussed to accurately measure short pulse, high chirp rate nonlinear signals. The target platform for the design is a Virtex 6 SX475 FPGA which is paired with a Calypso 12-bit ADC sampling at 2.56GSPS.

Real data is supplied to the FPGA from the Calypso ADC which will first pass through the TOA detection block. After a signal has been detected, the data will pass through the HT to generate the complex signal representation. The output from the HT will enter the first digital IFM, which will measure the nonlinear chirp rate, if one is present. Once the nonlinear chirp rate has been measured, the appropriate de-chirping signal can be selected and output from the nonlinear de-chirping LUT. The original signal is then mixed with the de-chirping signal, allowing for the removal of the nonlinear chirp rate. Once the nonlinear chirp rate has been removed, the 2^{nd} IFM is used to measure the linear chirp rate, if one is present. After measuring the linear chirp rate, the appropriate linear de-chirping signal can be generated from a linear de-chirping LUT. The de-chirping signal will be combined with the signal outputted from the first IFM to remove the linear chirp rate. Once both the nonlinear chirp rate

and linear chirp rate have been measured and removed, the third IFM is used to measure the starting frequency. If no chirp rate is present, the data will pass through the design and will not be de-chirped, instead the 3rd IFM will be used to measure the carrier frequency. The following sub-sections will describe each of these stages in detail.

7.2 Data Acquisition

The ADC is configured to sample at 2.56 GSPS, which is a speed at which the FPGA cannot operate. To overcome this, the incoming serial data is de-multiplexed at a ratio of 1:8, allowing 8 samples to be provided in parallel at a rate of 320MHz. Only the 4 most significant bits are used by digital chirp receiver design. The TOA algorithm will utilize 4 bits to detect the TOA and TOD of a signal. The samples will then pass through the HT to form the complex signal representation. The samples will be truncated to 2-bits after leaving the HT and will be kept as they pass through the rest of the chirp receiver design. Only after a signal has been detected by the TOA algorithm and passed through the HT will any frequency measurements begin. To aid in visualizing the differences between the linear receiver and nonlinear receiver, fig. 50 shows the flow of data as it passes through the nonlinear chirp receiver. Table IX shows the acceptable frequency ranges for the four different signal types.

Table IX - Digital nonlinear chirp receiver acceptable chirp rates and carrier frequency ranges.

Acceptable Frequency Ranges for Each Signal type			
Signal type	**Nonlinear Chirp Rate**	**Linear Chirp Rate**	**Starting Frequency**
Nonlinear Chirp Signal	50MHz - 1180MHz**	N/A	50MHz - 1180MHz
Linear Chirp Signal	N/A	50MHz - 1180MHz*	50MHz - 1180MHz
Nonlinear & Linear Chirp Signal	50MHz - 1130MHz**	50MHz - 1130MHz*	50MHz - 1130MHz
Stationary Signal	N/A	N/A	50MHz - 1230MHz
** Chirp rate period is 400ns^2			
* Chirp rate period is 400ns			

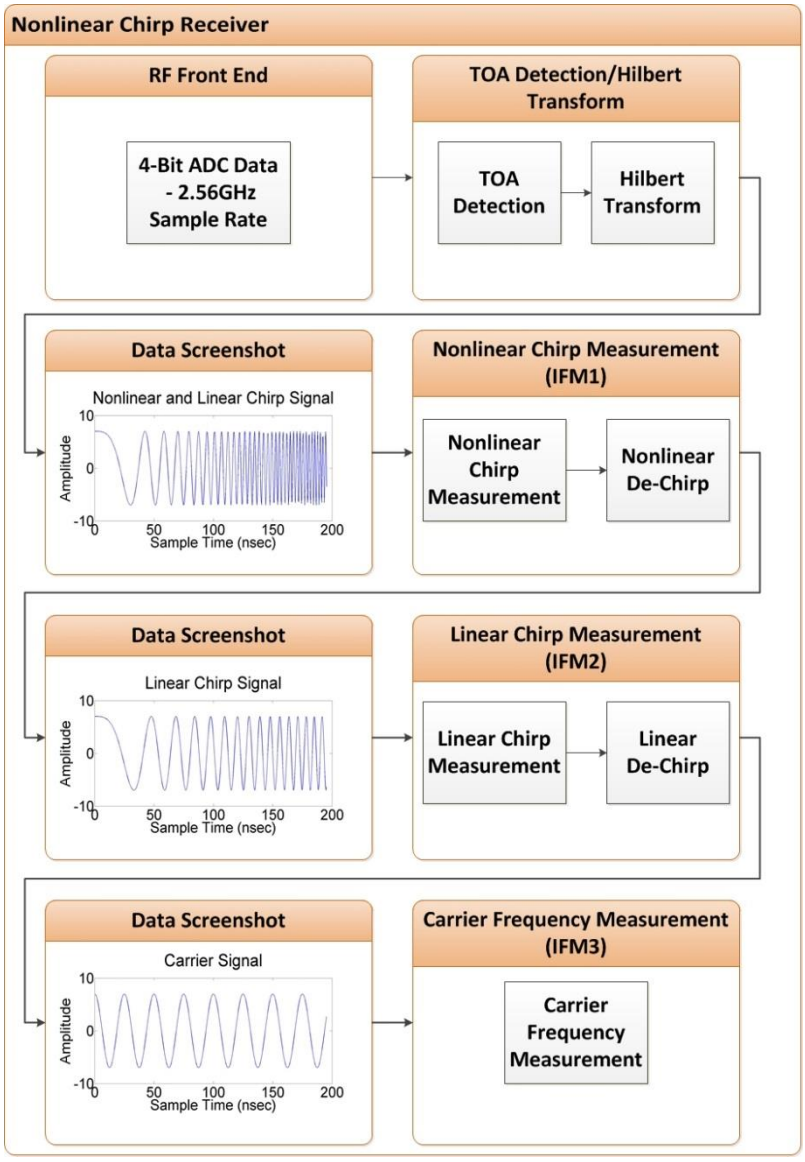

Figure 50 - Digital nonlinear chirp receiver data flow.

7.3 Nonlinear Chirp Rate Measurement

In order to isolate the nonlinear chirp rate, the complex signal output from the HT must be mixed with a delayed and conjugated copy of itself. The resulting signal must be mixed with a delayed and conjugated copy of itself once more.

In order to extract the highest nonlinear chirp rate sensitivity, the longest possible delay should be used for the two mixing processes. The longest possible delay is calculated by subtracting the IFM measurement latency period from the target pulse period. The target pulse period is 400ns and the IFM has a latency of 150ns. The remaining period for the time delay is 250ns, which is to be split between the two mixing processes. The optimal ratio for these delays is 1:2. Based on this, the first mixing process uses a delay of ~83.33ns while the other mixing process uses a delay of ~166.6ns.

After the incoming signal has been mixed twice, the data enters the first IFM to measure the nonlinear chirp rate, if one is present.

7.4 Nonlinear De-chirping

If a nonlinear chirp rate is detected, the appropriate nonlinear de-chirping signal is selected from a LUT which contains 1024 de-chirping signals, spaced apart in 1.12MHz increments from 40MHz in $400ns^2$ up to 1,190MHz in $400ns^2$. The de-chirping signal will be mixed with the original incoming signal to remove the nonlinear chirp rate. If a nonlinear chirp rate is not detected, the original signal is allowed to pass through the first IFM unaltered.

7.5 Linear Chirp Rate Measurement

Once the nonlinear de-chirping process has completed, the linear chirp measurement process will begin. To isolate the linear chirp rate, the incoming signal is mixed with a delayed and conjugated copy of itself. The longest delay value possible should be chosen in order to maximize the sensitivity of the chirp measurement. The optimal delay should therefore equal to the total measurement latency minus the pulse width.

For the nonlinear chirp receiver however, the delay length chosen inherently dictates the size of the nonlinear de-chirping LUT. If the optimal delay of 250ns (400ns pulse length – 150ns measurement latency = 250ns) is chosen, then the nonlinear de-chirping table must store 1024 complex signals of length of 400ns (1,024 samples). The size of the LUT can be reduced by over 31% if a delay value of 125ns is chosen instead.

Using a delay of 125ns will require the nonlinear de-chirping table to store de-chirping signals of length 275ns (150ns for the IFM + 125ns for the delay). The tradeoff of using a shorter delay value is a reduced linear chirp measurement performance.

After the linear mixing process is complete, the data passes into the 2^{nd} IFM where the linear chirp rate is measured, if one is present.

7.6 Linear De-chirping

If a linear chirp rate is detected, a 2^{nd} LUT is used to generate a de-chirping signal, which will be mixed with the signal from the first IFM to remove the linear chirp rate. The 2^{nd} LUT contains 1024 de-chirping signals, spaced uniformly from 40MHz in

400ns up to 1,190MHz in 400ns. If no linear chirp rate is detected, the signal will pass through unmodified into the 3rd stage of the chirp receiver.

7.7 Carrier Frequency Measurement

After the signal has been appropriately de-chirped or allowed to pass through, the 3rd digital IFM is used to measure the carrier frequency. After all frequency measurements are completed, a final classification block is used to verify the presence of the measured chirp parameters. An in depth discussion on classification follows in the next section.

7.8 Signal Classification

Signal classification takes place after all frequency measurements and signal power measurements have been completed. The classification block in the nonlinear chirp receiver utilizes 7 power measurements in addition to the 3 frequency measurements in order to correctly classify incoming signals. Recall that to measure the nonlinear component requires two separate mixing operations. A DC power measurement and signal power measurement is taken after each mixing operation resulting in two DC measurements and two signal power measurements.

To measure the linear frequency component requires a single mixing operation, therefore a single DC power measurement and signal power measurement is taken. Since the measurement of the carrier frequency does not require a mixing process, only a single power measurement is taken from the third IFM, resulting in a total of 7 power measurements.

Four possible signal types are considered for classification: nonlinear chirp signal, linear chirp signal, simultaneous nonlinear and linear chirp signal, and a stationary signal. Each signal type uses its own set of thresholds for each power and frequency measurement. If all of the measurements fall within an acceptable range for a given signal type, the signal will be classified as that type. If the measurements do not meet all the criteria for any signal type, the receiver reports no output.

A lower and upper threshold is stored in the receiver for each measurement and signal type combination. Fig. 51 shows the acceptable range for each measurement and signal type.

Threshold values were determined by simulating each signal type throughout its acceptable frequency ranges for SNRs ranging from 5dB to 20dB. Only simulations in which the parameter estimation error was small were used for threshold determination. The lowest simulated values from the simulations were used for the lower thresholds, while the highest values were used for the upper thresholds. A total of $3 * 10^6$ simulations were run for each signal type, resulting in a total of $12 * 10^6$ simulations.

Utilizing the thresholds found from simulations, the nonlinear receiver achieved an overall correct classification rate for the four signal types of 98.99% for SNRs between 5dB and 20dB. Detailed classification rates can be seen in Table X.

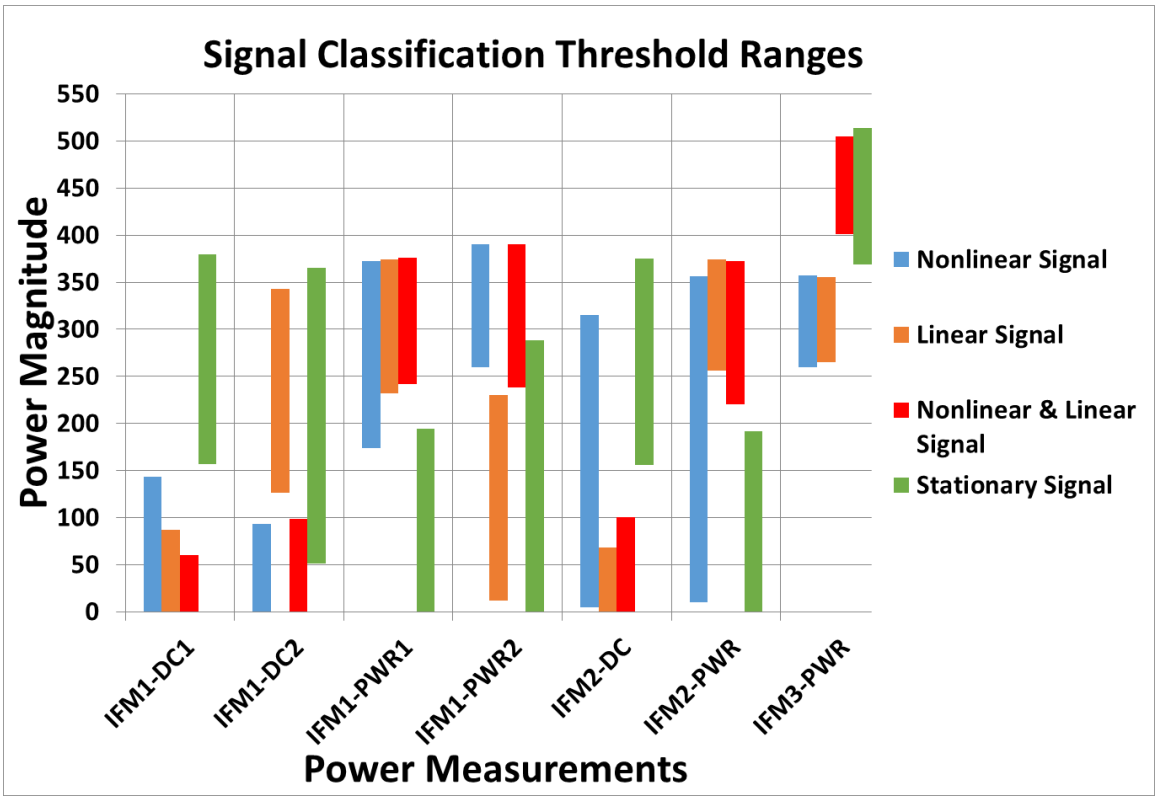

Figure 51 – Classification thresholds for the digital nonlinear chirp receiver (DC denotes DC power measurement and PWR denotes signal power measurement)

Table X - Digital nonlinear chirp receiver detection and correct classification rates.

Signal Type	Correct Classification	Misclassifications
Nonlinear Chirp Signal	97.69%	0
Linear Chirp Signal	99.51%	0
Nonlinear & Linear Chirp Signal	99.09%	0
Stationary Signal	99.67%	0
Overall	98.99%	0

VIII. Variable Chirp Receiver

8.1 Data Acquisition

The variable chirp receiver utilizes the TOA algorithm, Hilbert Transform, and digital IFM previously discussed to accurately measure variable pulse length linear chirp signals. The target platform for the design is a Virtex 6 SX475 FPGA, which is paired with a Calypso 12-bit ADC.

Real data is supplied to the FPGA from the Calypso ADC at a rate of 2.56GSPS. The data is de-multiplexed at a ratio of 1:8, setting the FPGA operating frequency at 320MHz. Real data from the ADC will first pass through the TOA detection block, which will detect the beginning and the end of a signal. After detection, the data will pass through the HT to generate the complex signal representation. The detection of the beginning of the pulse and the end of the pulse is used to control a set of FIFO's and a FILO (First In, Last Out), which will collect the first 384 samples of the signal as well as the last 384 samples of the signal. The use of FIFO's allows for a variable amount of delay, which optimizes the chirp rate measurement for various pulse widths. The chirp rate and carrier frequency can be measured simultaneously, however, a frequency correction based on the measured pulse width will need to be applied to both measurements. The correction factor for the carrier frequency is dependent on the measurement of the chirp rate. An abstract data flow for the

variable chirp receiver can be seen in Fig. 52.

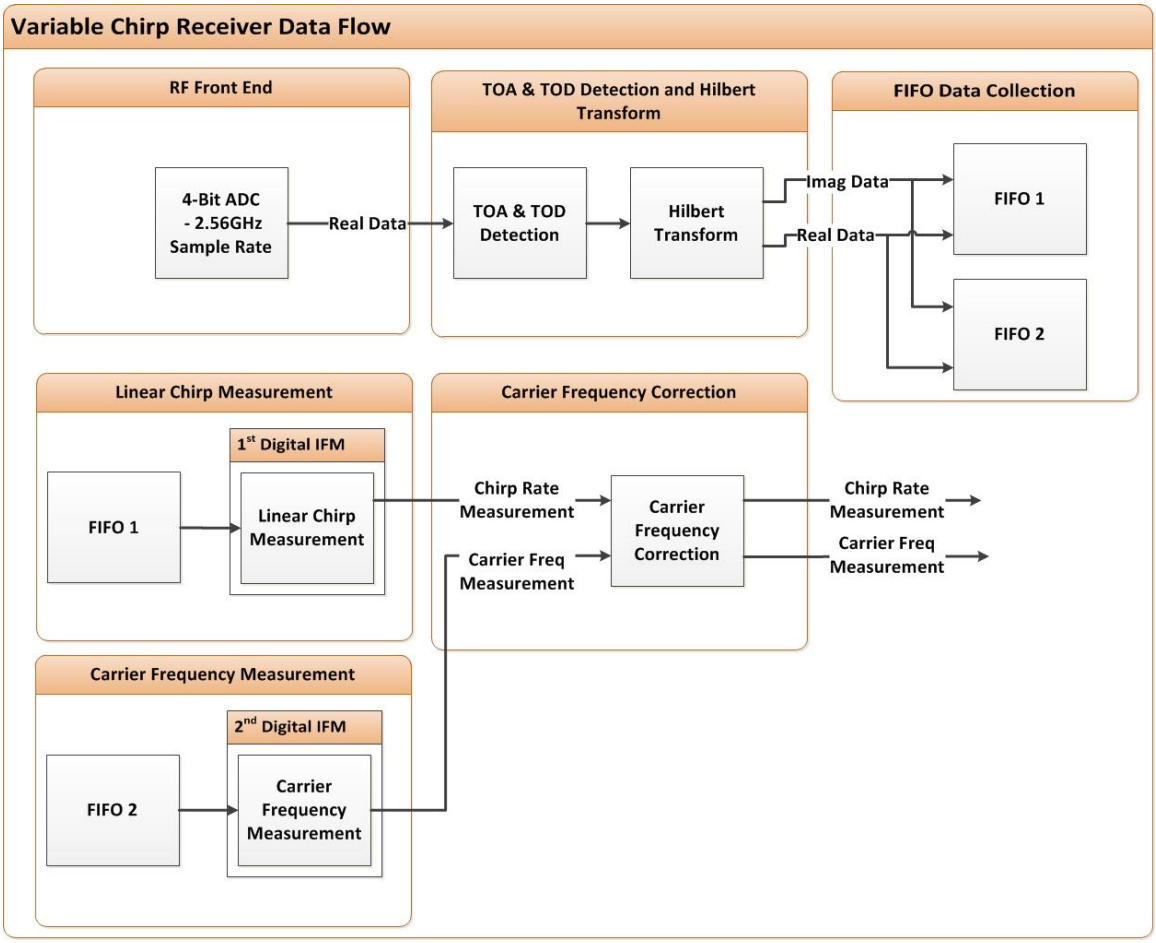

Figure 52 – Digital variable chirp receiver data flow.

8.2 Linear Chirp Rate Measurement

Measurement of the linear chirp rate in the variable chirp receiver is similar to the process used by the linear chirp receiver. However, instead of utilizing a static delay length, a set of FIFOs is used to collect the first 384 samples and the last 384 samples of a signal. This approach is made possible by the high accuracy of the TOA and TOD measurements from the TOA algorithm. The first 384 samples are element wise

multiplied by the conjugate of the last 384 samples. The resulting signal is used by the first digital IFM to measure the linear chirp rate, if one is present.

The same digital IFMs which are utilized in both the linear chirp receiver and nonlinear chirp receiver, are used in the variable chirp receiver. However, the chirp rate calculation is adjusted based on the measured pulse length, reported by the TOA algorithm. The equation for correcting the chirp rate measurement can be seen in Eq. (28) in Section V. The original IFM was calibrated to measure linear chirp rates based on a pulse width of 400ns (1024 samples at 2.56GHz) and a delay length of 250ns (640 samples at 2.56GHz). The measured pulse width of the signal is used to determine the ratio between the actual pulse width and the original pulse width of 400ns. This ratio is multiplied by the measured chirp rate, which is provided by the IFM to calculate the correct chirp rate, which is then output by the receiver.

$$\text{Corrected Chirp Rate} = \textit{Chirp Rate} * \frac{\left(\frac{\textit{Pulse Width}}{\textit{Pulse Width} - 384}\right)}{\left(\frac{1024}{640}\right)} \tag{31}$$

8.3 Carrier Frequency Measurement

Measurement of the carrier frequency in the variable chirp receiver is also similar to the process used by the linear chirp and nonlinear chirp receivers. The digital IFM used to measure the carrier frequency remains unchanged, however two FIFOs are used to supply the IFM with the necessary data. One of the FIFOs stores the first 384 elements in their original order, while the second FIFO stores the first 384 elements in reverse order. The outputs from the two FIFOs are element wise multiplied together, where the conjugate of one of the FIFOs data is taken. The resulting signal is passed as input into

the digital IFM, which measures the carrier frequency.

Since the variable chirp receiver does not attempt to de-chirp incoming chirp signals, a correction factor must be applied to the carrier frequency. The correction factor depends on the length of the pulse, as well as the chirp rate measured by the first digital IFM. The equation to correct the carrier frequency can be seen in Eq. (29).

Corrected Carrier Frequency = (32)

$$\frac{Carrier\ Frequency - Corrected\ Chirp\ Rate * (\frac{384}{Pulse\ Width})}{2}$$

IX. Performance Evaluations

9.1 Matlab / Simulink Based Simulations Overview

For all three digital chirp receiver designs, the Matlab / Simulink based simulations were setup in an identical manner. For each SNR from 5dB up to 20dB, the entire acceptable frequency range for each signal type was tested in 0.25MHz increments. For nonlinear chirp signals, this resulted in testing nonlinear chirp rates from 50MHz in $400ns^2$ up to 1180MHz in $400ns^2$ while the starting frequency was simultaneously swept from 1180MHz down to 50MHz in 0.25MHz steps. This resulted in a total of 4,520 signals being simulated for each SNR. Linear chirp signals were simulated in an identical manner.

In the case of combination nonlinear and linear chirp signals, the nonlinear chirp rate and linear chirp rate were simultaneously swept in opposing directions while the carrier frequency was kept constant. Thus, nonlinear chirp rates were swept from 50MHz in $400ns^2$ up to 1130 MHz in $400ns^2$ while linear chirp rates were simultaneously swept from 1130MHz in 400ns down to 50MHz in 400ns and the carrier frequency was kept constant at 50MHz. This resulted in a total of 4,320 simulations per SNR.

In the case of stationary signals, carrier frequencies were swept from 50MHz up to 1230MHz in 0.25MHz increments, resulting in a total of 4,720 simulations per SNR.

The variable chirp receiver required additional simulations beyond what the linear chirp receiver and nonlinear chirp receiver required as multiple pulse widths needed to be

simulated. For the variable chirp receiver simulations, pulse widths ranging from 400ns up to 4,000ns were tested in 400ns increments. The simulations conducted at each pulse width were identical to those used for the original linear chirp receiver.

9.2 FPGA Based Simulations Overview

Once all of the designs had been fully synthesized and all timing constraints were met, the designs were loaded onto a Virtex 6 SX475 FPGA. The board used for testing is a Microtek board, which interfaces a Calypso 12-bit 2.56GSPS ADC with the Xilinx Virtex 6 FPGA. Simulating the designs and gathering the results from the FPGA board required the use of 3 separate signal generators. One of the signal generators provided the 1.28GHz clock for the FPGA, setting the sample rate of the ADC. The ADC operated in dual edge triggered mode, thus a sample was collected every rising and falling edge of the clock.

The second signal generator provided the 100MHz clock required for the Ethernet communication between the FPGA and PC. The outputs from the digital chirp receivers were temporarily stored in FIFOs until they became full. Once filled, they would transmit their contents over Ethernet to the PC where they were written to a text file and stored on the hard drive.

A third signal generator was used to provide the high chirp rate signals used as input into the digital chirp receivers. This signal generator was a 12 GSa/S Arbitrary Waveform Generator, Keysight Model M8190A. The signal generator came paired with a Windows based PC, which also had Matlab installed. This allowed for the easy creation of arbitrary waveforms within Matlab, which were downloaded into the signal generator's DAC. The DAC is accompanied by a large memory module, which allows for

the storage of up to 1 second's worth of samples at the DAC's sampling rate of 12GSa/s.

Simulations for all three digital chirp receivers were setup identically to the original Matlab based simulations. AWGN was added to each of the signals in Matlab before loading them into the arbitrary waveform generator's DAC. SNRs ranging from 5dB up to 20dB were tested.

For the linear chirp receiver and the nonlinear chirp receiver a train of pulses were generated for each SNR, allowing all frequencies to be measured at each SNR. The pulse length for all signals was kept at 400ns, and the pulse repetition interval (PRI) was set at a constant 500ns. This was necessary as no external sync was used between the FPGA and the arbitrary waveform generator. By keeping the pulse width and PRI constant, it allowed for an accurate measurement of the detection rate for the receivers as well as preventing any frequency measurement bias from being injected into the results.

In addition to keeping the pulse width and PRI constant, a set of sync frequencies were also used to determine the beginning of a simulation set. This was important as the arbitrary waveform generator would continuously repeat the train of pulses, which were stored in its memory. A set of stationary carrier frequencies at a high SNR and pre-determined frequencies were used to determine the start of the pulse train.

This sync pattern was used by a Matlab script after the receiver data had been written to text files. The scripts were able to find the sync patterns allowing for the frequency measurement performance to be accurately measured.

For the variable chirp receiver design, the pulse width and PRI were kept constant, dependent on the current pulse width being simulated. The PRI was always set to be 100ns longer than the pulse width. Pulse widths ranging from 400ns up to 4,000ns

were simulated in 400ns increments.

9.3 Linear Chirp Receiver Matlab/Simulink

From the simulations, the digital linear chirp receiver obtained an average linear chirp measurement error of 0.30MHz in 400ns with a standard deviation of 0.23MHz in 400ns. The linear chirp receiver measured the starting frequency of linear chirp signals with an average measurement error of 1.80MHz and a standard deviation of 1.87MHz. For non-chirp stationary signals the average carrier frequency error was simulated to be 0.26MHz with a standard deviation of 0.20MHz.

Fig. 53 plot (a) shows the mean error in MHz per 400ns for linear chirp rate measurement error. Plot (b) shows the mean carrier frequency error in MHz for both linear chirp signal and stationary signal types. Table XI summarizes the performance of the linear chirp receiver for all signal types.

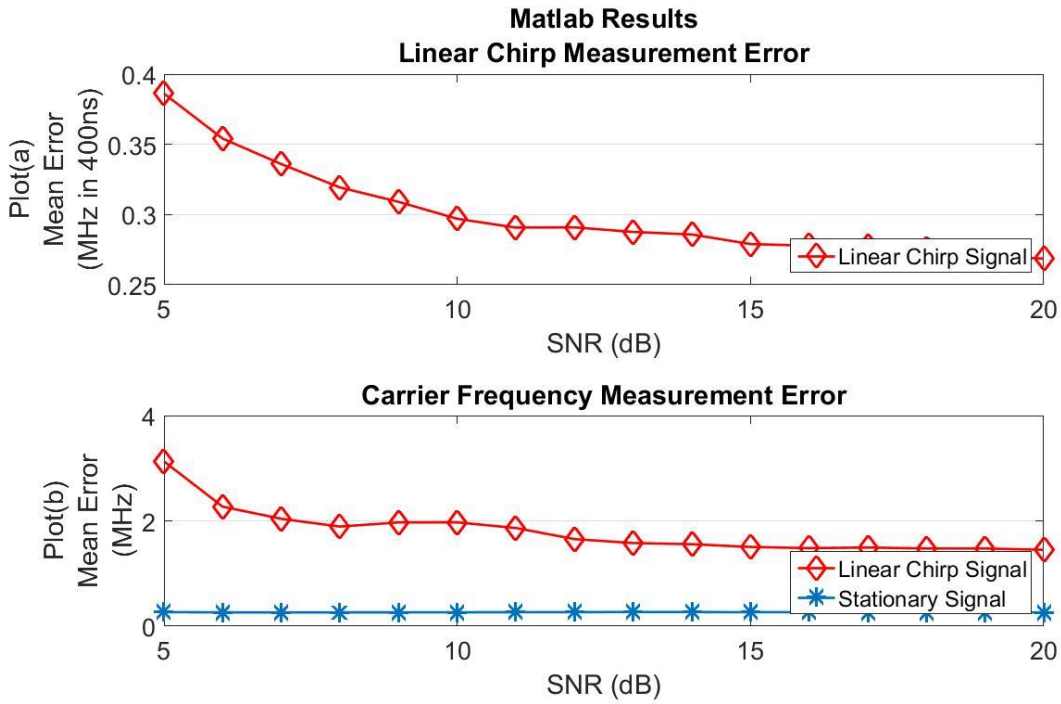

Figure 53 – Linear chirp receiver Matlab based frequency measurement performance.

Table XI - Linear chirp receiver Matlab based frequency measurement performance

Signal Type	Linear Chirp Rate Avg. Error	Linear Chirp Rate Std. Dev.	Carrier Frequency Avg. Error	Carrier Frequency Std. Dev.
Linear Chirp	0.30Mhz in 400ns	0.23Mhz in 400ns	1.80MHz	1.87MHz
Stationary	N/A	N/A	0.26MHz	0.20MHz

9.4 Linear Chirp Receiver FPGA

The linear chirp receiver was synthesized using Xilinx ISE and targeted for a Virtex 6 SX475 FPGA. Fig. 54 shows a highlight of the synthesis report for the design. The digital linear chirp receiver design only utilizes 9% of the total resources available and meets all timing constraints at a rate of 320MHz.

Project File:	Linear_Chirp_Receiver_43.xise	**Parser Errors:**	No Errors
Module Name:	toplevel_adc	**Implementation State:**	Programming File Generated
Target Device:	xc6vsx475t-2ff1759	▪ **Errors:**	No Errors
Product Version:	ISE 13.4	▪ **Warnings:**	1825 Warnings (0 new)
Design Goal:	Timing Performance	▪ **Routing Results:**	All Signals Completely Routed
Design Strategy:	Performance with IOB Packing	▪ **Timing Constraints:**	All Constraints Met
Environment:	System Settings	▪ **Final Timing Score:**	0 (Timing Report)

Device Utilization Summary				[-]
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	23,792	595,200	3%	
Number used as Flip Flops	23,782			
Number used as Latches	4			
Number used as Latch-thrus	0			
Number used as AND/OR logics	6			
Number of Slice LUTs	27,790	297,600	9%	
Number used as logic	23,718	297,600	7%	
Number using O6 output only	17,482			
Number using O5 output only	730			
Number using O5 and O6	5,506			
Number used as ROM	0			
Number used as Memory	3,473	122,240	2%	
Number used as Dual Port RAM	196			
Number using O6 output only	68			
Number using O5 output only	20			
Number using O5 and O6	108			
Number used as Single Port RAM	0			
Number used as Shift Register	3,277			
Number using O6 output only	3,277			
Number using O5 output only	0			
Number using O5 and O6	0			
Number used exclusively as route-thrus	599			
Number with same-slice register load	555			
Number with same-slice carry load	44			
Number with other load	0			
Number of occupied Slices	14,228	74,400	19%	

Figure 54 - Linear chirp receiver FPGA based frequency measurement performance.

From the FPGA tests, the digital linear chirp receiver obtained an average linear chirp measurement error of 0.35MHz in 400ns with a standard deviation of 0.39MHz in 400ns. The linear chirp receiver measured the starting frequency of linear chirp signals with an average measurement error of 1.75MHz and a standard deviation of 2.09MHz. For non-chirp stationary signals the average carrier frequency error was simulated to be

0.33MHz with a standard deviation of 0.40MHz.

Fig. 55 plot (a) shows the mean error in MHz per 400ns for linear chirp rate measurement error. Plot (b) shows the mean carrier frequency error in MHz for both linear chirp signal and stationary signal types. Table XII summarizes the performance of the linear chirp receiver for all signal types.

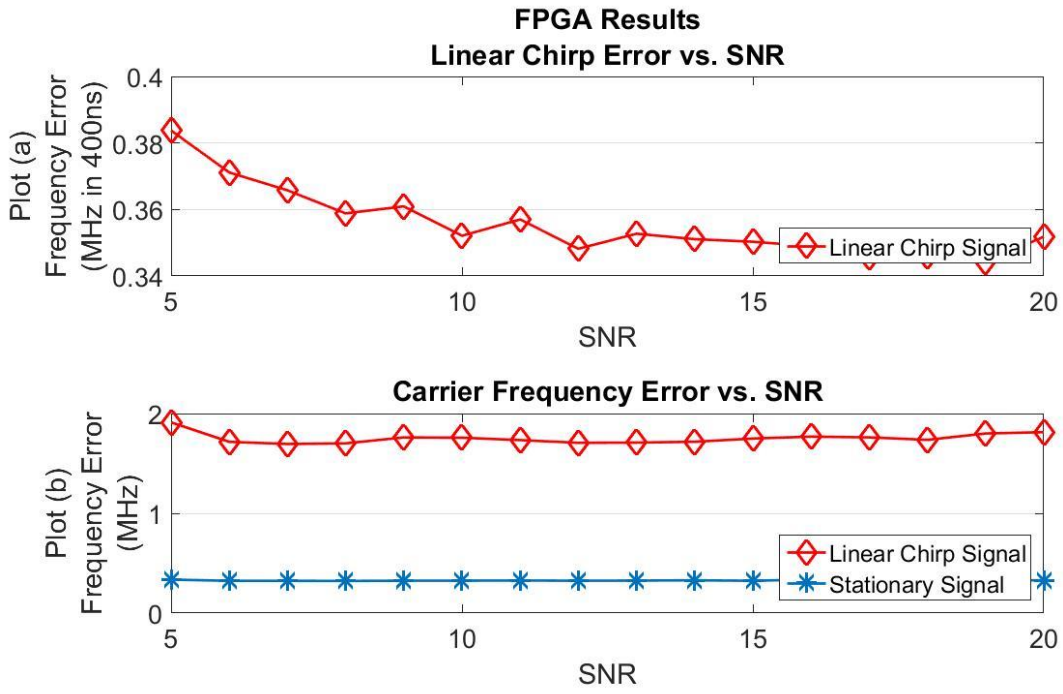

Figure 55 - Linear chirp receiver FPGA based frequency measurement performance.

Table XII - Linear chirp receiver FPGA based frequency measurement performance

Signal Type	Linear Chirp Rate Avg. Error	Linear Chirp Rate Std. Dev.	Carrier Frequency Avg. Error	Carrier Frequency Std. Dev.
Linear Chirp	0.35Mhz in 400ns	0.39Mhz in 400ns	1.75MHz	2.09MHz
Stationary	N/A	N/A	0.33MHz	0.40MHz

In addition to recording the frequency measurement performance, data was also collected and analyzed to properly adjust the receiver thresholds for signal classification based on the sampled data. After adjusting the thresholds, detection rates and misclassification rates were re-evaluated for the FPGA based linear chirp receiver. Fig. 56 shows the threshold ranges for the FPGA based linear chirp receiver for all three signal power measurements. Fig. 57 shows the achieved detection rates for both linear chirp signals and stationary signals across all the tested SNRs. Table XIII shows the average detection rates and also shows that no misclassification errors were encountered for all 800,000 signals tested on the FPGA design. The overall detection rate was maintained above 99% for both signal types.

The FPGA based linear chirp receiver achieved similar detection rates as predicted by the original Matlab simulations. The Matlab simulations predicted a correct detection rate of 99.99% and the FPGA based design achieved an overall correct detection rate of 99.27%.

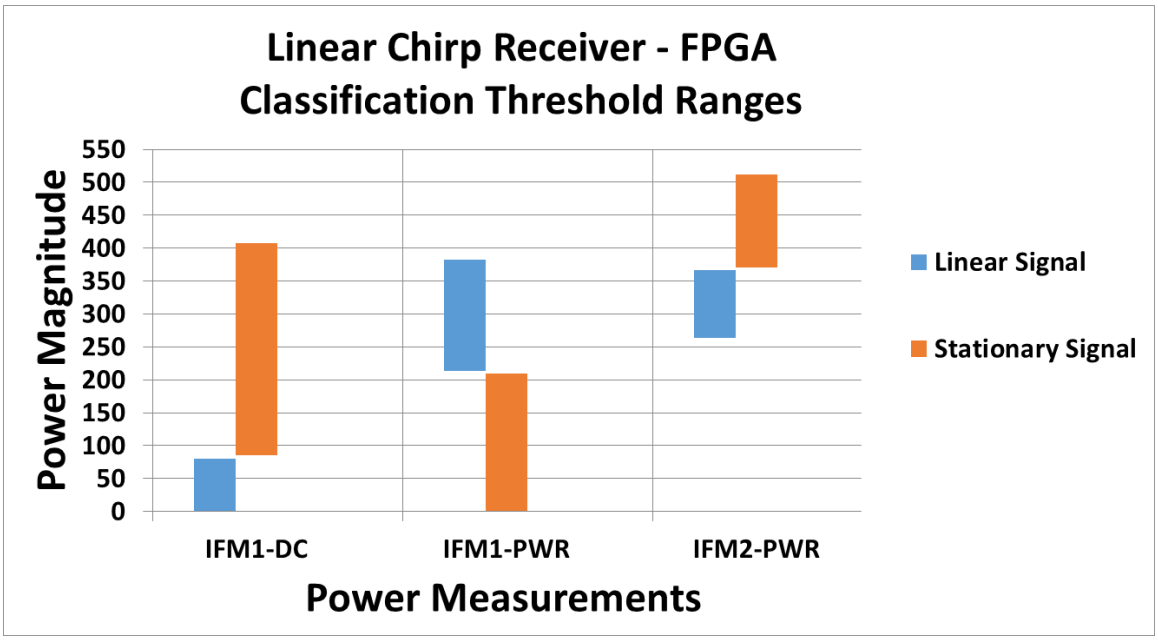

Figure 56 - Linear chirp receiver FPGA based thresholds for signal classification.

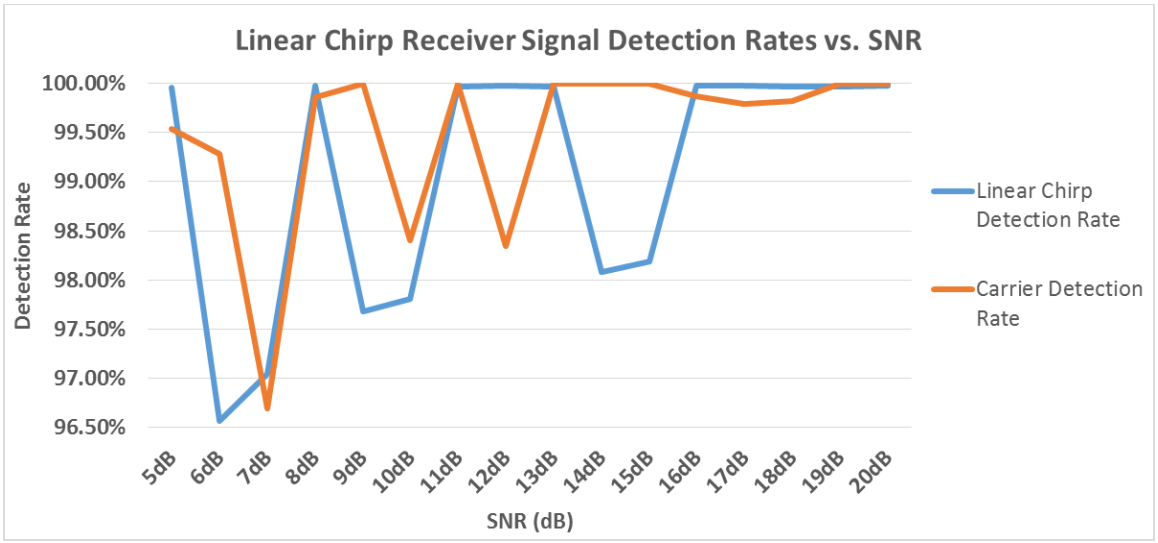

Figure 57 - Linear chirp receiver FPGA based detection rates vs. SNR.

Table XIII - Linear chirp receiver FPGA based detection and correct classification rates.

	Detection Rate	Misclassifications	Number of Simulations
Linear Chirp Signal	99.07%	0	418,898
Stationary Signal	99.47%	0	385,401
Overall	99.27%	0	804,299

9.5 Relevant Linear Chirp Receiver Comparisons

Table XIV compares the results presented in this research with the results obtained from the mono-bit linear chirp receiver in [27] and the digital channelized receiver in [28]. The digital linear chirp receiver presented in this dissertation achieves significantly better performance for both linear chirp measurement and carrier frequency measurement of linear chirp signals. It should also be noted that the mono-bit linear chirp receiver in [27] assumes perfect TOA detection, whereas TOA is estimated with the digital linear chirp receiver and the digital channelized receiver. This significantly degrades the performance of carrier frequency measurements. Though the chirp rates researched in [28] are significantly slower than the chirp rates presented in this paper, the high linear chirp receiver achieves significantly better measurement results.

Table XIV – Comparison of chirp rate and carrier frequency measurement abilities of various receivers

	Linear Chirp Range	**Linear Chirp Error**
Digital Linear Chirp Receiver	50MHz in 400ns - 1180MHz in 400ns	0.08%
Mono-bit Linear Chirp Receiver [27]	80MHz in 400ns - 1600MHz in 400ns	< 2%
Digital Channelized Receiver [28]	0.4MHz in 400ns - 4MHz in 400ns	1% - 10%
	Carrier Range	**Carrier Error**
Digital Linear Chirp Receiver	50MHz - 1230MHz	0.68%
Mono-bit Linear Chirp Receiver [27]	60MHz - 1.2GHz	< 4%
Digital Channelized Receiver [28]	500ns pulse - 1μs pulse	10%
*The original chirp period for the mono-bit linear chirp receiver was 300ns. This value was converted to its equivalent chirp in 400ns for ease of comparison.		
** The original chirp period for the digital channelized receiver was 1us. This value was converted to its equivalent chirp in 400ns for ease of comparison.		

9.6 Nonlinear Chirp Receiver Matlab/Simulink

The simulations for the digital nonlinear chirp receiver test four different signal types; nonlinear chirp signals, linear chirp signals, simultaneous nonlinear and linear chirp signals and stationary carrier signals.

For nonlinear chirp signals, the receiver measured nonlinear chirp rates with an average error of 1.27MHz in $400ns^2$ and a standard deviation of 0.99MHz in $400ns^2$. Starting frequencies were measured with an average error of 0.77MHz and a standard deviation of 0.69MHz.

For linear chirp signals, the receiver measured linear chirp rates with an average error of 0.46MHz in 400ns and a standard deviation of 0.36MHz in 400ns. Starting frequencies were measured with an average error of 1.70MHz and a standard deviation of 1.80MHz.

For simultaneous nonlinear and linear chirp signals, the receiver measured nonlinear chirp rates with an average error of 1.26MHz in $400ns^2$ and a standard

deviation of 0.98MHz in 400ns^2. Linear chirp rates were measured with an average error of 2.80MHz in 400ns and a standard deviation of 2.75MHz in 400ns. Starting frequencies were measured with an average error of 1.44MHz and a standard deviation of 1.42MHz.

For stationary carrier signals, the receiver measured carrier frequencies with an average error of 0.26MHz and a standard deviation of 0.19MHz.

Fig. 58 plot (a) shows the mean error in MHz per 400ns^2 for nonlinear chirp rate measurement error. Plot (b) shows the mean error in MHz per 400ns for linear chirp rate measurement error. Plot (c) shows the mean error in MHz for starting frequency / carrier frequency measurement error. Table XV summarizes the performance of the nonlinear chirp receiver for all signal types.

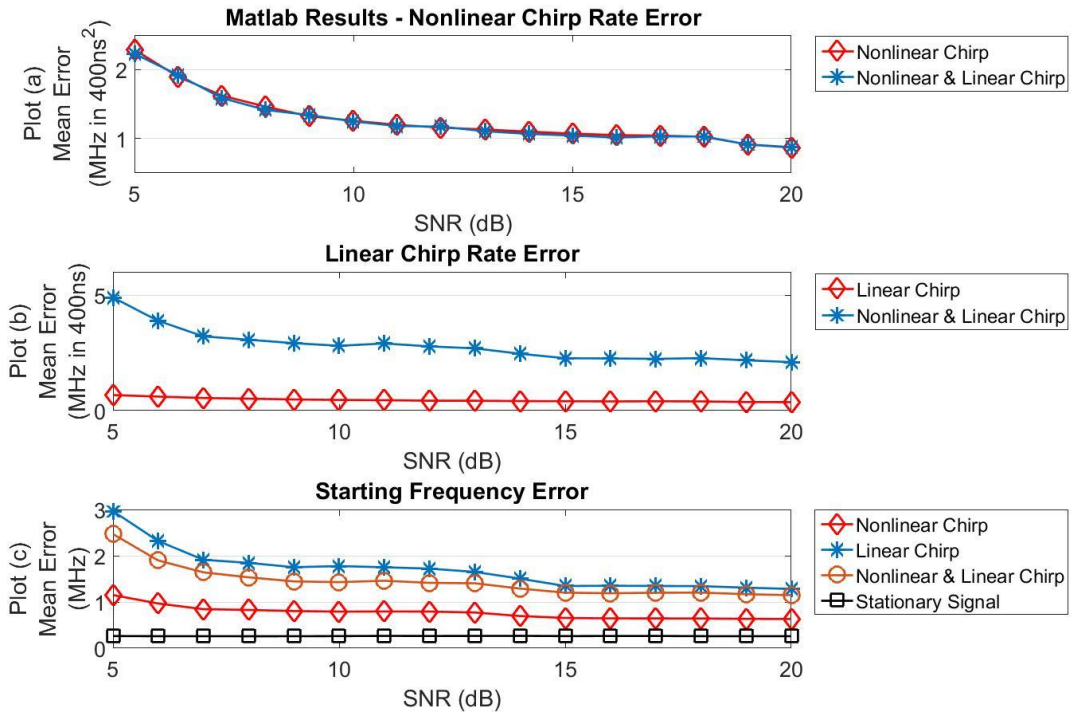

Figure 58 - Nonlinear chirp receiver Matlab based frequency measurement performance.

Table XV - Nonlinear chirp receiver Matlab based frequency measurement performance

Signal Type	Nonlinear Chirp Rate Avg. Error	Nonlinear Chirp Rate Std. Dev.	Linear Chirp Rate Avg. Error	Linear Chirp Rate Std. Dev.	Carrier Frequency Avg. Error	Carrier Frequency Std. Dev.
Nonlinear Chirp	1.27MHz in 400ns^2	0.99MHz in 400ns^2	N/A	N/A	0.77MHz	0.69MHz
Linear Chirp	N/A	N/A	0.46Mhz in 400ns	0.36Mhz in 400ns	1.70MHz	1.80MHz
Nonlinear & Linear Chirp	1.26MHz in 400ns^2	0.98MHz in 400ns^2	2.80Mhz in 400ns	2.75Mhz in 400ns	1.44MHz	1.42MHz
Stationary	N/A	N/A	N/A	N/A	0.26MHz	0.19MHz

9.7 Nonlinear Chirp Receiver FPGA

The nonlinear chirp receiver was synthesized using Xilinx ISE and targeted for a Virtex 6 SX475 FPGA. Fig. 59 shows a highlight of the synthesis report for the design. The digital nonlinear chirp receiver design only utilizes 13% of the total resources available and meets all timing constraints at a rate of 320MHz.

Project File:	Nonlinear_Chirp_Receiver_43.xise	**Parser Errors:**	No Errors
Module Name:	toplevel_adc	**Implementation State:**	Programming File Generated
Target Device:	xc6vsx475t-2ff1759	▪ **Errors:**	No Errors
Product Version:	ISE 13.4	▪ **Warnings:**	2512 Warnings (0 new)
Design Goal:	Balanced	▪ **Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	▪ **Timing Constraints:**	All Constraints Met
Environment:	System Settings	▪ **Final Timing Score:**	0 (Timing Report)

Device Utilization Summary				[-]
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	29,132	595,200	4%	
Number used as Flip Flops	29,106			
Number used as Latches	4			
Number used as Latch-thrus	0			
Number used as AND/OR logics	22			
Number of Slice LUTs	40,038	297,600	13%	
Number used as logic	33,600	297,600	11%	
Number using O6 output only	25,166			
Number using O5 output only	870			
Number using O5 and O6	7,564			
Number used as ROM	0			
Number used as Memory	4,627	122,240	3%	
Number used as Dual Port RAM	196			
Number using O6 output only	68			
Number using O5 output only	20			
Number using O5 and O6	108			
Number used as Single Port RAM	0			
Number used as Shift Register	4,431			
Number using O6 output only	4,431			
Number using O5 output only	0			
Number using O5 and O6	0			
Number used exclusively as route-thrus	1,811			
Number with same-slice register load	1,078			
Number with same-slice carry load	45			
Number with other load	688			
Number of occupied Slices	18,623	74,400	25%	

Figure 59 - Nonlinear chirp receiver Matlab based frequency measurement performance.

The FPGA results for the digital nonlinear chirp receiver test four different signal types; nonlinear chirp signals, linear chirp signals, simultaneous nonlinear and linear chirp signals and stationary carrier signals.

For nonlinear chirp signals, the receiver measured nonlinear chirp rates with an average error of 1.52MHz in $400ns^2$ and a standard deviation of 1.95MHz in $400ns^2$.

Starting frequencies were measured with an average error of 0.63MHz and a standard deviation of 0.81MHz.

For linear chirp signals, the receiver measured linear chirp rates with an average error of 0.80MHz in 400ns and a standard deviation of 0.98MHz in 400ns. Starting frequencies were measured with an average error of 1.74MHz and a standard deviation of 2.23MHz.

For simultaneous nonlinear and linear chirp signals, the receiver measured nonlinear chirp rates with an average error of 1.66MHz in $400ns^2$ and a standard deviation of 1.71MHz in $400ns^2$. Linear chirp rates were measured with an average error of 1.67MHz in 400ns and a standard deviation of 2.10MHz in 400ns. Starting frequencies were measured with an average error of 1.40MHz and a standard deviation of 1.77MHz.

For stationary carrier signals, the receiver measured carrier frequencies with an average error of 0.35MHz and a standard deviation of 0.43MHz.

Fig. 60 plot (a) shows the mean error in MHz per $400ns^2$ for nonlinear chirp rate measurement error. Plot (b) shows the mean error in MHz per 400ns for linear chirp rate measurement error. Plot (c) shows the mean error in MHz for starting frequency / carrier frequency measurement error. Table XVI summarizes the performance of the nonlinear chirp receiver for all signal types.

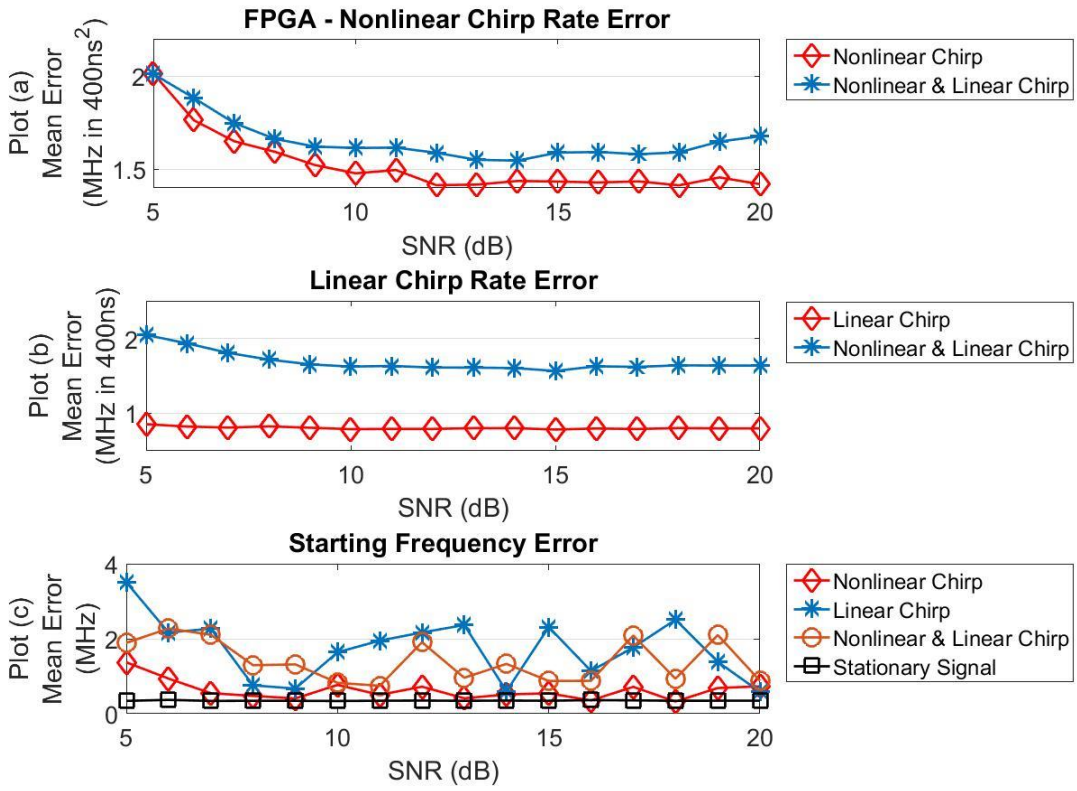

Figure 60 - Nonlinear chirp receiver FPGA based frequency measurement performance.

Table XVI - Nonlinear chirp receiver FPGA based frequency measurement performance

Signal Type	Nonlinear Chirp Rate Avg. Error	Nonlinear Chirp Rate Std. Dev.	Linear Chirp Rate Avg. Error	Linear Chirp Rate Std. Dev.	Carrier Frequency Avg. Error	Carrier Frequency Std. Dev.
Nonlinear Chirp	1.52MHz in 400ns^2	1.95MHz in 400ns^2	N/A	N/A	0.63MHz	0.81MHz
Linear Chirp	N/A	N/A	0.80Mhz in 400ns	0.98Mhz in 400ns	1.74MHz	2.23MHz
Nonlinear & Linear Chirp	1.66MHz in 400ns^2	1.71MHz in 400ns^2	1.67Mhz in 400ns	2.10Mhz in 400ns	1.40MHz	1.77MHz
Stationary	N/A	N/A	N/A	N/A	0.35MHz	0.43MHz

The thresholds for signal classification were also re-evaluated for the FPGA based nonlinear chirp receiver once all of the data had been collected. Fig. 61 shows the threshold ranges for the receiver for all seven signal power measurements. Fig. 62 shows

the detection rates achieved for all four signal types across the tested SNRs. Table XVII shows the average detection rates and also shows that no misclassification errors were encountered for all 1,380,000 signals tested on the FPGA. The overall detection rates are maintained above 97% for all four signal types.

The FPGA based nonlinear chirp receiver achieved nearly identical detection rates as predicted by the original Matlab simulations. The Matlab simulations predicted a correct detection rate of 98.99% and the FPGA based design achieved an overall correct detection rate of 99.06%.

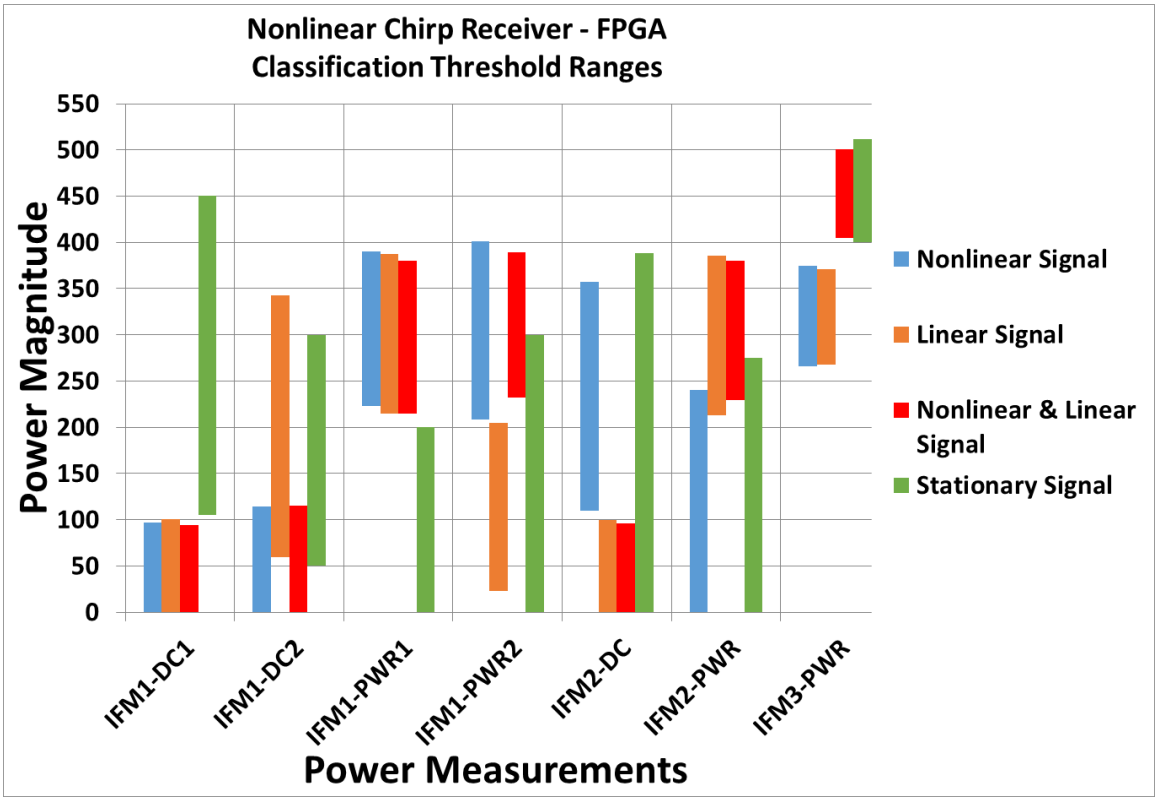

Figure 61 - Nonlinear chirp receiver FPGA based thresholds for signal classification.

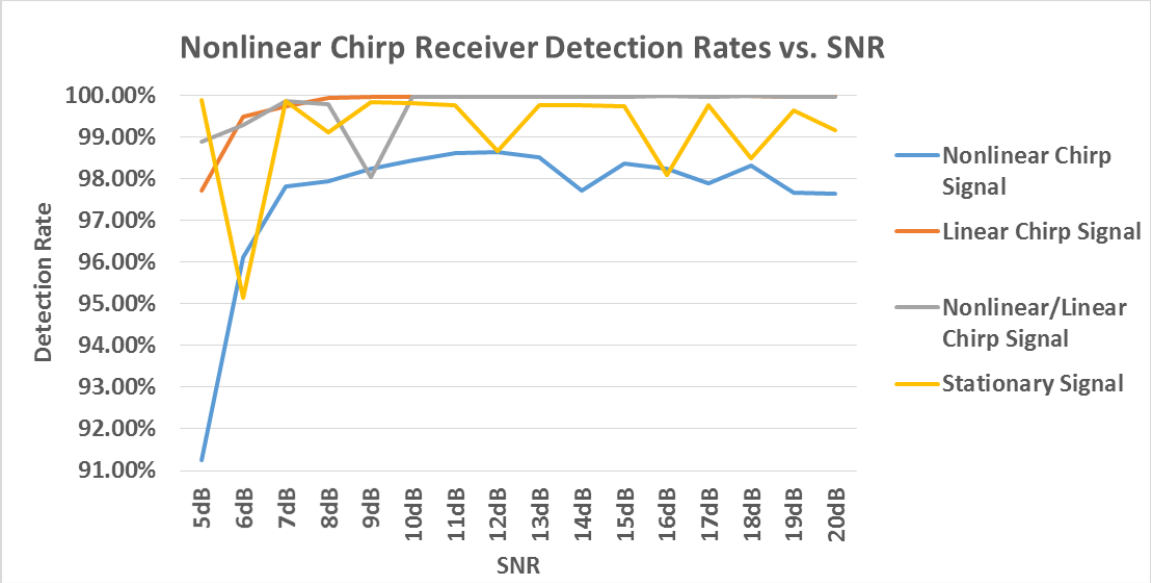

Figure 62 - Nonlinear chirp receiver FPGA based detection rates vs. SNR.

Table XVII - Nonlinear chirp receiver FPGA based detection and correct classification rates.

	Detection Rate	Misclassifications	Number of Simulations
Nonlinear Chirp Signal	97.59%	0	349,519
Linear Chirp Signal	99.79%	0	343,414
Nonlinear/Linear Chirp Signal	99.72%	0	347,768
Stationary Signal	99.16%	0	345,346
Overall	**99.06%**	**0**	**1,386,047**

9.8 Variable Chirp Receiver Matlab/Simulink

The variable chirp receiver performance was first tested on signals with a pulse width of 400ns. From the simulations, the digital variable chirp receiver obtained an average linear chirp measurement error of 3.65MHz in 400ns with a standard deviation of 4.95MHz in 400ns. The linear chirp receiver measured the starting frequency of linear chirp signals with an average measurement error of 1.82MHz and a standard deviation of 2.06MHz. For non-chirp stationary signals the average carrier frequency error was

simulated to be 0.13MHz with a standard deviation of 0.13MHz.

Fig. 63 plot (a) shows the mean error in MHz per 400ns for linear chirp rate measurement error. Plot (b) shows the mean carrier frequency error in MHz for both linear chirp signal and stationary signal types. Table XVIII summarizes the variable chirp receiver performance for signals with a pulse width of 400ns.

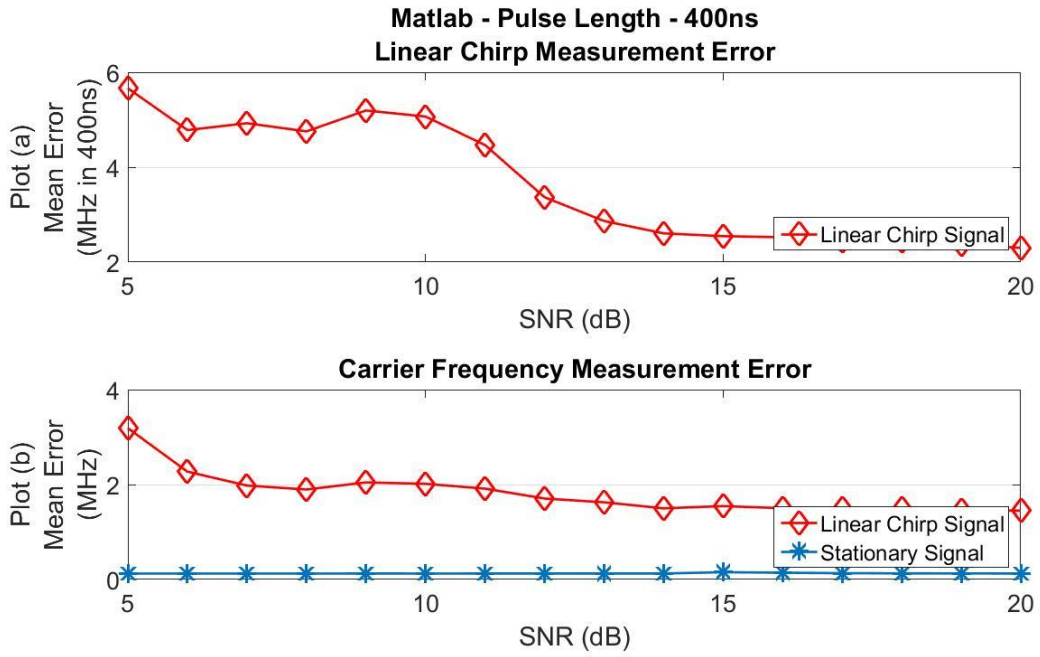

Figure 63 – Variable linear chirp receiver Matlab based frequency measurement performance for 400ns pulse widths.

Table XVIII - Variable chirp receiver Matlab based frequency measurement performance for 400ns pulse widths.

Signal Type	Linear Chirp Rate Avg. Error	Linear Chirp Rate Std. Dev.	Carrier Frequency Avg. Error	Carrier Frequency Std. Dev.
Linear Chirp	3.65Mhz in 400ns	4.95Mhz in 400ns	1.82MHz	2.06MHz
Stationary	N/A	N/A	0.13MHz	0.13MHz

A second set of simulations were run to test the variable chirp receiver's ability to measure linear chirp signals across a range of different pulse lengths. Pulse widths were varied from 400ns up to 4,000ns while maintaining the previous SNR and frequency ranges for each pulse width test. Fig. 64 plot (a) shows the linear chirp rate measurement performance across the range of pulse widths. Plot (b) shows the carrier frequency measurement performance for linear chirp signals and stationary signals across the range of pulse widths. The average linear chirp rate measurement error was simulated to be 1.06MHz with a standard deviation of 2.31MHz. The receiver obtained an average starting frequency measurement error of 0.54MHz with a standard deviation of 1.05MHz. For non-chirp stationary signals, the average carrier frequency measurement error was simulated to be 0.13MHz with a standard deviation of 0.13MHz. Table XIX summarizes the variable chirp receiver performance for linear chirp signals with various pulse widths.

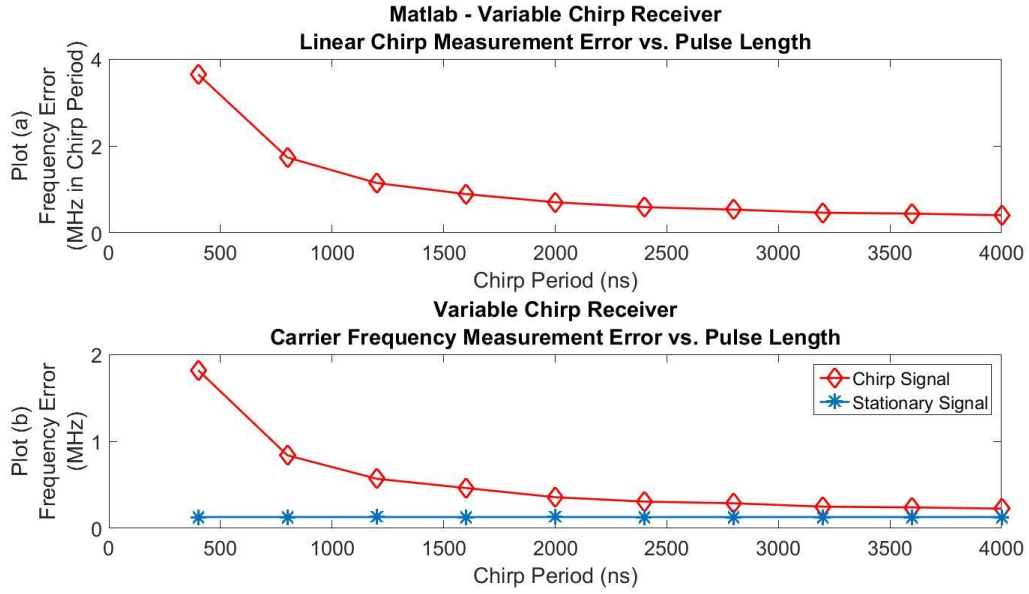

Figure 64 - Variable linear chirp receiver Matlab based frequency measurement performance for various pulse widths.

Table XIX - Variable chirp receiver Matlab based frequency measurement performance for various pulse widths.

Signal Type	Linear Chirp Rate Avg. Error	Linear Chirp Rate Std. Dev.	Carrier Frequency Avg. Error	Carrier Frequency Std. Dev.
Linear Chirp				
400ns Pulse	3.65MHz in 400ns	5.54MHz in 400ns	1.82MHz	2.29MHz
800ns Pulse	1.74MHz in 400ns	2.49MHz in 400ns	0.84MHz	1.25MHz
1200ns Pulse	1.15MHz in 400ns	1.63MHz in 400ns	0.57MHz	0.86MHz
1600ns Pulse	0.89MHz in 400ns	1.36MHz in 400ns	0.46MHz	0.67MHz
2000ns Pulse	0.70MHz in 400ns	0.98MHz in 400ns	0.36MHz	0.51MHz
2400ns Pulse	0.59MHz in 400ns	0.81MHz in 400ns	0.31MHz	0.44MHz
2800ns Pulse	0.54MHz in 400ns	0.71MHz in 400ns	0.29MHz	0.40MHz
3200ns Pulse	0.46MHz in 400ns	0.62MHz in 400ns	0.25MHz	0.37MHz
3600ns Pulse	0.44MHz in 400ns	0.58MHz in 400ns	0.24MHz	0.32MHz
4000ns Pulse	0.41MHz in 400ns	0.52MHz in 400ns	0.23MHz	0.31MHz
Stationary	N/A	N/A	0.13MHz	0.13MHz

9.9 Variable Chirp Receiver FPGA

The variable chirp receiver was synthesized using Xilinx ISE and targeted for a Virtex 6 SX475 FPGA. Fig. 65 shows a highlight of the synthesis report for the design. The digital variable chirp receiver design only utilizes 11% of the total resources available and meets all timing constraints at a rate of 320MHz.

Project File:	Variable_Chirp_Receiver.xise	**Parser Errors:**	No Errors
Module Name:	toplevel_adc	**Implementation State:**	Programming File Generated
Target Device:	xc6vsx475t-2ff1759	▪ **Errors:**	No Errors
Product Version:	ISE 13.4	▪ **Warnings:**	1895 Warnings (48 new)
Design Goal:	Timing Performance	▪ **Routing Results:**	All Signals Completely Routed
Design Strategy:	Performance with IOB Packing	▪ **Timing Constraints:**	All Constraints Met
Environment:	System Settings	▪ **Final Timing Score:**	0 (Timing Report)

Device Utilization Summary				[-]
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	27,251	595,200	4%	
Number used as Flip Flops	27,243			
Number used as Latches	4			
Number used as Latch-thrus	0			
Number used as AND/OR logics	4			
Number of Slice LUTs	34,771	297,600	11%	
Number used as logic	30,201	297,600	10%	
Number using O6 output only	23,806			
Number using O5 output only	1,002			
Number using O5 and O6	5,393			
Number used as ROM	0			
Number used as Memory	3,905	122,240	3%	
Number used as Dual Port RAM	196			
Number using O6 output only	68			
Number using O5 output only	20			
Number using O5 and O6	108			
Number used as Single Port RAM	0			
Number used as Shift Register	3,709			
Number using O6 output only	3,669			
Number using O5 output only	0			
Number using O5 and O6	40			
Number used exclusively as route-thrus	665			
Number with same-slice register load	614			
Number with same-slice carry load	47			
Number with other load	4			
Number of occupied Slices	16,925	74,400	22%	

Figure 65 - Nonlinear chirp receiver Matlab based frequency measurement performance.

The variable chirp receiver performance was first tested on signals with pulse widths of 400ns. From the FPGA simulations, the digital variable chirp receiver obtained an average linear chirp measurement error of 2.94MHz in 400ns with a standard deviation of 4.00MHz in 400ns. The variable chirp receiver measured the starting

frequency of linear chirp signals with an average measurement error of 1.69MHz and a standard deviation of 2.12MHz. For non-chirp stationary signals the average carrier frequency error was simulated to be 0.41MHz with a standard deviation of 0.41MHz.

Fig. 66 plot (a) shows the mean error in MHz per 400ns for linear chirp rate measurement error. Plot (b) shows the mean carrier frequency error in MHz for both linear chirp signal and stationary signal types. Table XX summarizes the variable chirp receiver performance for signals with a pulse width of 400ns.

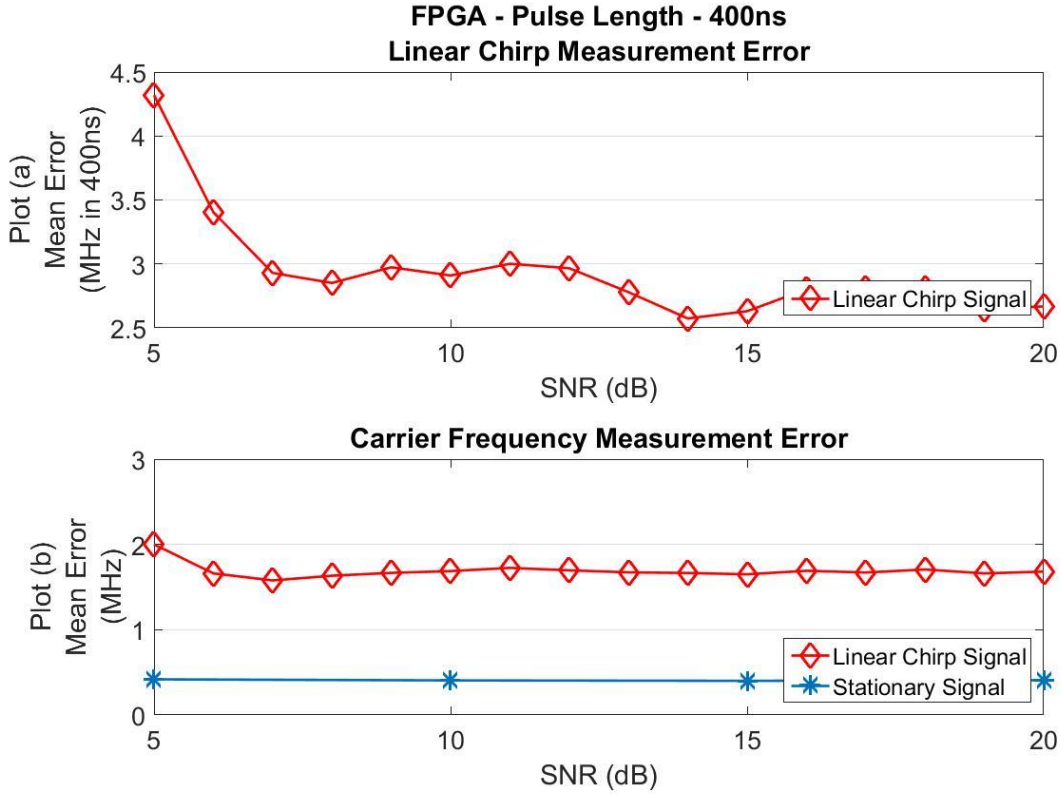

Figure 66 - Variable linear chirp receiver FPGA based frequency measurement performance for 400ns pulse widths.

Table XX - Variable chirp receiver Matlab based frequency measurement performance for 400ns pulse widths.

FPGA				
Signal Type	**Linear Chirp Rate Avg. Error**	**Linear Chirp Rate Std. Dev.**	**Carrier Frequency Avg. Error**	**Carrier Frequency Std. Dev.**
Linear Chirp	2.94Mhz in 400ns	4.00Mhz in 400ns	1.69MHz	2.12MHz
Stationary	N/A	N/A	0.41MHz	0.41MHz

A second set of simulations were run to test the variable chirp receiver's ability to measure linear chirp signals across a range of pulse lengths. Pulse widths were varied from 400ns up to 4,000ns while maintaining the previous SNR and frequencies ranges for each pulse width test. Fig. 67 plot (a) shows the linear chirp rate measurement performance across the range of pulse width. Plot (b) shows the carrier frequency measurement performance for linear chirp signals across the range of pulse widths. The average linear chirp rate measurement error was simulated to be 1.12MHz with a standard deviation of 1.60MHz. The receiver obtained an average starting frequency measurement error of 0.66MHz with a standard deviation of 0.92MHz. Table XXI summarizes the variable chirp receiver performance for linear chirp signals with various pulse widths.

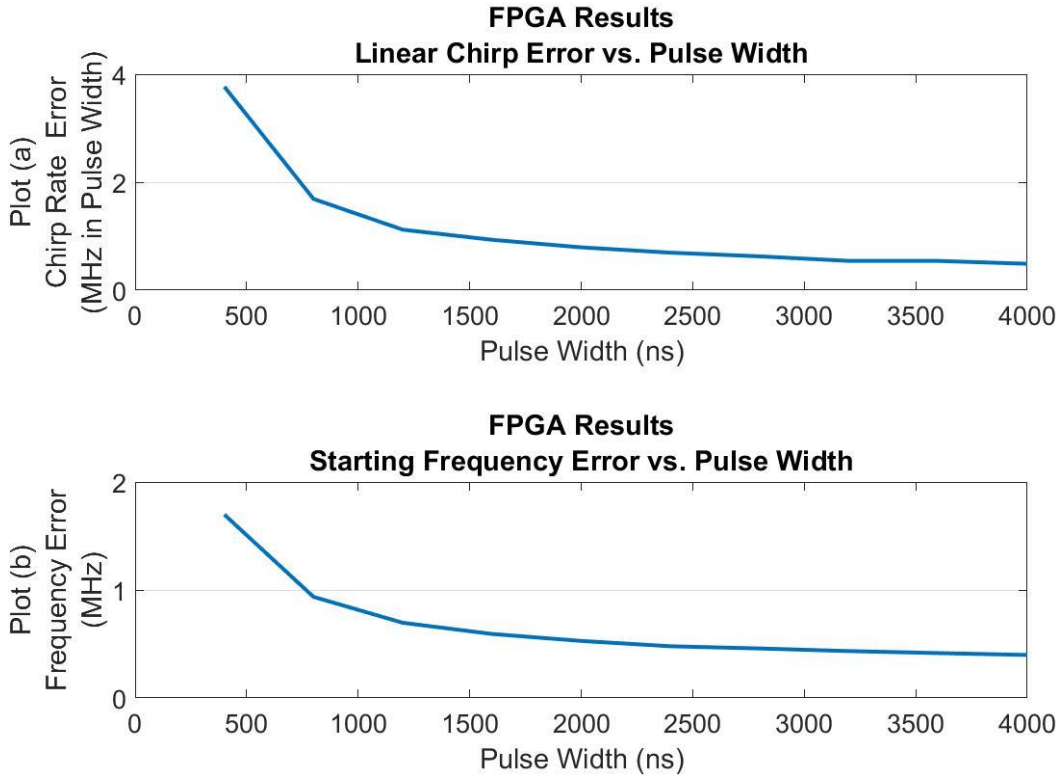

Figure 67 - Variable linear chirp receiver FPGA based frequency measurement performance for various pulse widths.

Table XXI - Variable chirp receiver FPGA based linear chirp signal frequency measurement performance for various pulse widths.

Signal Type	Linear Chirp Rate Avg. Error	Linear Chirp Rate Std. Dev.	Carrier Frequency Avg. Error	Carrier Frequency Std. Dev.
Linear Chirp				
400ns Pulse	3.77MHz in 400ns	5.07MHz in 400ns	1.70MHz	2.14MHz
800ns Pulse	1.69MHz in 400ns	2.40MHz in 400ns	0.94MHz	1.18MHz
1200ns Pulse	1.12MHz in 400ns	1.61MHz in 400ns	0.70MHz	0.92MHz
1600ns Pulse	0.93MHz in 400ns	1.32MHz in 400ns	0.59MHz	0.83MHz
2000ns Pulse	0.79MHz in 400ns	1.17MHz in 400ns	0.53MHz	0.81MHz
2400ns Pulse	0.69MHz in 400ns	1.05MHz in 400ns	0.48MHz	0.75MHz
2800ns Pulse	0.63MHz in 400ns	0.99MHz in 400ns	0.46MHz	0.75MHz
3200ns Pulse	0.54MHz in 400ns	0.89MHz in 400ns	0.43MHz	0.70MHz
3600ns Pulse	0.54MHz in 400ns	0.83MHz in 400ns	0.42MHz	0.65MHz
4000ns Pulse	0.49MHz in 400ns	0.64MHz in 400ns	0.40MHz	0.47MHz

The thresholds for signal classification were also re-evaluated for the FPGA based variable chirp receiver once all of the data had been collected. After adjusting the thresholds, detection rates and misclassification rates were re-evaluated for the FPGA based variable chirp receiver. Fig. 68 shows the threshold ranges for the receiver for all seven signal power measurements. Table XXII shows the average detection rates and also shows that no misclassification errors were encountered for all 960,000 signals tested on the FPGA. The overall detection rates are maintained above 98% for both signal types across all the pulse widths tested.

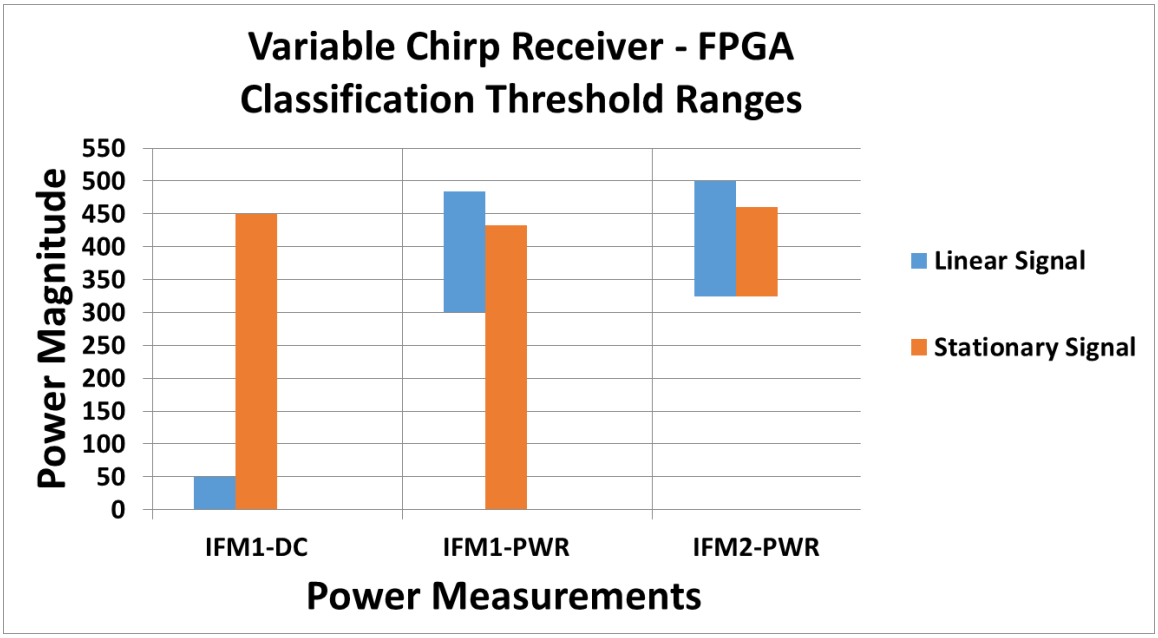

Figure 68 - Variable chirp receiver FPGA based thresholds for signal classification.

Table XXII - Variable chirp receiver FPGA based detection and correct classification rates.

Linear Chirp Signal	Detection Rate	Misclassifications	Number of Simulations
400ns Pulse Width	98.49%	0	87,298
800ns Pulse Width	98.65%	0	87,297
1200ns Pulse Width	98.16%	0	87,297
1600ns Pulse Width	98.18%	0	87,298
2000ns Pulse Width	98.25%	0	87,297
2400ns Pulse Width	98.06%	0	87,297
2800ns Pulse Width	98.08%	0	87,298
3200ns Pulse Width	98.06%	0	87,297
3600ns Pulse Width	98.05%	0	87,298
4000ns Pulse Width	98.08%	0	87,297
Stationary Signal	98.06%	0	87,362
Linear Chirp Overall	**98.21%**	**0**	**872,974**
Overall	**98.13%**	**0**	**960,336**

X. Conclusion

10.1 Research Contributions

Measuring high chirp rate, short pulse duration signals is a difficult problem. Ideally, a long pulse duration is desirable to achieve a high SNR and allow for an accurate and reliable frequency measurement. This allows for a large size FFT to be utilized. However, when the pulse duration is very short, the FFT is unable to measure frequencies with a high resolution unless a very high sampling rate is utilized. The difficulty of this problem is further compounded when no a priori knowledge of the signal is available. Typical approaches utilize a matched filter approach, which is a maximum likelihood solution. The research conducted for this dissertation successfully shows a proof of concept, which is able to detect and measure high chirp rate linear and nonlinear signals with no a priori knowledge.

The contributions from this research include the production of a high resolution TOA algorithm, capable of accurately detecting both time of arrival and time of departure of incoming chirp and stationary signals. Significant research and simulations were run to ensure a 90% detection rate with less than a 10^{-7} false alarm rate for SNRs ranging from 5dB up to 20dB. The algorithm was developed to run in real time, at clock rates up to 320MHz on a Virtex 6 FPGA. The algorithm can detect the TOA of a signal with an average error of 2.76 samples (1.07ns).

In addition to the TOA algorithm, significant research was conducted to investigate the use of various Hilbert Transform designs and their effect on chirp signal measurement. For this research, a 43 tap Type III Hilbert Transform was implemented, able to operate at a clock rate of 320MHz on a Virtex 6 FPGA.

Significant research regarding the use of digital IFMs for high chirp rate signal measurement has also been conducted and presented in this dissertation. Both the strengths and weaknesses of the digital IFM have been extensively quantified. It was found that digital IFMs are more than adequate for high chirp rate signal measurement.

The linear chirp receiver is able to measure linear chirp rates up to 1180MHz in 400ns with an average linear chirp rate error of 0.35MHz in 400ns and carrier frequencies up to 1230MHz with an average error of 0.33MHz.

The nonlinear chirp receiver is able to measure nonlinear chirp rates up to 1180MHz in $400ns^2$ with an average chirp rate error of 1.52MHz in $400ns^2$, linear chirp rates up to 1180MHz in 400ns with an average chirp rate error of 0.80MHz in 400ns and carrier frequencies up to 1230MHz with an average error of 0.35MHz.

The variable chirp receiver is able to measure linear chirp signals with a wide range of pulse widths and achieve a chirp rate error of less than 3.77MHz in 400ns.

10.2 Future Research

The biggest limitation of the three digital chirp receivers is their inability to measure more than one simultaneous signal. This is due to the inherent functionality of the digital IFM, which takes a single discrete phase measurement. It should be possible to measure more than one simultaneous signal by utilizing multiple channelized receivers,

for example one receiver operates from 50MHz up to 1,230MHz and a second receiver operates from 1,330MHz up to 2,510MHz.

Though a variable chirp receiver was designed to measure linear chirp signals with varying pulse widths, the same approach cannot be used for nonlinear chirp signals, due to the nonlinear rate of change. A new methodology will need to be investigated for use with the nonlinear chirp receiver.

The receivers could also be modified to detect and measure additional modulation types, such as frequency hopping spread spectrum, phase-shift keying, or frequency-shift keying.

XI. References

[1] Toomay, J. C. Radar Principles for the Non-specialist. 3rd ed. Raleigh: SciTech, 2004. Print.

[2] P. Kanagaratnam, T. Markus, V. I. Lytle, B. Heavey, P. Jansen, G. Prescott, and S. P. Gogineni, "Ultra-wideband radar measurements of thickness of snow over sea ice," Trans. Geosci. Remote Sens., vol. 45, no. 9, pp. 2715–2724, Sep. 2007.

[3] M. Kronauge, and H. Rohling, "New Chirp Sequency Radar Waveform", *IEEE Trans. on Aerospace and Electronic Systems*, vol. 50, no. 4, pp. 2870-2877, Oct. 2014.

[4] F. Want, C. Yang, X. Yang and Z. Wang, 'Short-Distance Detecting and Imaging System with UWB Chirp Signal', in IEEE International Conference on Microwave Technology & Computational Electromagnetics, Qingdao, 2015, pp. 368-371.

[5] P. Leong, T. Abhayapala and T. Lamahewa, 'Multiple Target Localization Using Wideband Echo Chirp Signals', IEEE Transactions on Signal Processing, vol. 61, no. 16, pp. 4077-4089, 2013.

[6] W. Q. Wang, "Large Time-Bandwidth Product MIMO Radar Waveform Design Based on Chirp Rate Diversity," IEEE Sensors Journal, vol.15 no. 2, pp. 1027-1034, February 2015.

[7] Wang, Quan, "Non-Linear Chirp Spread Spectrum Communication Systems of Binary Orthogonal Keying Mode" (2015). *Electronic Thesis and Dissertation Repository*. Paper 2898. http://ir.lib.uwo.ca/etd/2898

[8] M. A. Khan, R. K. Rao, and X. Wang, "Performance of quadratic and exponential multiuser chirp spread spectrum communication systems," in *8th IEEE International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, July 2013.

[9] G. N. Saddik, R. S. Singh, and R. Brown, "Ultra-wideband multifunctional communications/radar system," IEEE Trans. Microw. Theory Tech., vol. 55, no. 7, pp. 1431–1437, Jul. 2007.

[10] J. Tsui, Special Design Topics in Digital Wideband Receivers, Boston; London: Artech House, 2010.

[11] G. F. Boudreaux-Bartels and T. W. Parks, "Time-Varying Filtering and Signal Estimation Techniques Using Wigner Distribution Synthesis Techniques," IEEE Trans. Acoustic, Speech, Signal Processing, vol. ASSP-34, June 1986.

[12] W. Li, "Wigner distribution method equivalent to dechirp method for detecting a chirp signal," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-35, Aug. 1987.

[13] J. C. Wood and D. T. Barry, "Radon transformation of time-frequency distributions for analysis of multicomponent signals," IEEE Trans. Signal Processing, vol. 42, pp. 3166–3177, Nov. 1994.

[14] M. Wang, A. K. Chan, and C. K. Chui, "Linear Frequency-Modulated Signal Detection Using Radon-Ambiguity Transform," IEEE Trans. on Signal Processing, vol. 46, no. 3, pp. 571-585, Mar. 1998.

[15] O. Akay, and G. F. Boudreaux-Bartels, "Fractional Convolution and Correlation via Operator Methods and an Application to Detection of Linear FM Signals," IEEE Trans. on Signal Processing, vol. 49, no. 5, pp. 979-993, May 2001.

[16] Y. Sun; P. Willett "Hough Transform for Long Chirp Detection," IEEE Transactions on Aerospace and Electronic Systems, vol.38, no.2, pp.553-569, Apr 2002.

[17] I. Rusnak and L. Rafael, 'New approach to estimation of chirp signal with unknown parameters', in IEEE International Conference on Microwaves, Communications, Antennas and Electronics Systems, Tel Aviv, 2015.

[18] E. O. Brigham. The Fast Fourier Transform and Its Applications. Englewood Cliffs, Prentice-Hall, NJ, 1988.

[19] J. G. Proakis and D. G. Manolakis. Digital Signal Processing: Principles, *Algorithms, and Applications*. Prentice-Hall, third edition, 1996.

[20] Weisstein, Eric W. "Hilbert Transform." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/HilbertTransform.html

[21] "Hilbert Transform." *Wikipedia*. Wikimedia Foundation, n.d. Web. 13 Oct. 2015.

[22] D. S. Murthy, "Real-Time Hilbert Transform and autocorrelation for digital wideband communication applications," M.S. thesis, EE, WSU, Dayton, OH, 2008.

[23] P.W. East, "Fifty years of instantaneous frequency measurement," IET Radar Sonar Navig., vol. 6, no. 2, pp. 112–122, 2012.

[24] C. Pandolfi, E. Fitini, G. Gabrielli, E. Megna and A. Zaccaron, "Comparison of Analog IFM and Digital Frequency Measurement Receivers for Electronic Warfare," Proc. 7th European Radar Conference 2010, pp. 232-235, October 2010.

[25] D. Lam, B. Buckley, C. Lonappan, A. Madni and B. Jalali, 'Ultra-wideband instantaneous frequency estimation', IEEE Instrum. Meas. Mag., vol. 18, no. 2, pp. 26-30, 2015.

[26] J. Helton, C. -I. H. Chen, D. M. Lin, J.B.Y. Tsui, "FPGA-Based 1.2GHz bandwidth digital instantaneous frequency measurement receiver," in *Proceedings of 9th International Symposium on Quality Electronic Design*, pp. 568-571, San Jose, CA, 2008.

[27] D. M. Lin, L. L. Liou, S. Benson and C.-I. H. Chen, "Mono-bit digital chirp receiver using mono-bit IFM (Instantaneous Frequency Measurement) Receiver as a core," in *Proceedings of Aerospace and Electronics Conference*, pp. 348-351, July 2011.

[28] G. L. Risueno, J. Grajal, and A. Sanz-Osorio, "Digital Channelized Receiver Based on Time-Frequency Analysis for Signal Interception," *IEEE Trans. on Aerospace and Electronic Systems, Vol.21, No. 3*, pp. 879-898,2005