Chaotic Based Self-Synchronization for RF Steganography Radar/Communication Waveform

A Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Electrical Engineering

by

Michael A. Gonnella
B.S.E.E., Wright State University, 2016

2018
Wright State University

Wright State University
GRADUATE SCHOOL

December 3, 2018

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Michael A. Gonnella ENTITLED Chaotic Based Self-Synchronization for RF Steganography Radar/Communication Waveform BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Electrical Engineering.

Zhiqiang Wu, Ph.D.
Thesis Director

Fred D. Garber, Ph.D.
Interim Chair, Department of Electrical Engineering

Committee on
Final Examination

Zhiqiang Wu, Ph.D.

Saiyu Ren, Ph.D.

Yan Zhuang, Ph.D.

Barry Milligan, Ph.D.
Interim Dean of the Graduate School

ABSTRACT

Gonnella, Michael A. M.S.E.E., Department of Electrical Engineering, Wright State University, 2018. *Chaotic Based Self-Synchronization for RF Steganography Radar/Communication Waveform.*

In this project, we continue previous CSR project entitled RF Steganography based Joint Radar/Communication Waveform Design to develop a bio-inspired secure low probability detection (LPD) radio frequency (RF) waveform that can serve multiple purposes simultaneously. Previously, we have developed an RF steganography based RF waveform to conceal a secure digital communication within a linear frequency modulated (LFM) chirp radar signal. By exploiting novel reduced phase shift keying modulation and variable symbol duration, the new waveform is resistant to time domain analysis, frequency domain analysis and cyclostationary analysis. However, to demodulate the hidden communication message, the intended receiver has to know the entire sequence of variable symbol duration, or the entire sequence of pseudo-random phases. We are developing a chaotic based self-synchronization scheme to solve this problem and provide enhanced security. Specifically, a chaotic sequence generator is employed to generate an aperiodic chaotic sequence to control the phase of the reduced phase shift keying modulation. The intended receiver only needs to have knowledge of the initial condition of the chaotic sequence generator to generate the entire pseudo-random phase sequence to achieve self-synchronization.

Contents

List of Figures

Acknowledgment

I would like to take this opportunity to first and foremost thank Dr. Zhiqiang Wu, my advisor, mentor, and friend. Dr. Wu's support, guidance, and patience during the research and writing of this thesis were invaluable and this would not have been possible without him.

Thank you to the Center for Surveillance Research for providing the numerous presentation opportunities, constructive feedback from industry and Defense partners, and funding for the project.

I would also like to thank my committee members, Dr. Saiyu Ren and Dr. Yan Zhuang for taking the time and effort to assist me.

And finally to my wife, Kasey, who has endured a lifetimes worth of sarcasm in these last few months. Thank you for putting up with me.

There is no obstacle too great,

no challenge too difficult, if we have faith.

Dedicated to my parents

For always having faith in me

Introduction

1.1 Motivation

Electronic warfare is an ongoing battle in the electromagnetic (EM) spectrum. One party will develop a new secure method of communications and another will immediately begin work to defeat the implemented security measures. Once the security is defeated this allows the other party access to the communications leading to continued development of new and more secure methods of communication.

As early as 1900 BCE, evidence has been found showing the use of cryptography, the creation and study of codes and techniques used to communicate securely, when non-standard hieroglyphs were found on the walls. In one instance a formula for a pottery glaze was hidden, protecting trade secrets. Since then the methods and techniques have evolved through many different stages to keep up with the forms of communication being used and the type of data being hidden. Some more recent examples are the various Axis powers cipher machines during World War II, such as the German Enigma and the Japanese JN-25 along with the associated allied decryption efforts. [1]

Figure 1.1: The German Enigma machine

Military use of encryption became commonplace as each side attempted to securely transmit information as it was proven that gaining access to an adversaries movements and strategies could turn the tide of war. Contemporary systems face these challenges more than ever as more and more data is being transmitted wirelessly it is imperative that secure transmissions be made.

In this project we expand upon a previous project, titled *RF Steganography based Joint Radar/Communication Waveform Design*. In the previous project a secure, low probability of detection (LPD) radio frequency (RF) waveform was designed which could be used both as a radar waveform, as well as a communications waveform.

One common approach to LPD waveforms uses spread-spectrum methods to distribute a signals power across a larger frequency band such that the power spectral density of the signal lies below the noise floor making it difficult to differentiate the signal from the noise. Another method emulates naturally occuring waveforms, such as random noise, or animal sounds, to hide a signal within an environment.

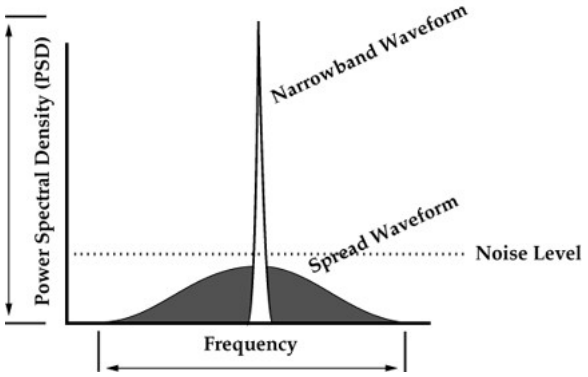


Figure 1.2: Spread-Spectrum Technique

In the previous project a different approach was chose - RF Steganography: the use of an existing waveform to conceal a secondary communication within it. Previous research has developed a novel means of embedding a communications signal onto an existing radar waveform that is not able to be detected using more advanced forms of signal analysis, namely cyclostationary analysis. Our goal is to improve upon this design by implementing a chaotic based self-synchronization scheme. The following background chapter will give an overview of cyclostationary analysis and outline the development of the designed waveform thus far.

Background

2.1 Cyclostationary Analysis

The advent of cyclostationary analysis rendered many LPD waveforms obsolete and driven the design of this waveform, so it is helpful to discuss what it is. Cyclostationary analysis identifies man-made waveforms by their periodic features such as amplitude, phase, and frequency modulation. [2] By correlating a signal with itself at various offsets, we are able to find these periodic features. Because noise has no cyclostationary features, the autocorrelation of the noise at any time offset will be low, but the periodic features will be exposed. This allows us to discover signal which are hidden below the noise-floor or noise-like. [3]

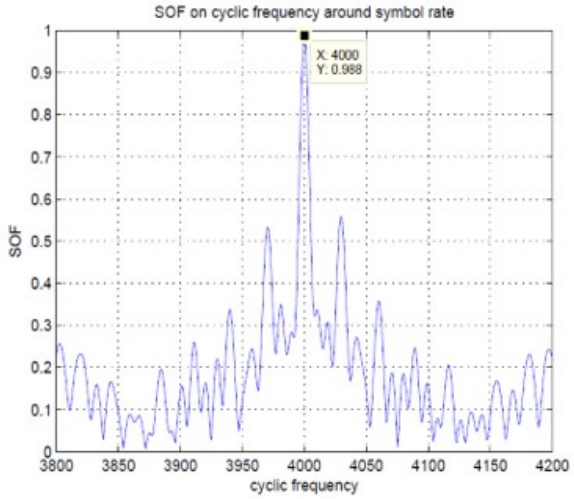


Figure 2.1: Spectral Correlation Function

2.1.1 Cyclic Autocorrelation Function

A fundamental parameter when distinguishing cyclic features from random data is the *limit cyclic autocorrelation*, which is a general form of the conventional limit autocorrelation and limit spectum.[2][4] We begin by defining the mean of an assumed complex signal $x(t)$ as follows: [5][6]

$$M_x(t, \tau) = E[x(t + \tau] \tag{2.1}$$

In the equation above, τ is the lag value.

For a cyclostationary signal, the mean is independent of τ. Next we look at the *auto-correlation function* (AF). Sometimes called the temporal lag product series, this is the correlation of a signal with itself with a temporal lag, defined by the following equation:

$$R_x(t_1, t_2) = E[x(t_1)x^*(t_2)] \tag{2.2}$$

Rewritten with $t_1 = t + \tau/2$, $t_2 = t - \tau/2$, $t = (t_1 + t_2)/2$, and $\tau = t_1 - t_2$:

$$R_x(t, \tau) = E[x(t + \frac{\tau}{2})x^*(t - \frac{\tau}{2})] \tag{2.3}$$

We can also use a Fourier series to represent the AF.

$$R_x(t,\tau) = \sum_{\alpha} R_x^{\alpha}(\tau) e^{j2\pi\alpha t} \tag{2.4}$$

In the above equation $R_x^{\alpha}(\tau)$ is the *cyclic autocorrelation function* or CAF and α is the *cyclic frequency* (CF). The CAF can be defined as:

$$R_x^{\alpha}(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} R_x(t,\tau) e^{-j2\pi\alpha t} dt \tag{2.5}$$

Where T is the period and $\alpha = m/T$.

2.1.2 Spectral Correlation Function

Next we look at the *spectral correlation function* (SCF). The SCF is the most used for identifying cyclostationary features of random signals. Similar to how the power spectrum is the spectral density of variance, the SCF is the spectral density of covariance and is defined as follows: [5][6]

$$S_x^{\alpha}(f) = \int_{-\infty}^{\infty} R_x^{\alpha}(\tau) e^{-j2\pi f\tau} d\tau \tag{2.6}$$

Where α is the cyclic frequency, and f is the spectrum frequency.

2.2 Waveform Design

2.2.1 Bio-Inspired Chirp Carrier

The advent of cyclostationary analysis rendered previous LPD waveforms obsolete. While previous LPD designs utilized techniques to suppress the power spectral density below the ambient noise floor, cyclostationary analysis is able to detect man-made periodicities within a signal, such as modulation, despite being below the noise floor as the noise has no cyclostationary features [7][8]. Instead the designed waveform hides in plain sight using what was termed *RF Steganography* - embedding the communications waveform on an existing radar waveform, which we've defined to be a linear frequency modulated (LFM) chirp signal (seen below). [8]

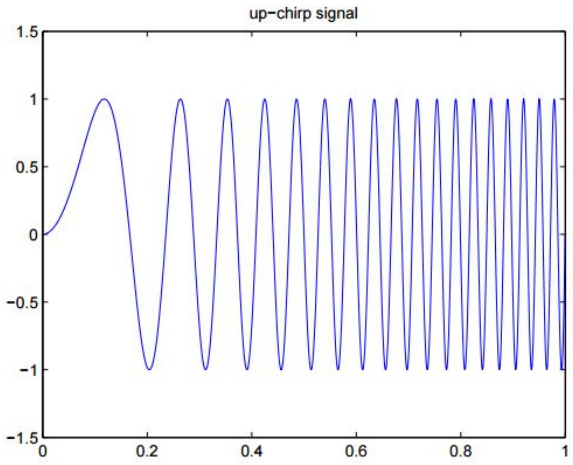
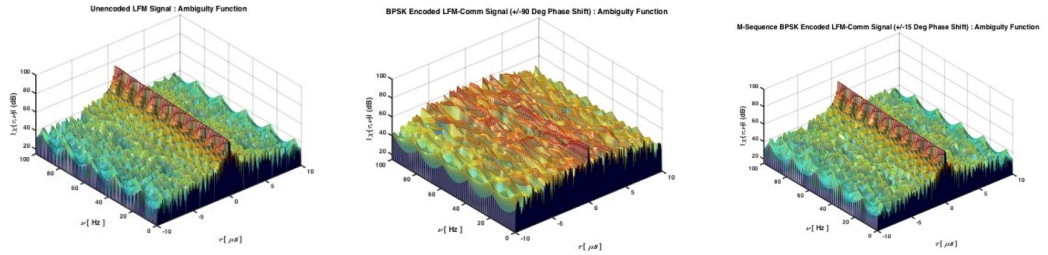


Figure 2.2: LFM Signal

2.2.2 Data Modulation

In order to embed communications on the existing LFM chirp a binary phase shift keying (BPSK) modulation scheme was explored and subsequently rejected after observing the negative effect on the radar performance using the ambiguity function (Figure 2.3(b)). [8]



(a) Unmodulated LFM Signal (b) BPSK Modulated Signal (c) RBPSK Modulated Signal

Figure 2.3: Ambiguity Functions Comparison

To solve this issue a reduced binary phase shift keying (RBPSK) modulation scheme was implemented, which uses a smaller, $\pm 15°$, and as we can see from the ambiguity function plots below Figure 2.3(c) much more closely matches Figure 2.3(a). Normally this would not be a feasible solution as the reduced phase increases the likelihood of error due to noise, however due to the fact that the embedded signal will only need to reach the target receiver and not return back to the radar it will benefit from a high signal-to-noise ratio (SNR) making this less of an issue. [8]

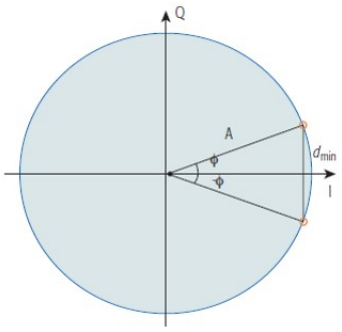


Figure 2.4: RPSK Constellation

2.2.3 Variable Phase/Symbol Duration

However, by modulating data onto the radar waveform we have introduced new cyclostationary features which need to be eliminated, and this is accomplished by using variable symbol duration reduced binary phase shift keying (VSDRBPSK). By varying the symbol durations though we have now caused a variation in energy (E_b) from symbol to symbol, wherein symbols with a longer duration have a higher energy and symbols with a shorter duration have a lower energy. Fortunately this is simply solved by varying the symbol phase using the relationship shown in Equation 2.7 and demonstrated in Figure 2.5. [8]

$$T_{b_1} sin^2 \phi_1 = T_{b_2} sin^2 \phi_2 \tag{2.7}$$

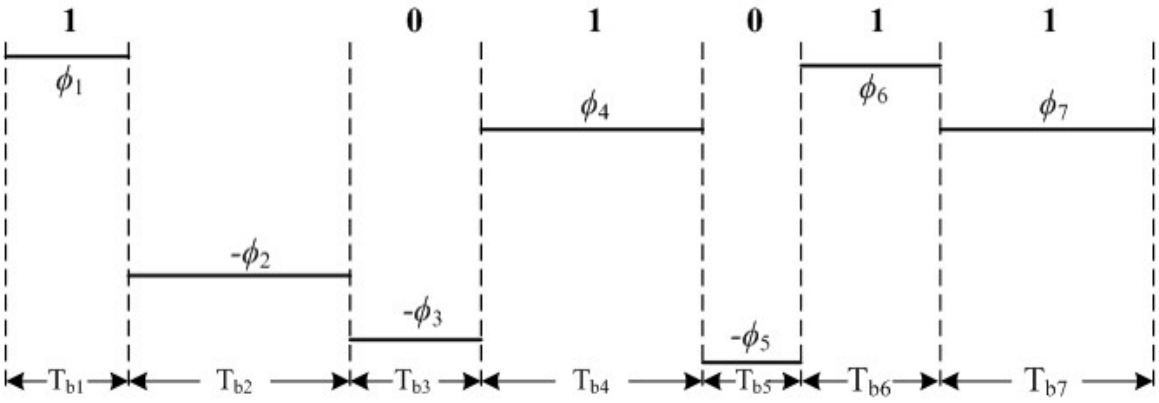


Figure 2.5: VSDRBPSK with Phase Offsets

2.2.4 Design Summary

With that we have an LPD communication waveform embedded within an existing LFM radar chirp. Below we can see the block diagram for how the transmitter would operate and the equation for the output signal $s(t)$.

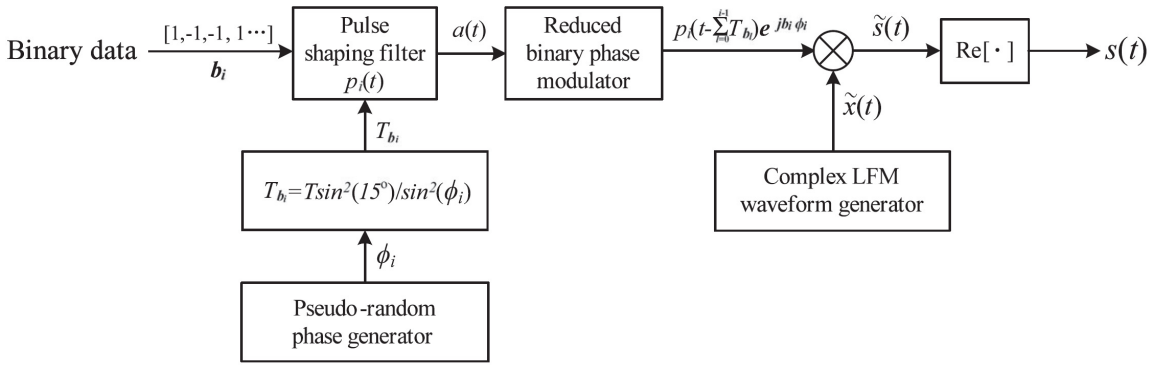


Figure 2.6: Transmitter Block Diagram

$$s(t) = \sum_{i=0}^{N-1} p_i(t) \cdot A_c cos(2\pi f_0 t + 2\pi \frac{k}{2} t^2 + b_i \cdot \phi_i + \phi_I) \tag{2.8}$$

Chaos and Chaos-based Communication Systems

From the outlined design we can see that in order for this system to function the transmitter has a psuedo-random phase generator which is used to modulate the data onto the LFM carrier. If the transmitter generates a pseudo-random sequence of numbers this would allow for the modulation of the signal, but then the receiver would have to also know the entire sequence in order to demodulate this data. To prevent this we propose the use of a chaotic sequence generator at both the transmitter and receiver. Here we will describe chaos and chaos-based communication systems.[9]

3.1 Chaos Theory

Chaos theory focuses on the study of nonlinear dynamical systems. A nonlinear system is one in which there is either a multiplying effect or feedback into the system. Dynamical means that the current state of the system affects the future state of the system. The father of chaos theory, Edward Lorenz, describes chaos as *"when the present determines the future, but the approximate present does not approximately determine the future."* [10]

3.1.1 Chaotic systems

Chaotic systems exhibit a number of desirable traits which are useful when considering security. First is the sensitivity to initial conditions, or what is also known as the butterfly effect. As we can see in Figure 3.1 below we have a chaotic system with the same growth rate and a very small difference in initial condition. Over the first half of the plot the values are very close to each other, however as time progresses, the slight deviation causes the systems to diverge. From a communications security standpoint this is desirable as even a slight error in the initial conditions would not allow an adversary access to your data. [10]

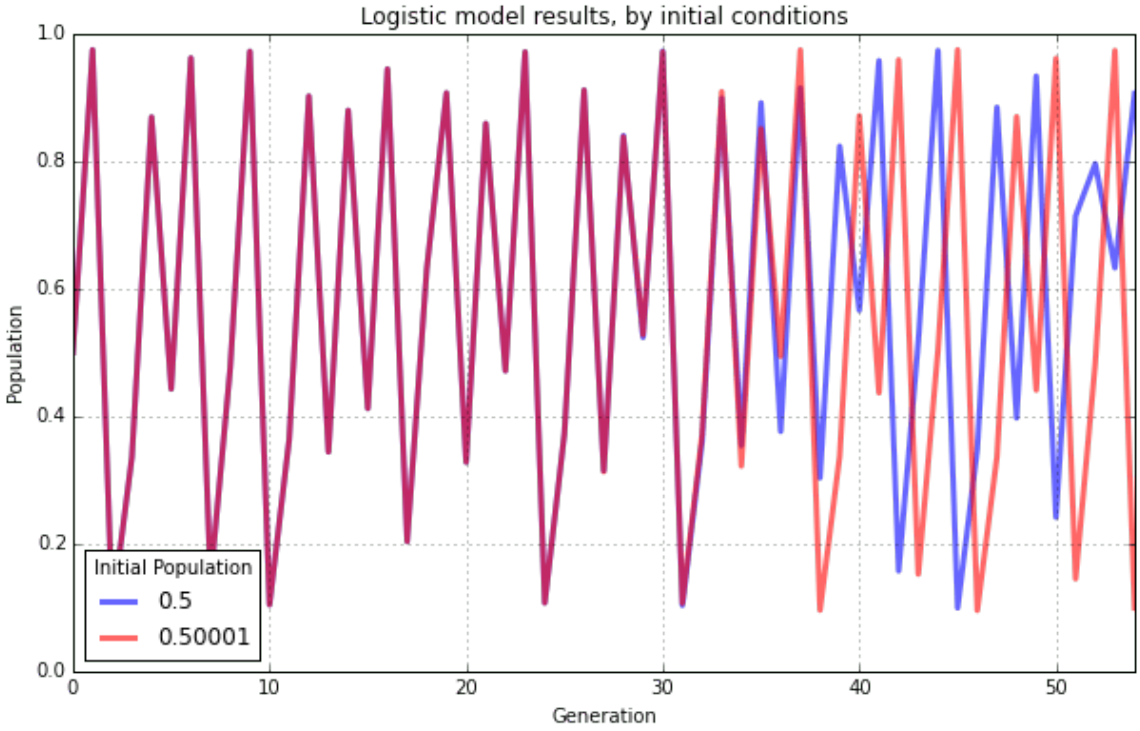


Figure 3.1: Butterfly Effect - Sensitivity to initial conditions

Another desirable trait that chaotic systems exhibit is that they are deterministic. Because of this, once an appropriate chaotic system is found, if the rate and initial value are known, the sequences are reproducible. For our purposes in creating a phase sequence for modulation this is highly desirable. Rather than having the receiver know the entire sequence of phases that were generated at the transmitter, the receiver only has to know the system, rate and initial value and it can generate it's own sequence locally for demodulation. Further because of the sensitivity to initial conditions this also makes it easy for the transmitter and receiver to change the scheme. [9]

Should an adversary somehow figure out how to demodulate the data, one could simply alter the initial conditions slightly which would create an entirely new phase sequence, which is much easier than transmitting an entirely new phase sequence to the receiver each time a change is made. Further, when comparing the chaotic generators to pseudo-noise sequences such as M-sequences and Gold sequences, the chaotic sequence is aperiodic. While there are only a limited number of Gold sequences and M-sequences, there are virtually an unlimited number of chaos sequences due to the varying equations, rates, and initial conditions. [11][10]

N	M-Sequence	Gold	Chaos
7	2	9	$\gg$9
15	2	17	$\gg$17
31	6	33	$\gg$33
63	6	65	$\gg$65
127	18	129	$\gg$129
255	16	257	$\gg$257
511	48	513	$\gg$513

Table 3.1: Number of sequences of length N

3.1.2 Chaos-based Communication Systems

While our proposed method seems to be the first attempt to use a chaotic sequence to phase map, there are other communications which have used chaotic systems to great effect. One such method utilizing chaotic systems performs modulation by chaos shift keying. As seen in Figure 3.2 below there are two chaotic generators. The bit modulated determines which chaotic generator output is used. At the output, the receiver replicates the chaotic signals to determine the threshold and demodulate the data. [12]

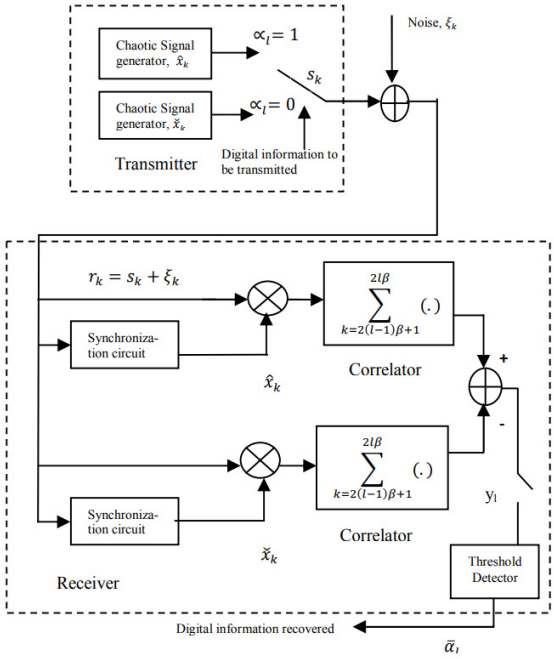


Figure 3.2: Chaos Shift Keying

Another method which uses chaotic sequences uses the output of a chaotic generator to frequency map an output. This utilizes the spread spectrum/frequency hopping idea but uses the chaotic generator to determine the hopping of the signal. Our proposed method will use the output of the chaotic generator similarly but will use it to map a phase offset versus a frequency. [13]

3.2 Autocorrelation

Next we look at the autocorrelation function which is used for alignment in time. In performing the synchronization we will be using the autocorrelation function to find the delay in the received signal. The delay is found by maximizing the output of the autocorrelation function at various lags. The definition of the autocorrelation function is as follows [14]:

$$R(s,t) = \frac{E[(X_t - \mu_t)(X_s - \mu_s)]}{\sigma_t \sigma_s} \tag{3.1}$$

Where X is a random process, t and s are the times, μ is the mean, and σ^2 is the variance, the above equation defines the autocorrelation between the times s and t.
When used without normalization in signal processing, the autocorrelation of a function $f(t)$ with itself at a given lag τ is as follows [14]:

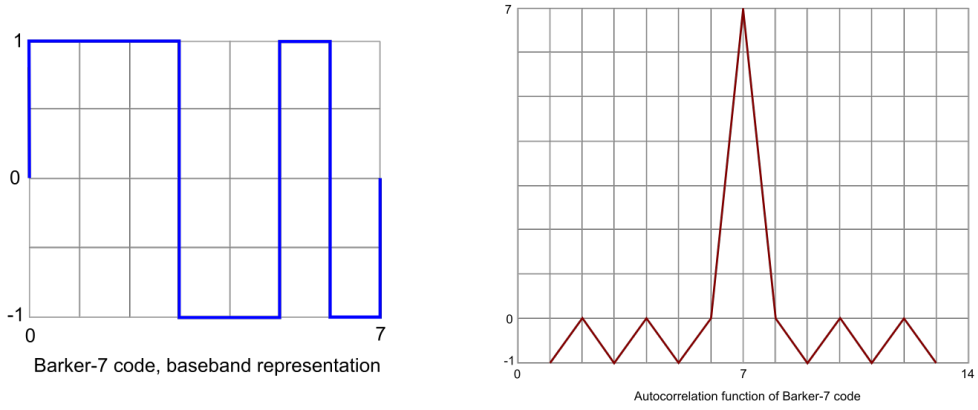
$$R_{ff}(\tau) = (f * g_{-1}(\overline{f}))(\tau) = \int_{-\infty}^{\infty} f(u)\overline{f}(u-\tau)du \tag{3.2}$$

Where $\overline{f}$ is the complex conjugate, g_{-1} is a function which manipulates the function f as defines $g_{-1}(f)(u) = f(-u)$ and $*$ represents convolution.

The discrete autocorrelation is then [14]:

$$R_{yy}(\ell) = \sum_{n \in Z} y(n)\overline{y}(n-\ell) \tag{3.3}$$

In order to aid this function pilot sequences are often sent out for synchronization purposes. Specifically we use pilot sequences which have a low autocorrelation when there is any delay in order to aid the autocorrelation function in properly achieving synchronization. Barker sequences are a special set of sequences which have ideal autocorrelation properties, as seen below. [15][16]



(a) Barker-7 Code

(b) Barker-7 Autocorrelation

Figure 3.3: Barker-7 Sequence

With this information we can implement a Barker sequence assisted autocorrelation function to detect the delay of an incoming signal.

3.2.1 Autocorrelation of Chaotic Sequence

Another desirable trait that chaotic sequences have is there is a low autocorrelation when there is any lag. This helps with synchronization and security as slight offsets would yield an incorrect demodulation of the transmitted data.

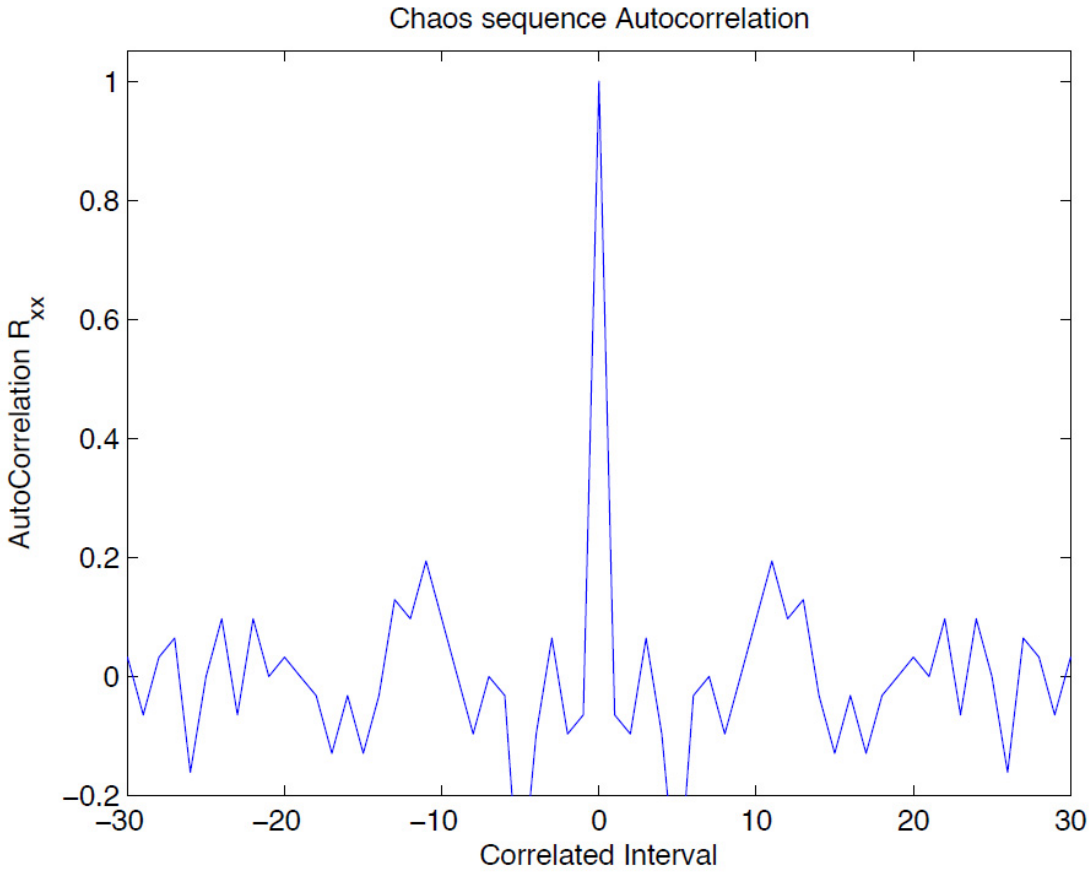


Figure 3.4: Chaotic Sequences - Autocorrelation

Next we will move on to the implementation of the proposed changes.

Implementation

As described in the previous chapter, chaotic sequences have been used in other forms of digital communication, but to our knowledge, chaotic sequences have not been used to generate a phase offset to perform phase modulation. This novel approach will eliminate the need for a transmitter and receiver to have knowledge of an entire pseudo-random sequence and instead simply know the chaotic sequence generator and initial seed in order to be able to communicate.

4.1 Transmitter

4.1.1 Chaotic Sequence Generation

The first step to implementing the proposed changes was to add the chaotic generator to the transmitter side. If we refer back to Figure 2.6 we can see where the pseudo-random phase generator is. In order to implement our proposed self-synchronization scheme this block must be changed for a chaotic sequence generator.

For our design we implemented the following chaotic sequence, called the logistic map, to generate our phases:

$$x_{n+1} = r x_n (1 - x_n) \tag{4.1}$$

Using this equation for the logistic map we are able to generate chaotic sequences by varying both the r value such that:

$$3.57 < r < 4$$

$$0 < x_1 < 1$$

This sequence was chosen for it simplicity and the large number of possible seeds which could be used. This sequence was also ideally bounded between 0 and 1. Other chaotic sequence generators could be used as well though in this place, the mapping that follows however may vary slightly.

Once each of these was set we were able to generate a chaotic sequence, shown below with $r = 3.95$ and $x_1 = 0.8$.

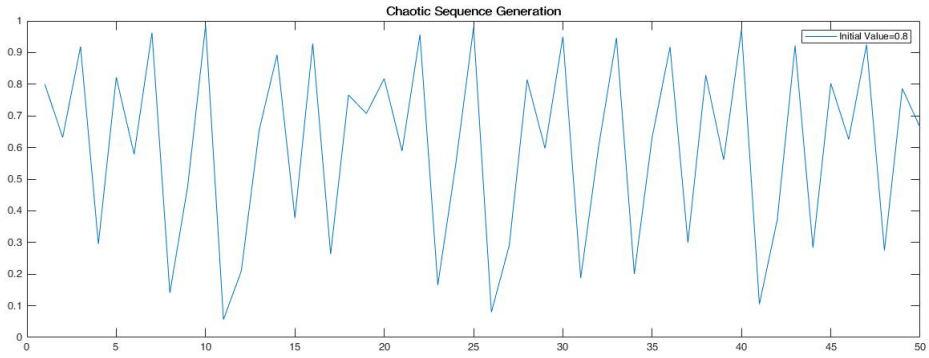


Figure 4.1: Chaotic Sequence Generated

4.1.2 Phase Mapping

Once the sequence was generated it was converted into a phase map using the following equation, where ϕ is the phase offset, ϕ_{min} is the minimum phase offset (5 in our example), ϕ_{max} is the maximum phase offset (15 in our example), and x_i:

$$\phi_i = \phi_{min} + (\phi_{max} - \phi_{min})x_i \tag{4.2}$$

This phase mapping equation works given that the chaotic sequence generated is between 0 and 1. If this is not the case then the chaotic sequence would need to be normalized to meet these criteria then can be used in the above equation.

From that equation our generated phase map is as follows:

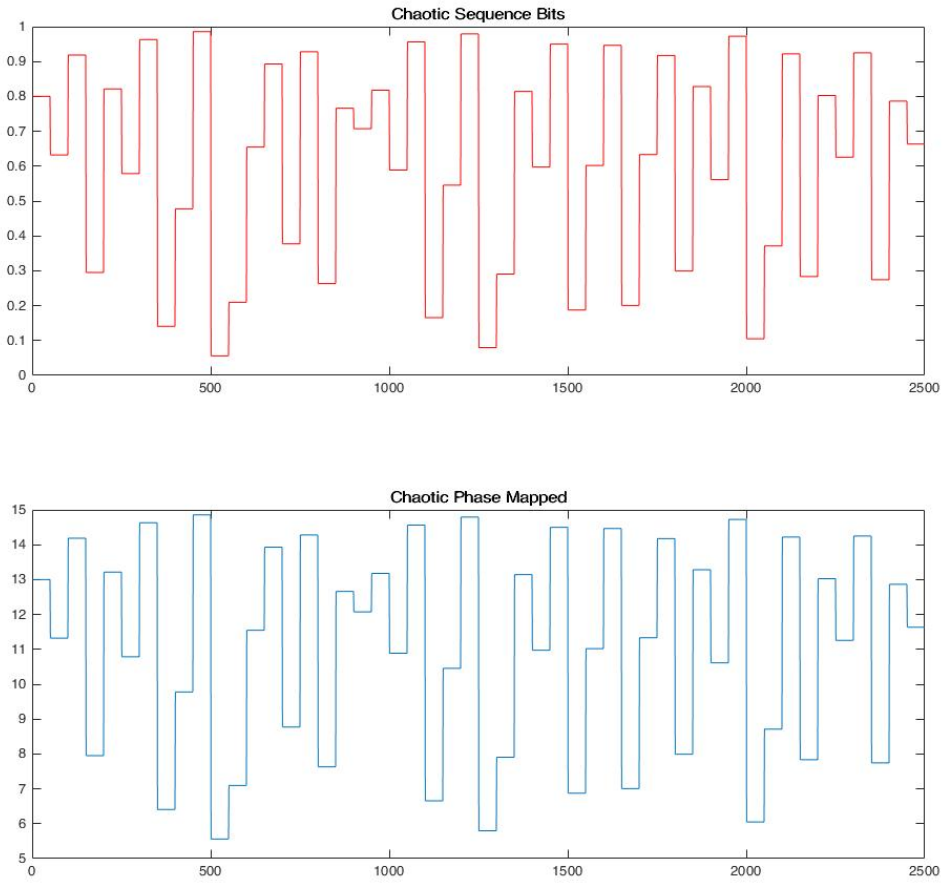


Figure 4.2: Chaotic Sequence to Phases

We can see that the generated chaos sequence has now been scaled to match the desired phase offsets we will be using in our VSDRBPSK modulation of the data.

4.1.3 Variable Duration

The next step is to compensate for the change in symbol energy due to the varying phases by also varying the symbol duration. By rearranging Equation 2.7 we get the following relationship to generate a duration factor:[15]

$$T_{factor_i} = \frac{sin^2\phi_{max}}{sin^2\phi_i} \tag{4.3}$$

Using this time scaling factor on the phase mapped sequence produces the following:

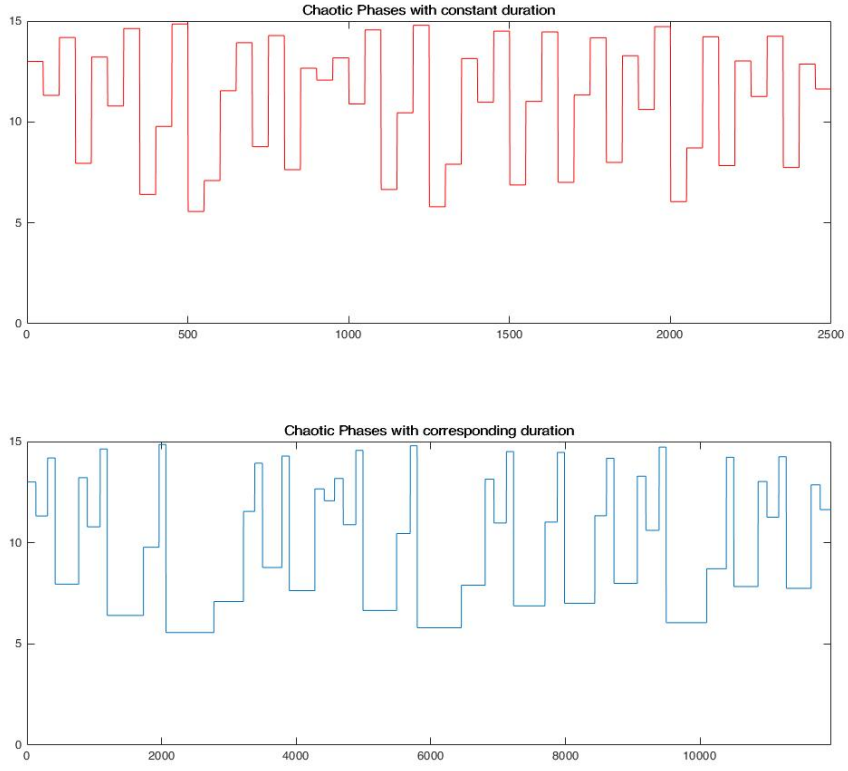


Figure 4.3: Chaotic Sequence Phase and Time Adjusted

We can now see how the higher phase offsets have a shorter duration and the lower phase offsets have a longer duration.

4.1.4 Pilot Sequence

Now that the phase offsets have been created with the appropriate duration factors that portion is ready to multiply with our data, however we must first add in our pilot bits. Below is a table of the barker sequences available:[16]

N	Barker Code
2	+ -, + +
3	+ + -
4	+ - + + , + - - -
5	+ + + - +
7	+ + + - - + -
11	+ + + - - - + - - + -
13	+ + + + + - - + + - + - +

For the example cases we will use a Barker Code of $N = 13$.

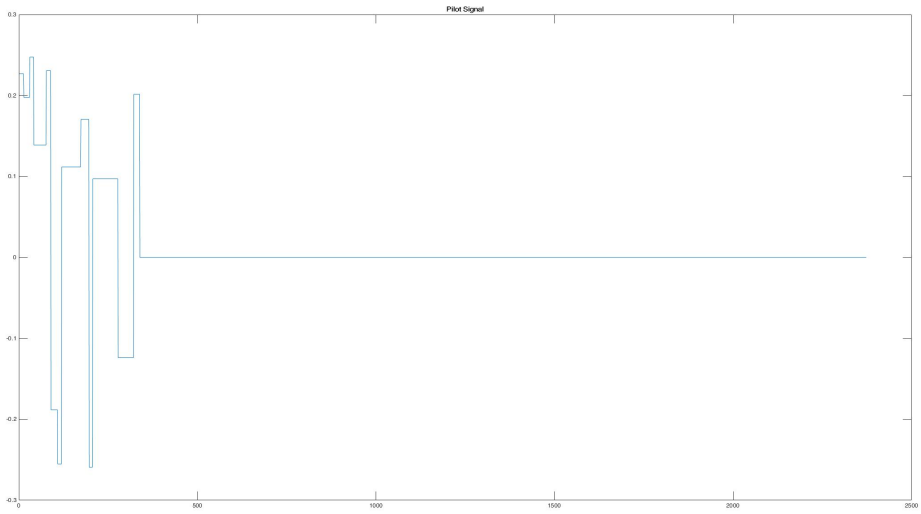


Figure 4.4: Barker Signal Pilot

Once this has been done we can append the desired data bits or generate any additional binary data, multiply the binary string with the phase offset and time adjusted chaos sequence, and then modulate the LFM chirp.

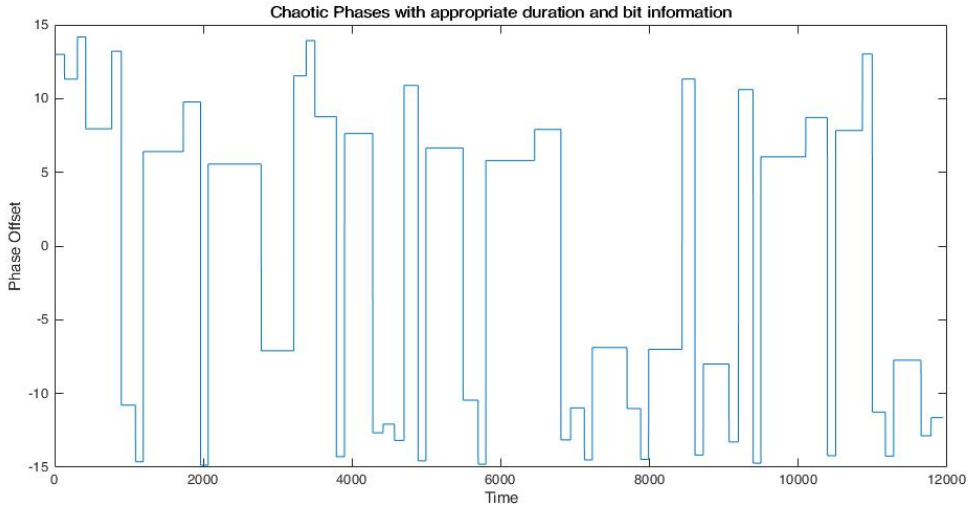


Figure 4.5: Chaotic Sequence Phase and Time Adjusted with Bit Data

The final output from the transmitter is as seen below: In the above diagram the fre-

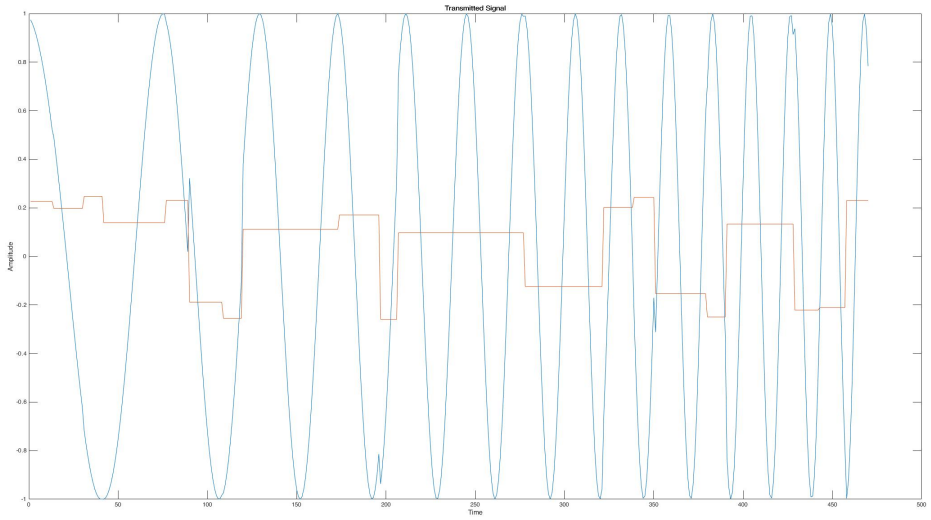


Figure 4.6: Transmitted Signal

quency was intentionally set low, and the phase shifts set relatively high, so that the modulation would be easily recognizable. This was done for visualization purposes only; in reality this could be implemented with a much higher frequency and much smaller offsets depending on the system parameters and the environment.

4.2 Receiver

4.2.1 Chaotic Sequence Generation

The next step is to implement the changes proposed in the receiver. First we will implement the chaotic sequence generator at the receiver, which is simply done by using the same equation, r value, and initial seed.

4.2.2 Compensating for Delay

The next step is accounting for any delay in the received signal. Once again, because we are enjoying a relatively high SNR this can be done by simply correlating the unmodulated LFM chirp with the received signal. Once the SNR is low enough that it has difficulty locating the delay, then due to the minimal phase shift in the VSDRBPSK modulation, the data will be unusable, but still, in order to aid in the detection we utilize the barker signal to modulate LFM waveform and then use that signal to find the delay using autocorrelation.

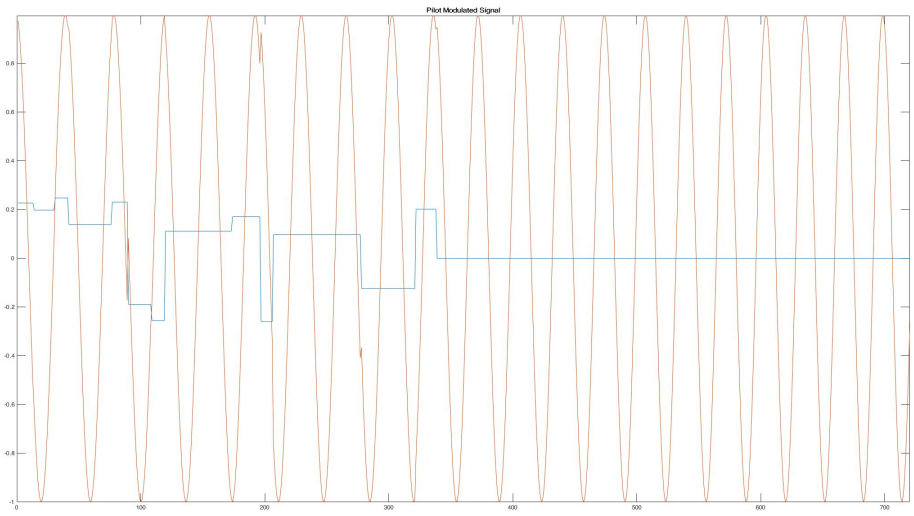


Figure 4.7: Pilot Modulated Signal

As we can see in the figure below the pilot sequence makes little difference.

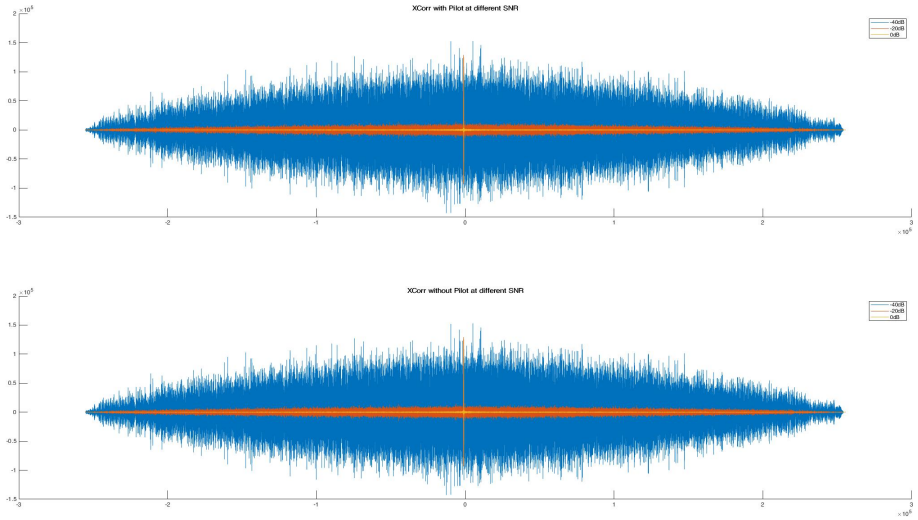


Figure 4.8: Cross Correlation with and without Barker Sequence

Once the delay has been compensated for we then move to demodulating the signal.

4.2.3 Demodulation

Next we begin working on the demodulation of the signal. One interesting property of the RPSK signal is that when we project the data onto each axis, only the quadrature component contains the data being transmitted. Knowing this we can demodulate the data using only the quadrature component.[17]

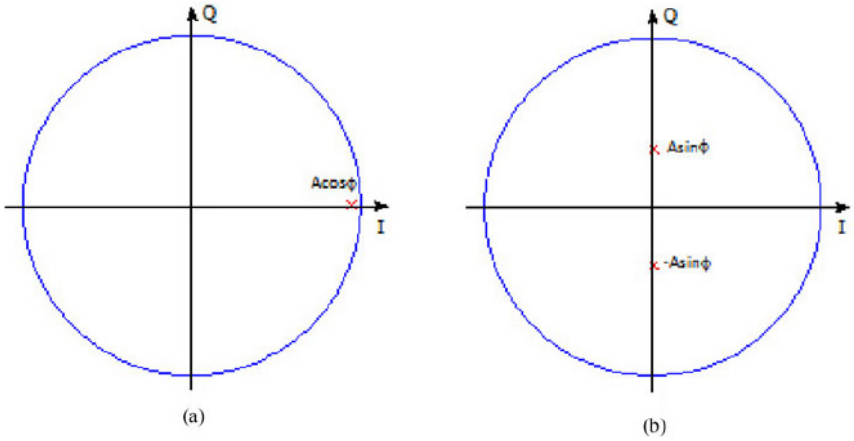


Figure 4.9: Projection of data onto component axes

With this knowledge we can create a matched filter receiver for the signal. A block diagram can be seen here: [17]

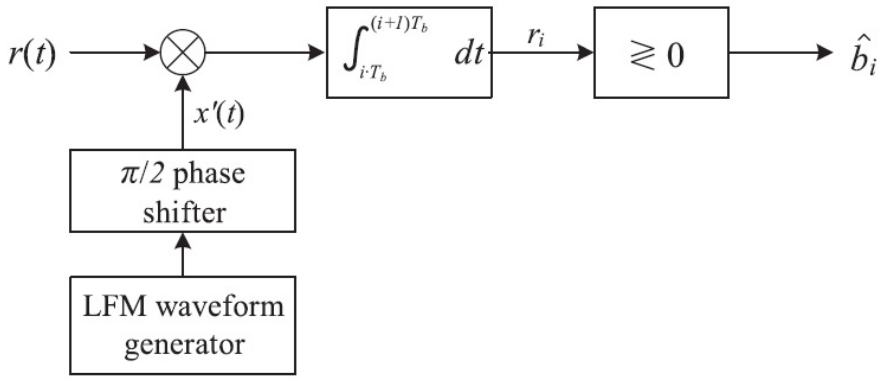


Figure 4.10: Block Diagram for Demodulation

We can describe the matched filter receiver mathematically as follows:

$$r_i = \int_{iT_b}^{(i+1)T_b} r(t)x'(t)dt \tag{4.4}$$

In this equation $x'(t)$ is a $\pi/2$ shifted signal which can be described by:

$$x'(t) = A_c cos(2\pi f_0 t + 2\pi \frac{k}{2} t^2 + \frac{\pi}{2} + \phi_I) \tag{4.5}$$

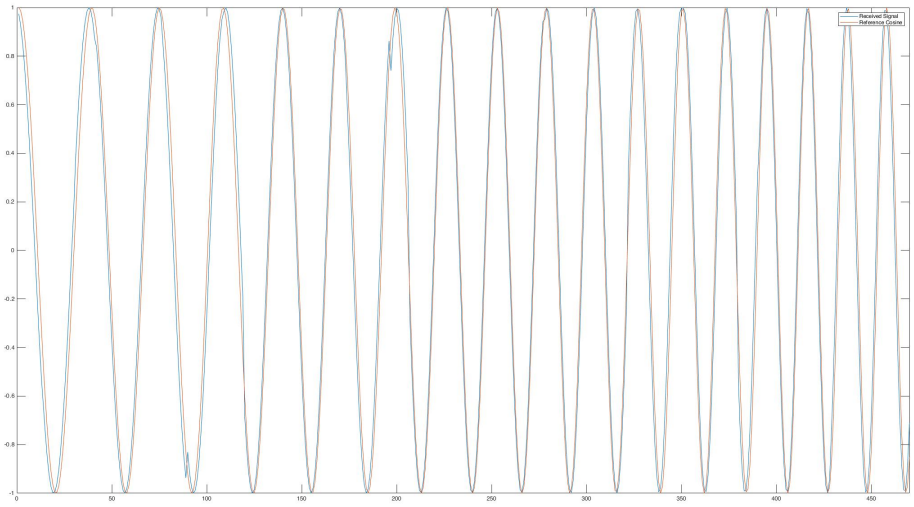


Figure 4.11: Signal with Reference

Then a hard decision is made:

$$\hat{b_i} = \begin{cases} 1, & \text{if } r_i > 0 \\ -1, & \text{otherwise} \end{cases} \tag{4.6}$$

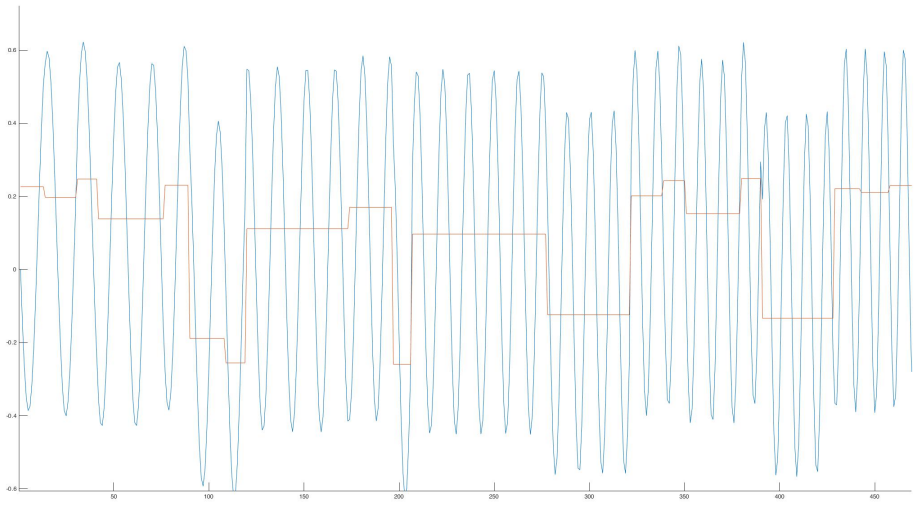


Figure 4.12: Signal Multiplied by Reference Sine Chirp and Inverted

4.2.4 BER Calculation

To calculate the BER for the VSDRBPSK signal we can use a BPSK BER equation as a basis. The theoretical BER for BPSK with additive white gaussian noise (AWGN) is as follows:

$$Q\Big(\frac{d_{min}/2}{\sigma}\Big) = Q\Big(\frac{A}{\sigma}\Big) = Q\Big(\sqrt{\frac{2E_b}{N_0}}\Big) \tag{4.7}$$

Where $A = sqrt(E_b)$, $\sigma^2 = N_0/2$ is the PSD of the AWGN, E_b is the bit energy, and $Q(x)$ is the Q function:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp\Big(-\frac{u^2}{2}\Big) du \tag{4.8}$$

However when we change the $d_{min}/2$ to account for the reduced phase shift we get the following:

$$Q\Big(\frac{d_{min}/2}{\sigma}\Big) = Q\Big(\frac{Asin(\phi)}{\sigma}\Big) = Q\Big(\sqrt{\frac{2E_b sin^2(\phi)}{N_0}}\Big) \tag{4.9}$$

By looking at this equation we can see that when $\phi = 90^o$, where this is the same as BPSK, the sin^2 component goes away and it is equal to the BPSK BER. As the phase is reduced this lowers the value of the term inside the radical, dropping the value passed to the Q function, which increases the BER. [17]

By passing in the various reduced phases and plotting the BER curves we can see a full BER plot for various phase angles and confirm that as the phase angle is reduced the BER increases, mathching our theoretical.

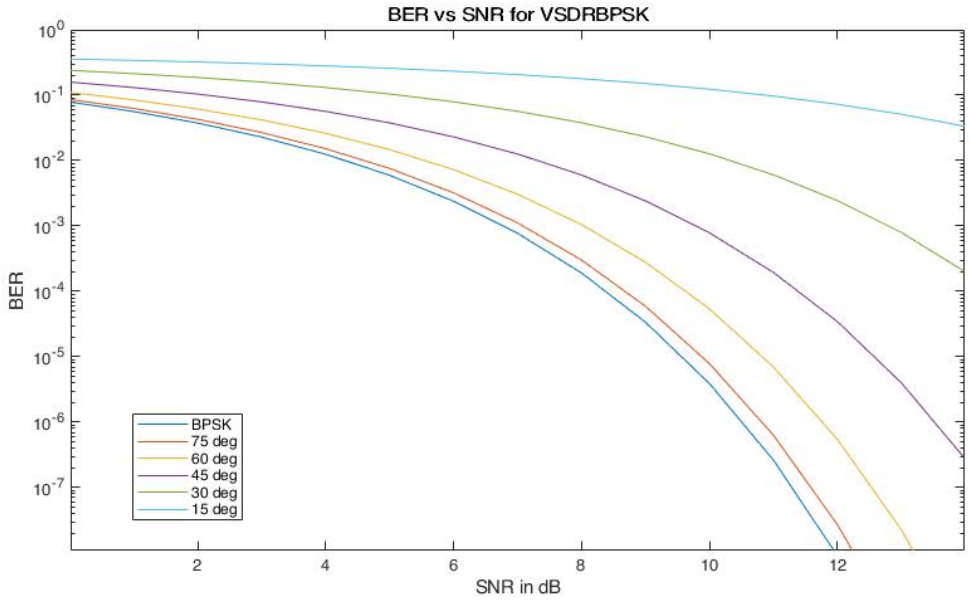


Figure 4.13: BER for BPSK vs Varied Reduced Phases

4.3 Software Defined Radio Implementation

One the coding was completed and tested via Matlab was to implement the change on the Software Defined Radio (SDR) that was used with the previous projects.

4.3.1 SDR Environment

The test setup utilizes two Ettus Research USRP X300 High Performance SDRs. The USRP X300 utilizes a Xilinx Kintex-7 FPGA for digital signal processing. For our purposes we utilized the high speed Gigabit ethernet interfaces. The hardware architecture provides compatible with the GNU Radio and C++/Python APIs which we use in our testing.

Figure 4.14: USRP X300

Each USRP X300 in our setup has two wideband RF daugherboard slots which allow for up to 160 MHz bandwidth each and can go from DC to 6GHz.

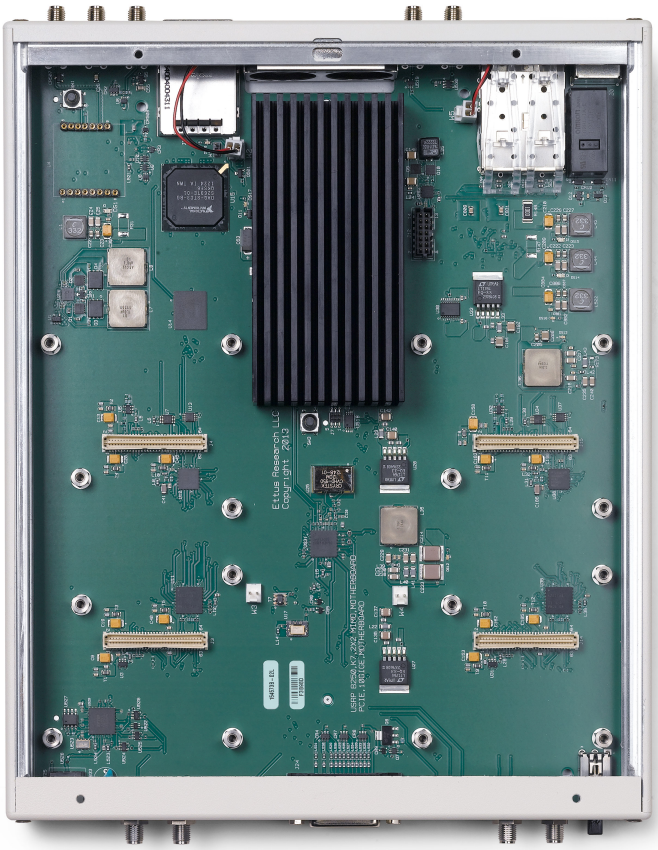


Figure 4.15: USRP X300 Internals without daughterboards

The configuration used in our testing consisted of two WBX-120 daugherboards in each USRP X300. These are full-duplex wideband transceivers which cover freqeuncies from 50 MHz to 2.2 GHz with a 120 MHz bandwidth.

Each USRP X300 was attached to a Linux workstation which utilized GNU Radio to con-

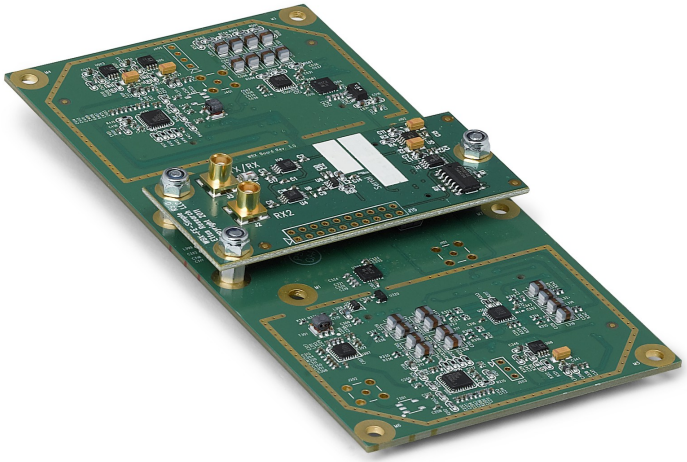


Figure 4.16: WBX-120 Daugherboard

trol the hardware via the high speed ethernet interface. The SDRs were located approximately 3 ft apart.

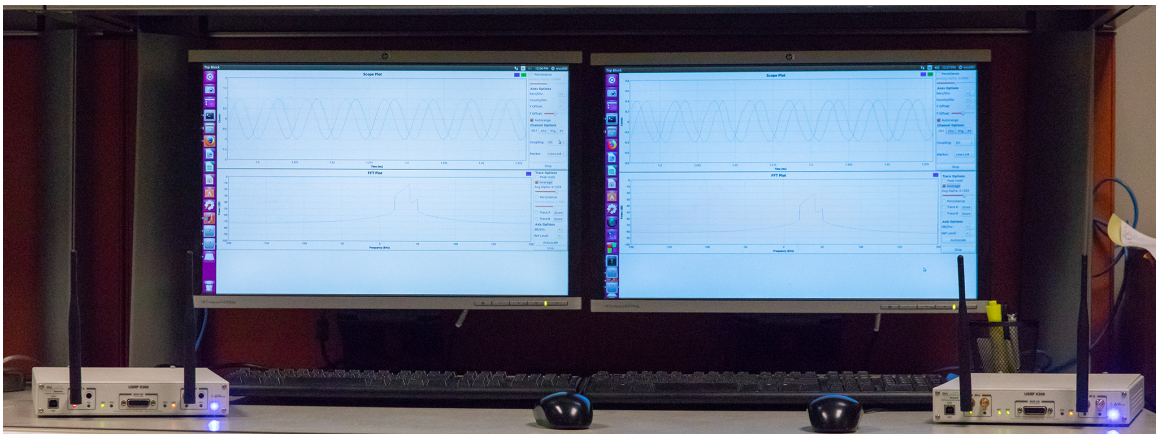


Figure 4.17: USRP X300 Setup

4.3.2 SDR Control

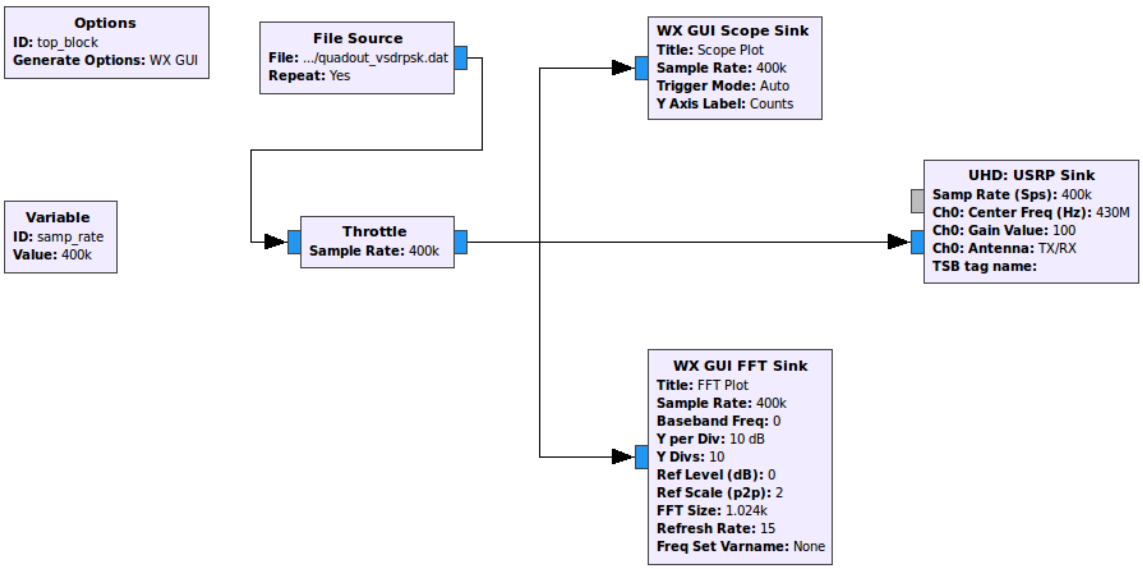


Figure 4.18: SDR Transmit Block Diagram

Above we can see a screenshot from the GNU Radio companion showing the block diagram for the transmit side of the SDR setup. The transmission file source is generated via a Matlab script which generates the components of the signal to be transmitted. This is similarly modified as the code above to change the random phase generation to a chaotic phase generator. Because of this the bit durations no longer have to be transmitted. Figure 4.19 below is a plot of the transmit signal output for a VSDRBPSK modulated signal with random phase. As the plots typically show the instantaneous received frequency, the averaging function was enabled allowing us to see the sweep from 20 kHz to 50 kHz.

For our test case we performed a linear chirp from 20 kHz to 50 kHz with a sweep duration of 2 seconds and a 1 second delay between chirps. The software defined radios were set to utilize a center frequency of 430 MHz at both the transmit and receive side.

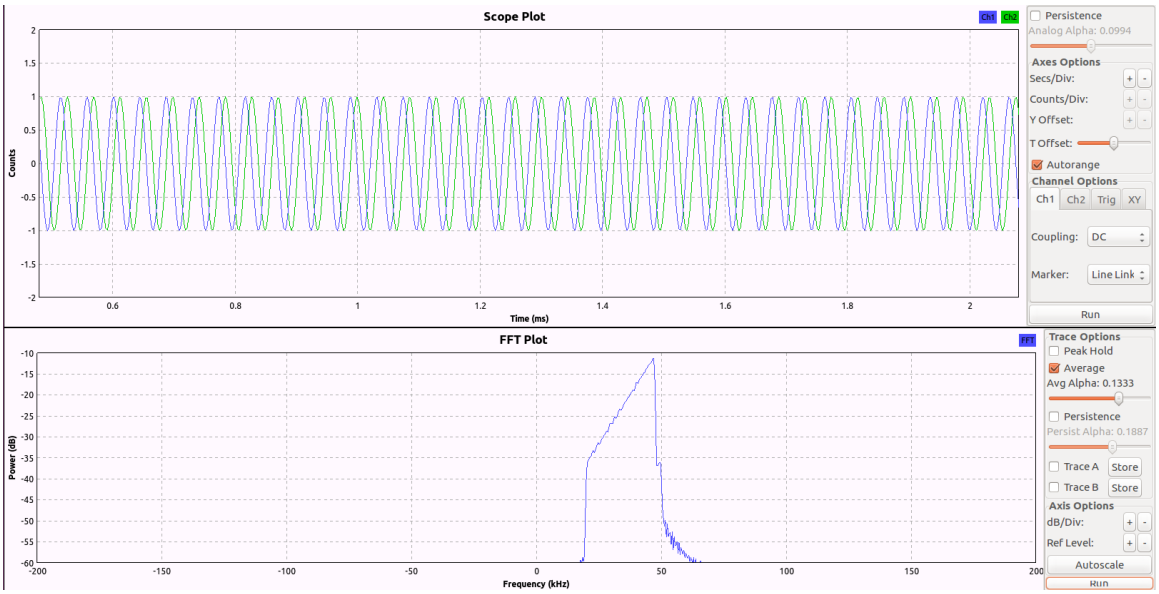


Figure 4.19: SDR Transmission Plots - Random Phase

Figure 4.20 below shows a VSDRBPSK modulated signal with chaotic phase generation.

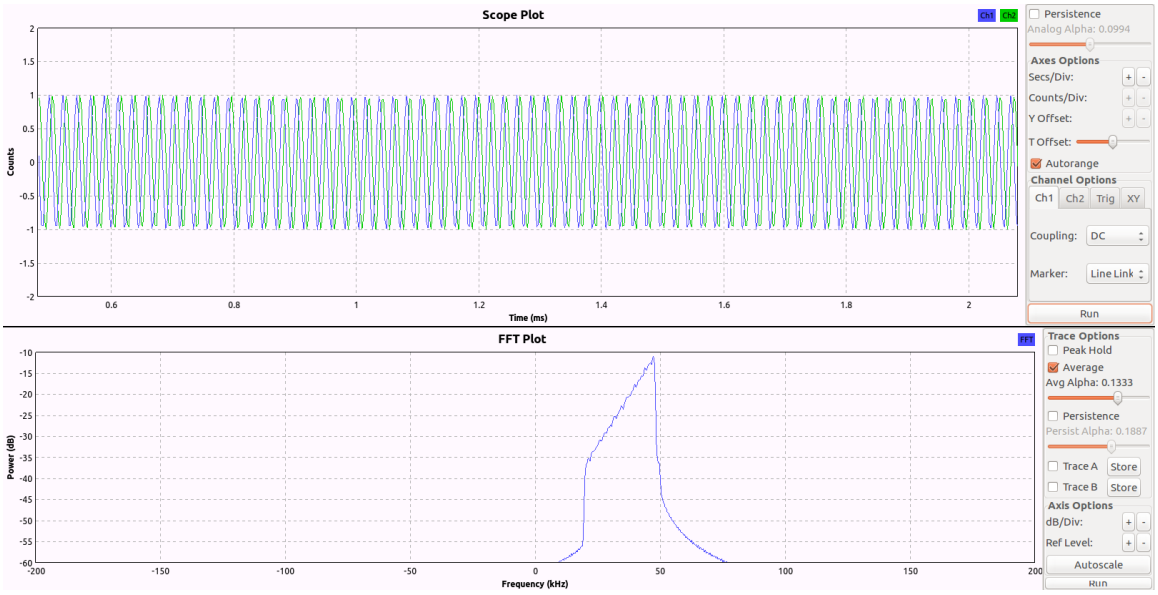


Figure 4.20: SDR Transmission Plots - Chaotic Phase

Next we look at the receiver block diagram:

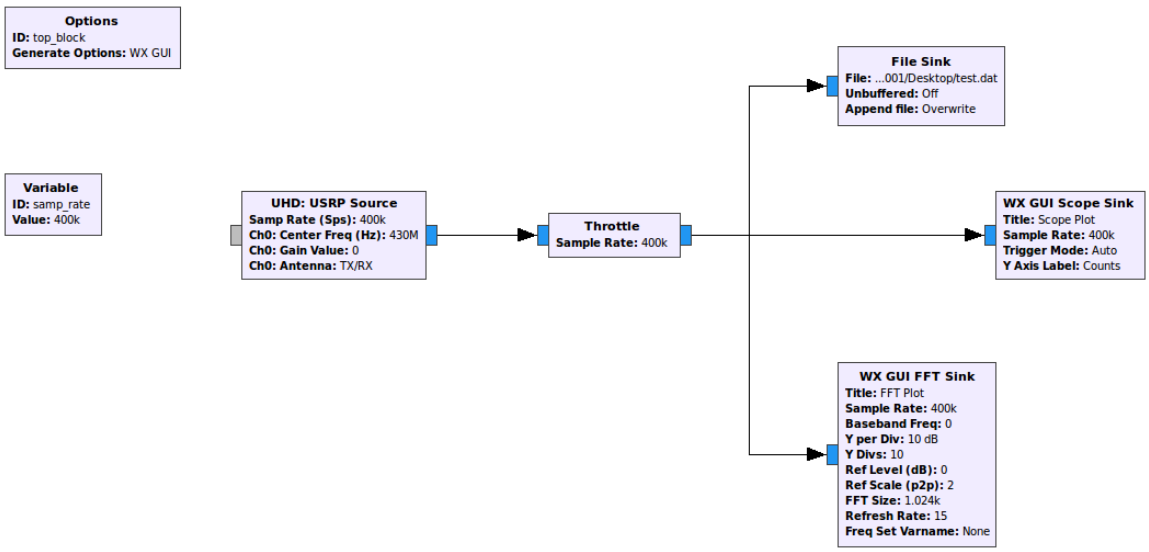


Figure 4.21: SDR Receive Block Diagram

Here we see the data is received and saved to a file where it can be demodulated by our Matlab script. Because we no longer receive the bit durations some additional changes had to be made to the demodulation code in order to be able to get this working properly, but overall the process was similar to what was discussed in the previous chapters.

The following plot shows the SDR received signal with the random phase generation:

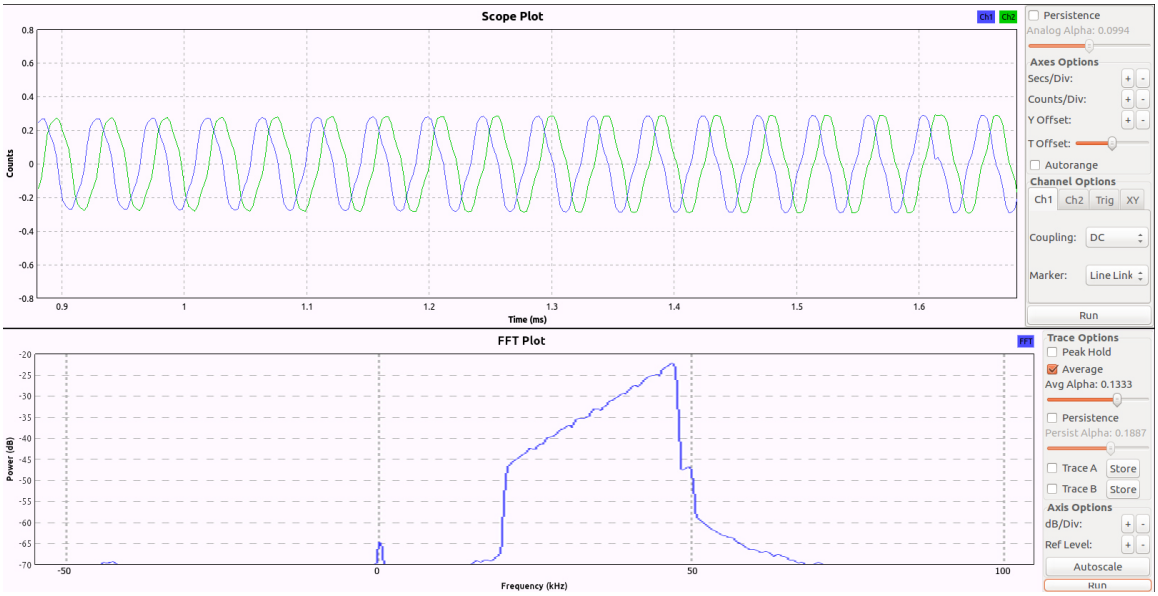


Figure 4.22: SDR Receive Plots - Random Phase

Next, we have the following plot showing the SDR received signal with the chaotic phase generation:

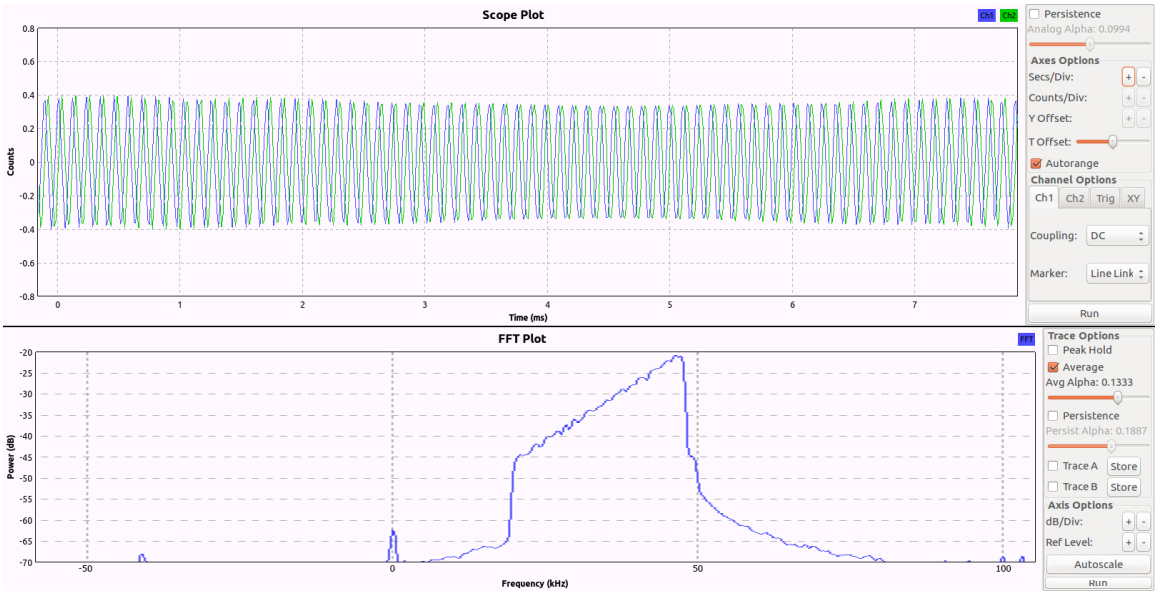


Figure 4.23: SDR Receive Plots - Chaotic Phase

Prior to examining the new data lets first look at the plots from a BPSK modulated signal so we have a point of reference. First lets look at what is transmitted for a BPSK modulated signal. All signals will be transmitted similarly, just changing the modulation scheme.

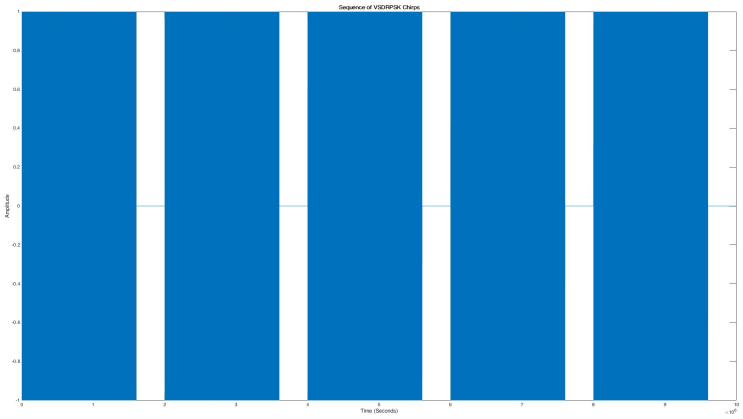


Figure 4.24: BPSK Modulated Chirps

Because of the high frequency it is difficult to see the waveform. In order to get a better view we will next look at a spectrogram so that we can see the frequency sweep.

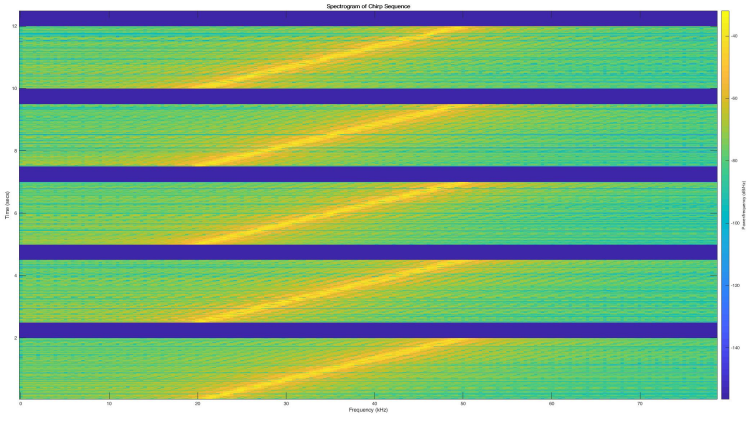


Figure 4.25: Spectrogram - BPSK

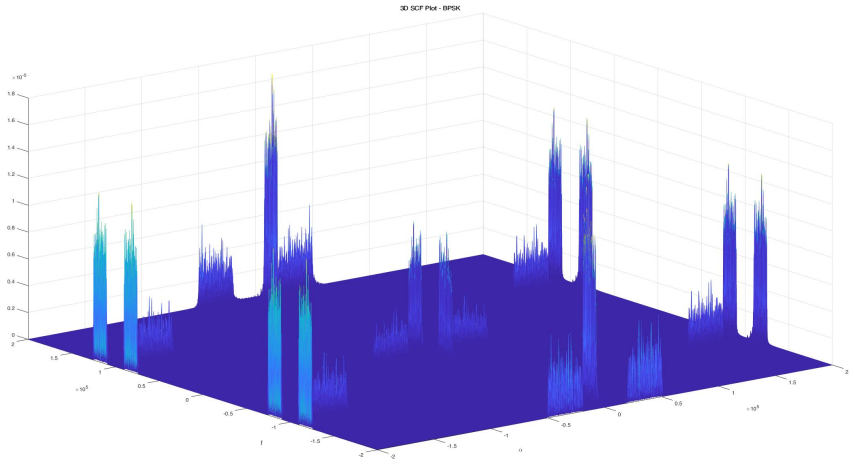
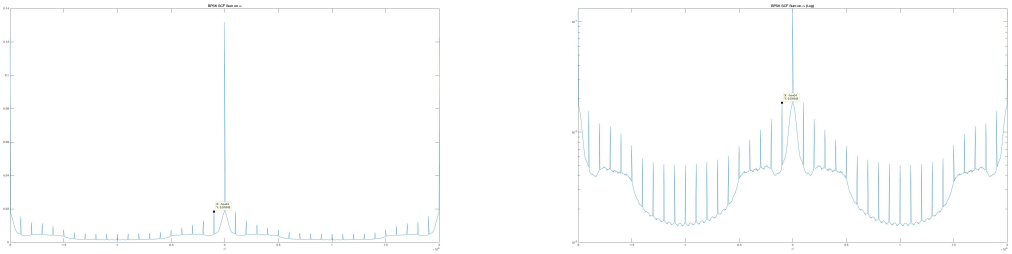


Figure 4.26: 3D SCF Plot - BPSK

First an overall look at the 3D SCF Plot. To identify features we will look along the alpha domain to identify the frequency of the modulation. This shows the features we hope to eliminate using our LPD waveform, to make the features more obvious we use a logarithmic plot and we can see the peaks corresponding with multiples of the symbol rate of 10 kHz.



(a) Linear (b) Logarithmic

Figure 4.27: SCF Plot on α - BPSK

Once we completed the implementation we began looking at the SCF of the received signal and comparing the results versus the original randomly generated phase SCF. First we'll look at the 3D SCF plot.

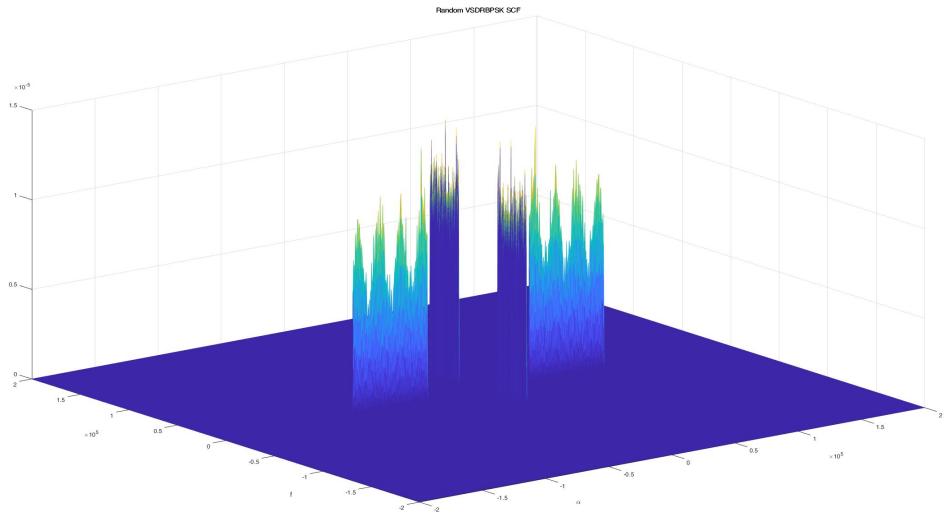


Figure 4.28: 3D SCF Plot - Random Phase

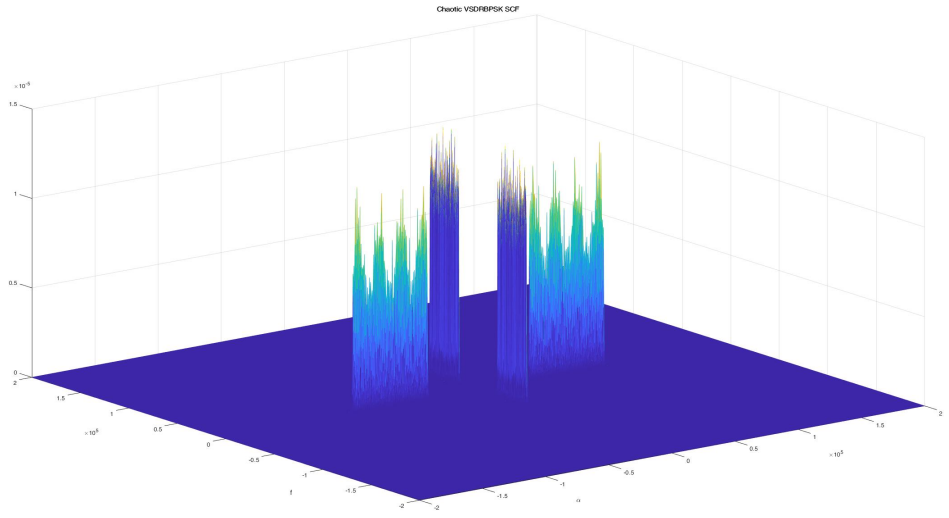


Figure 4.29: 3D SCF Plot - Chaotic Phase

Next we'll look at the SCF along the cyclic frequency axis to try and identify any cyclostationary features.

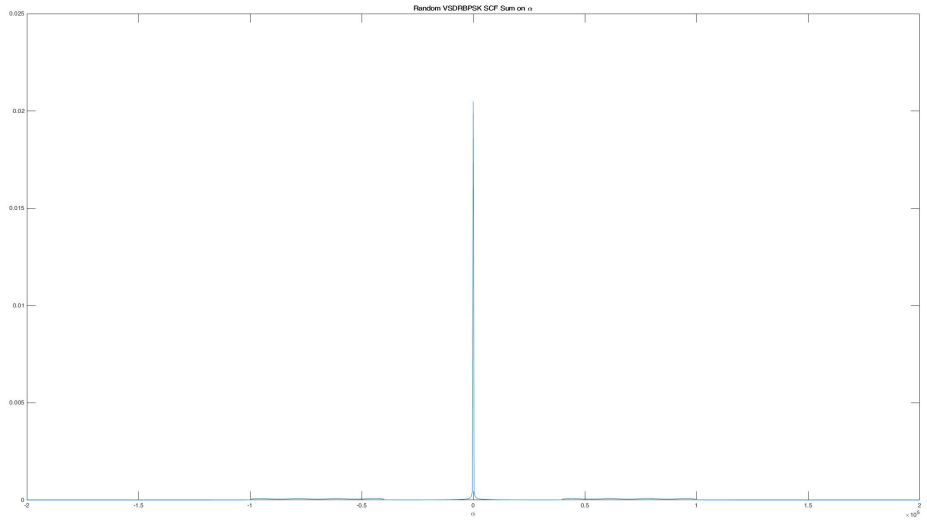


Figure 4.30: SCF Plot on α - Random Phase

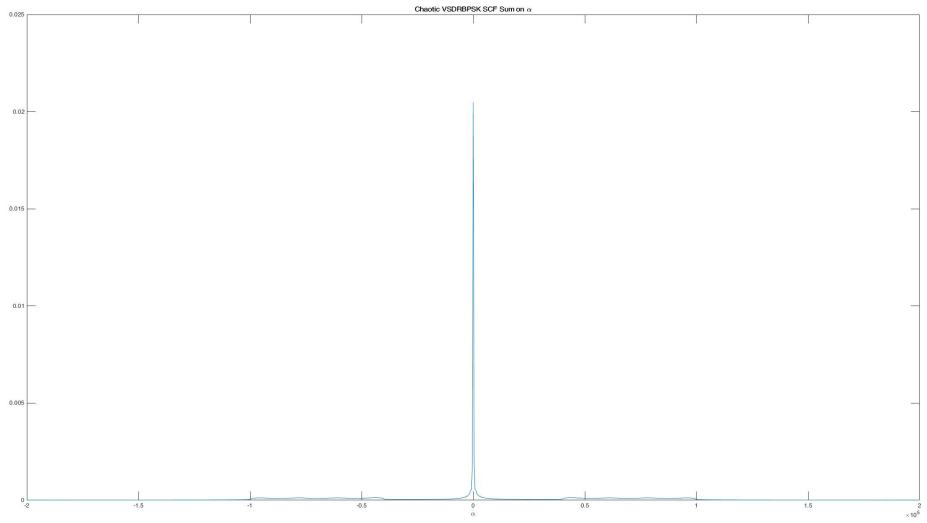


Figure 4.31: SCF Plot on α - Chaotic Phase

In order to more closely look at these two we will look at this plot again on a log scale for the magnitude.

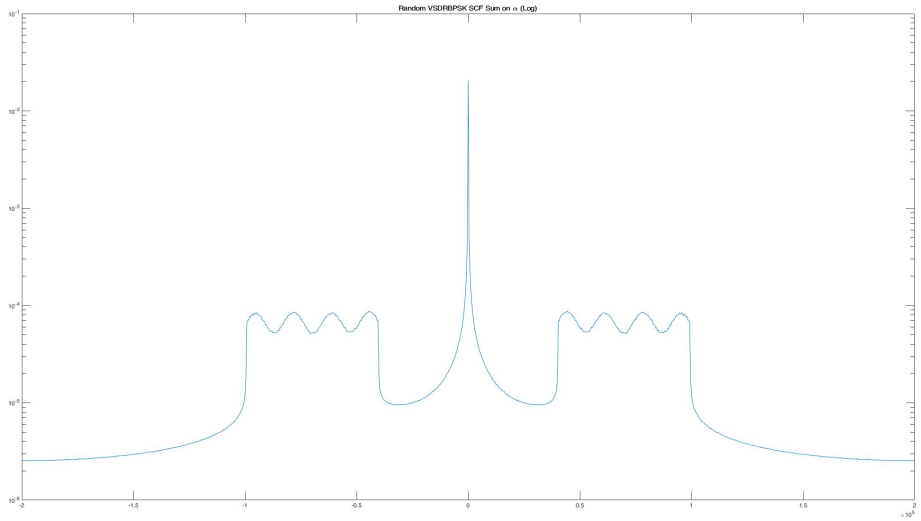


Figure 4.32: SCF Log Plot on α - Random Phase

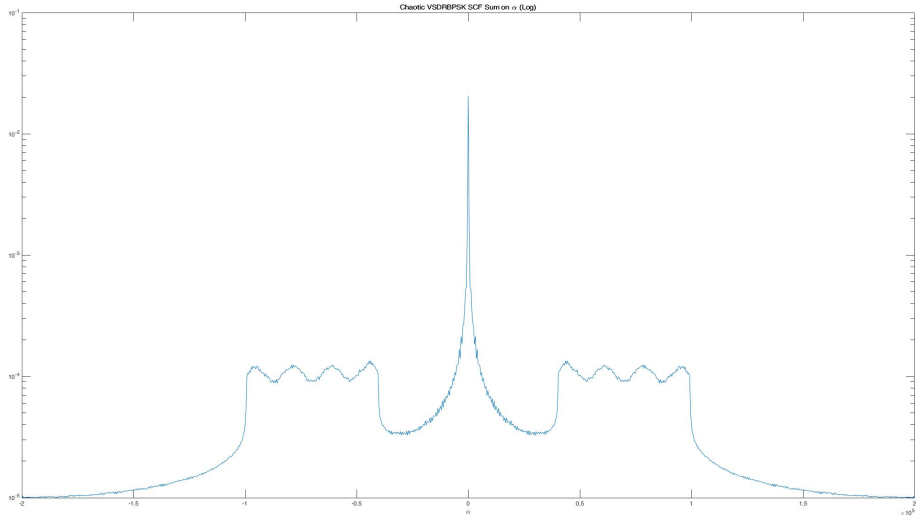


Figure 4.33: SCF Log Plot on α - Chaotic Phase

Finally we will look at the magnitude plot of frequency versus the cyclic frequency.

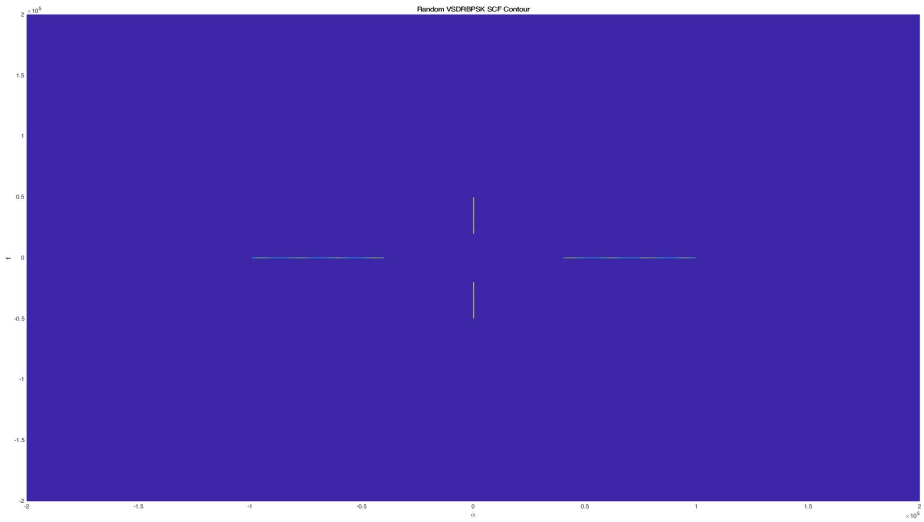
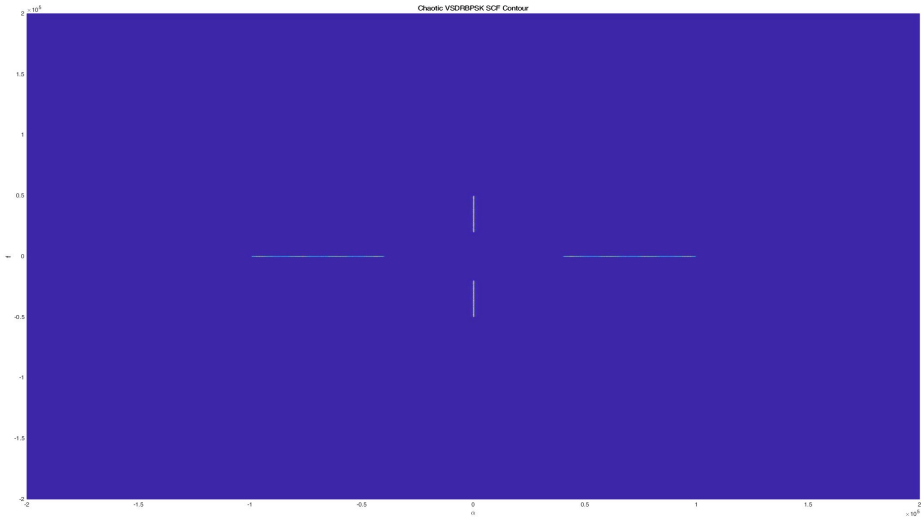


Figure 4.34: SCF Frequency Vs α - Random Phase



captionSCF Frequency Vs α - Chaotic Phase

Conclusion

Overall this method of synchronization provides many benefits:

- Chaotic sequences are more random providing extra security
- Chaotic generator eliminates need for entire length of pseudo-random sequence to be known, just a seed
- Can easily change the seed value if sequence is compromised
- Maintains LPD Characteristics

5.0.1 Further Research

In the course of this project a logistic map was decided due to its simplicity and the fact that it was bounded. One topic of further research would be to investigate the merits of using other chaotic sequence generators. Other sequence generators could be unbounded and may require additional work.

Additionally, the distribution of the chaotic sequence was found to be not uniform and could be another factor to consider when investigating alternative chaotic sequences. A uniformly distributed chaotic sequence would be ideal for efficient use of the spectrum and to reduce the probability of detection by cyclostationary analysis. A comparison of the probability density functions below.

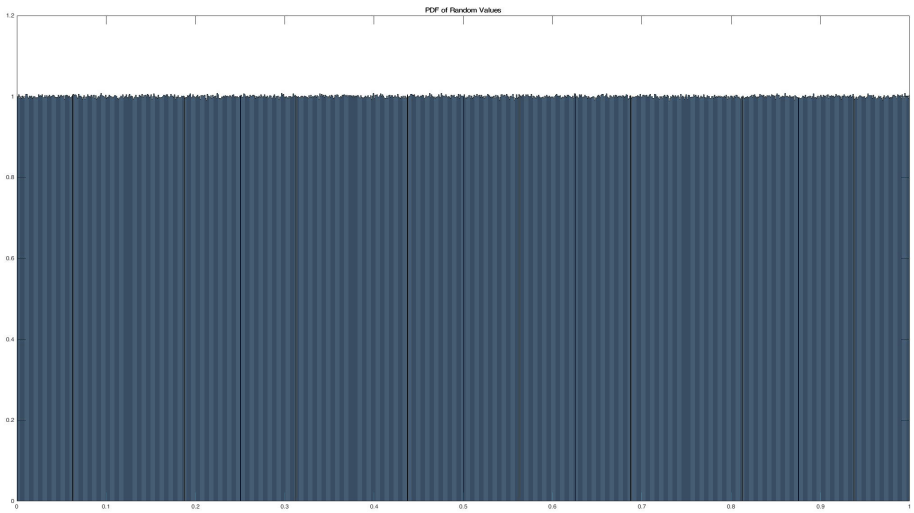


Figure 5.1: Probability density function - Random Sequence

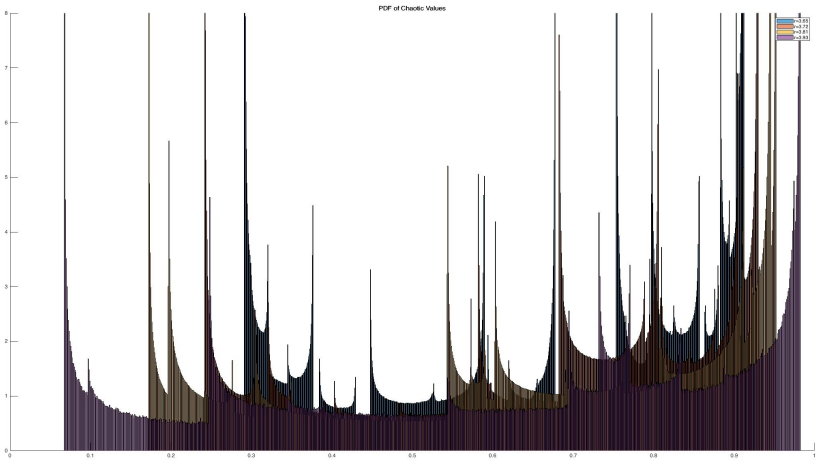


Figure 5.2: Probability density function - Chaotic Sequence

Bibliography

[1] A brief history of cryptography. http://www.cypher.com.au/crypto_history.htm, January 2006.

[2] William A. Gardner. The spectral correlation theory of cyclostationary time-series. *Signal Processing*, 11:13–36, 1986.

[3] William A. Gardner. Cyclostationarity in communications and signal processing. *University of California, and Statistical Signal Processing, Inc., Tech. Rep.*, 1993.

[4] Norbert Wiener. Generalized harmonic analysis. *Acta Mathematica*, 55:117–258, 1930.

[5] Chad Spooner. Cyclostationary processing, understanding and using the statistics of communication signals. https://cyclostationary.blog, September 2015.

[6] William A. Gardner and Chad M. Spooner. The cumulant theory of cyclostationary time-series part 1: Foundation. *IEEE Transactions on Signal Processing*, 42:3387–3408, 1994.

[7] R. Schoolcraft. Low probability of detection communications-lpd waveform design and detection techniques. *IEEE MILCOM*, 1991.

[8] Zhiping Zhang, Michael J. Nowak, Michael Wicks, and Zhiqiang Wu. Bio-inspired rf steganography via linear chirp radar signals. *IEEE Communications Magazine*, 54(6):82–86, June 2016.

[9] P. Stavroulakis. *Chaos Applications in Telecommunications*. CRC Press, 2006.

[10] G. Boeing. Visual analysis of nonlinear dynamical systems: Chaos, fractals, self-similarity and the limits of prediction. *Systems*, 4(4):37, 2016. doi:10.3390/systems4040037.

[11] A. Abel and W. Schwarz. Chaos communications - principles, schemes, and system analysis. *Proceedings of IEEE Trans. Circuits Syst. I*, 90(5):691–710, May 2002.

[12] A. F. M. Nokib Uddin Md. Zahid Hasan, Iftekhar Idris and Md. Shahjahan. Performance analysis of a coherent chaos-shift keying technique. *International Conference on Computer and Information Technology (ICCIT)*, 2012. doi:10.1109/ICCITechn.2012.6509721.

[13] Vasu Chakarvarthy Dan Sundersingh, Yao Ma and Zhiqiang Wu. Multiuser chaos communication through polyphase spreading for overlay cognitive radio. *International Waveform Diversity Design Conference (WDD)*, 2012. doi:10.1109/WDD.2012.7311271.

[14] Patrick F. Dunn. *Measurement and Data Analysis for Engineering and Science*. McGraw-Hill, New York, 2005. ISBN 0-07-282538-3.

[15] Kunio Takaya David E. Dodds and Qingyi Zhang. Frame synchronization for pilot symbol assisted modulation. *IEEE Canadian Conference on Electrical and Computer Engineering*, 1999.

[16] R. H. Barker. Group synchronizing of binary digital systems. *Communication Theory*, 1953. London: Butterworth. pp. 273287.

[17] Yang Qu Zhiping Zhang, Michael Wicks Michael J. Nowak, John Ellinger, and Zhiqiang Wu. Rf steganography via lfm chirp radar signals. *IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS*, 54(3), June 2018.