AN ALTERNATIVE APPROACH TO DIRECT SEQUENCE SPREAD SPECTRUM ACQUISITION

A graduate project submitted in partial fulfillment of the requirements
For the degree of Master of Science in
Electrical Engineering

By

Eric Neal VanLandingham

December 2012
The graduate project of Eric Neal VanLandingham is approved:

________________________________________  ______________________________________
Deborah van Alphen, Ph.D.                  Date

________________________________________  ______________________________________
James Flynn                                  Date

________________________________________  ______________________________________
Sharlene Katz, Ph.D., Chair                 Date

California State University, Northridge
ACKNOWLEDGEMENT

I would like to thank my committee members, professor Katz, Flynn and van Alphen. I truly am grateful for their commitment and dedication. Professor Katz's many hours of review, guidance and feedback provided me the confidence and direction I needed to achieve and develop this report. Professor Flynn's ingenuity, technical expertise and ease at explaining complicated material allowed me to develop and succeed in my efforts. Without each of their help through my educational experience at California State University, Northridge I would have not been able to accomplish my achievements.

I would also like to thank my motivational support group, my family. My beautiful wife Carinn for her constant encouragement and understanding of the time I needed to spend away attending to my studies. To my children, Amelia, Neal, Breanne and Sophie who sacrificed time with their dad so that I could achieve my goals. You all provided me with the inspiration and will power to spend the time needed to finish my project and report. Thank you!
# TABLE OF CONTENTS

SIGNATURE PAGE...........................................................................................................ii

ACKNOWLEDGEMENT....................................................................................................iii

LIST OF FIGURES............................................................................................................vii

I. INTRODUCTION........................................................................................................1

II. BACKGROUND.........................................................................................................3

2.1 Binary Phase Shift Keying....................................................................................3

2.2 Spread-spectrum/Direct Sequence Spread Spectrum...........................................4

2.3 Pseudonoise Code Synchronization.....................................................................7

2.4 Universal Software Radio Peripheral....................................................................9

III. SYSTEM DESCRIPTION........................................................................................11

3.1 DSSS Simulation..................................................................................................11

3.1.1 Overview..........................................................................................................11

3.1.2 Transmitter......................................................................................................12

3.1.3 Receiver...........................................................................................................15

3.2 Impulse Response Based PN Code Acquisition................................................16

3.2.1 Calculated Impulse Response.........................................................................17

3.2.2 Delay Locked Loop.........................................................................................22

3.2.3 Pseudonoise Code Replica.............................................................................23
3.2.4 Decision Logic ................................................................. 24

3.3 Simulation of a Direct Sequence Spread Spectrum Link with Impulse Response based PN Code Acquisition ................................................................. 25

3.3.1 Channel ............................................................................. 25

3.3.2 Transmitter ........................................................................ 26

3.3.3 Receiver ............................................................................. 28

3.4 Design of an over the Air DSSS Link with Impulse Response based PN Code Acquisition ................................................................................................. 40

3.4.1 Transmitter ........................................................................ 41

3.4.2 Receiver ............................................................................. 42

IV. SYSTEM PERFORMANCE ......................................................................................... 46

4.1 Bit Error Rate ........................................................................ 47

4.1.1 DSSS/BPSK simulation ..................................................... 47

4.1.2 SsABoIR System Simulation .............................................. 48

4.1.3 SsABoIR System Over The Air Link ................................. 50

4.2 Acquisition and Tracking Time ................................................ 51

4.3 Limitations and Constraints ..................................................... 53

4.3.1 Impulse Response and Noise ............................................. 53

4.3.2 Data Recovery Block and Noise ....................................... 54

V. CONCLUSION ...................................................................................... 56
LIST OF FIGURES

Figure 1: Binary data sequence (top), BPSK signal (bottom) .................................................. 4

Figure 2: Binary data message m(t) (top), PN code p(t) (middle), product of data, m(t) and PN code, p(t) (bottom) .................................................................................................................. 5

Figure 3: m(t) data message spectrum (left), p(t) × m(t) spread spectrum (right) ........ 6

Figure 4: Serial Search method [4] .......................................................................................... 8

Figure 5: Implementation of software defined radio S,AB,IR system .................................. 9

Figure 6: DSSS simulation model ............................................................................................. 12

Figure 7: DSSS Transmitter Stages ....................................................................................... 13

Figure 8: DSSS Transmitted Spectrum .................................................................................. 13

Figure 9: Simulink Data message generator ........................................................................... 14

Figure 10: PN code generator ............................................................................................... 14

Figure 11: DSSS Receiver model .......................................................................................... 15

Figure 12: Decision Logic ....................................................................................................... 16

Figure 13: Receiver Impulse Response acquisition system ................................................ 17

Figure 14: Delay due to the transmitter, channel and receiver ............................................. 18

Figure 15: Impulse Response amplitude relative to channel delay offset ....................... 19

Figure 16: h(t) with channel delay of 512 samples or 32ms ................................................. 20

Figure 17: Frequency response of PN Code, X(f) ................................................................. 21

Figure 18: Channel representation ......................................................................................... 26

Figure 19: Transmitter block diagram .................................................................................. 27

Figure 20: PN Code Sequence Generator Block Parameter .............................................. 27

Figure 21: Ideal Rectangular Pulse Filter Tx block parameters ......................................... 28
Figure 22: Calculated Impulse Response subsystem .......................................................... 29
Figure 23: Delay Locked Loop .......................................................................................... 30
Figure 24: Peak Finder Block parameters ........................................................................ 31
Figure 25: PN delay logic subsystem ................................................................................ 32
Figure 26: Valid Delay Logic ............................................................................................ 32
Figure 27: Switch block parameters .................................................................................. 33
Figure 28: Delay Measurement subsystem ....................................................................... 33
Figure 29: Measured delay valid with data (top), measured delay valid without data
(bottom).......................................................................................................................... 34
Figure 30: Reference Delay Measurement Subsystem ....................................................... 36
Figure 31: Relay block parameters ................................................................................... 37
Figure 32: Trigger Locked Subsystem .............................................................................. 38
Figure 33: PN Code Replica ............................................................................................. 39
Figure 34: Decision Logic Subsystem .............................................................................. 40
Figure 35: S_oAB_oIR Transmitter ..................................................................................... 42
Figure 36: USRP Transmitter block parameters ............................................................... 42
Figure 37: S_oAB_oIR Receiver ......................................................................................... 43
Figure 38: USRP Receiver block parameters ................................................................... 44
Figure 39: Enabled S_oAB_oIR subsystem ....................................................................... 45
Figure 40: Bit Error Rate for theoretical BPSK and DSSS/BPSK .......................................... 48
Figure 41: Data pattern of received bit sequence of the S_oAB_oIR system over the air .... 50
Figure 42: Minimum synchronization time with respect to $EbN0$ .................................... 52
Figure 43: Maximum synchronization time with respect to $EbN0$ .................................... 52
Figure 44: Impulse Response spike without noise (Left) and with noise (Right) .......... 54

Figure 45: Picture of $S_0AB_0\text{IR system}$ ................................................................. 56
ABSTRACT

AN ALTERNATIVE APPROACH TO DIRECT SEQUENCE SPREAD SPECTRUM ACQUISITION

By

Eric Neal VanLandingham

Master of Science in Electrical Engineering

In this report an alternative approach to Direct Sequence Spread Spectrum acquisition and tracking will be explained and implemented. The system designed uses the impulse response of the channel to acquire and track the time difference between the locally generated pseudonoise code and received signal's pseudonoise code. The results of both simulation and over the air tests confirm the validity of this alternative approach. Several limitations and constraints that are innate in the technique and design used were also identified. Additional research and modifications that would improve the performance of this system are suggested.
I. INTRODUCTION

This report describes a project to develop and implement an alternative approach to Direct Sequence Spread Spectrum (DSSS) acquisition and tracking system. The name of the system is the Spread-spectrum Acquisition Based on Impulse Response or SsABoIR. The SsABoIR system acquires and tracks the time difference of a locally generated pseudonoise code and received signal's pseudonoise code by using the calculated impulse response of the channel\(^1\).

The SsABoIR system is implemented as part of a software defined radio. The software used for the development and test of this system is Simulink. The hardware used, which enables it to become an over the air communication link, is the Ettus Universal Software Radio Peripheral (USRP) [1]. In today's industry there is tremendous application for the development of software defined radios. The increasing growth in cellular phones and their multifunctional capabilities verifies that logically a move toward software defined radios is beneficial. Along with current cellular technology the Long Term Evolution (LTE) cellular standard implements the technology of DSSS [2]. The development of the SsABoIR system provides an alternative approach to traditional DSSS acquisition and tracking using software defined radio technology.

This report describes the SsABoIR system along with its development and testing. Section 2 provides background information on Binary Phase Shift Keying, DSSS and the hardware used in the development of the SsABoIR system. Section 3 presents a baseline

\(^1\) The SsABoIR system was originally suggested by Professor James Flynn of California State University, Northridge.
model of DSSS, an overview of how the S_{3,AB_{0}}IR system will work, the implementation of a simulation of the S_{3,AB_{0}}IR system and the design of the over the air S_{3,AB_{0}}IR system. Section 4 summarizes the performance determined by simulations and over the air tests. Section 5 provides the summary and suggestions for future work.
II. BACKGROUND

In this section the primary communication system concepts used in this report will be discussed to provide a better understanding of the system. These concepts consist of Binary Phase Shift Keying, Direct Sequence Spread Spectrum, Pseudonoise Code Synchronization and the Universal Software Radio Peripheral.

2.1 Binary Phase Shift Keying

There are several different methods of modulating a carrier wave that allows one to send data messages. Equation 1 shows the different characteristics of a carrier wave. Using the amplitude (A), frequency (\(w_0\)), phase (\(\theta\)) of an RF carrier, or a combination of them allows for the intended message to be sent over the air [3] [4].

\[
s(t) = A \cos(w_0 t + \theta(t))
\]

In this project Binary Phase Shift Keying (BPSK) will be used as the modulation method to send information over the communication link. The term “binary” in BPSK represents the data bits 0 and 1, however, they are accessible differently in phase shift keying. A 0 bit will be presented as a phase of \(\pi\) and the 1 bit will be presented as phase of zero. Figure 1 shows the effects on the carrier wave as the bit sequence changes over time.
2.2 Spread-spectrum/Direct Sequence Spread Spectrum

In digital communications the spread-spectrum (SS) technique is used in many applications, such as military communication systems, anti-jam protection, low probability of detection and/or interception to even the most popular application, code division multiple access found in cellular phone technology. There are several methods of SS such as direct-sequence spread-spectrum, frequency hopping and hybrid direct-sequence frequency hopping. A system is defined to be a spread-spectrum system if it meets the following requirements [4]:

1. The signal occupies a frequency bandwidth greater than the minimum bandwidth necessary to send the required information.
2. Spreading of the data signal is accomplished by means of a spreading signal, known as a pseudo random code or pseudonoise binary sequence.
3. The receiver recovers the data by despreading the signal with a synchronized replica of the pseudonoise code or spreading signal.

The most common form of spread spectrum that will be used in this project is direct-sequence spread-spectrum (DSSS). This involves multiplying the transmitted binary bipolar data sequence, \( m(t) \), by a higher rate pseudonoise (PN) binary bipolar sequence \( p(t) \), implying the term direct sequence, shown in Figure 2. The term spread spectrum refers to the spreading of the frequency spectrum of the transmitted signal, shown in figure 3. This is caused by the mixing of the higher rate binary pseudonoise sequence with the lower rate binary data sequence.

Figure 2: Binary data message \( m(t) \) (top), PN code \( p(t) \) (middle), product of data, \( m(t) \) and PN code, \( p(t) \) (bottom)
Two important concepts of spread spectrum that can be seen from figure 3 are that the power spectral density (psd) is dramatically reduced in amplitude and that the signal occupies a larger amount of bandwidth. It can be noted that the magnitude difference in the psd of the two signals is about 20dB. This will be important when recovering the data signal out of the noise. The data message m(t) contains most of its energy between 0 and 10 Hz of frequency, while the spread signal contains most of its energy between 0 and 1000 Hz.

As discussed earlier spreading a signal allows for multiple applications and benefits. However, the whole purpose of communication systems is to communicate a message. In order for the receiver to demodulate the data sequence the spread signal must be despread [4]. This can be achieved by multiplying the received signal by the same locally generated synchronized PN code. When PN codes are perfectly synchronized in time the original data will be generated. This can be visualized shown in figure 2. The received encoded data (bottom plot, figure 2) aligned in time and
multiplied with the PN code (middle plot, figure 2) produces the data sequence (top plot, figure 2). However, a slight misalignment with the PN code could cause major degradation in the data message.

When the two codes are perfectly aligned the data signal gets amplified by a parameter called processing gain [4]. Processing gain expresses the performance advantage of the spread-spectrum system over a narrowband system. As shown in figure 3 the receiver receives a -30dB signal and amplifies that data message back to its original -10dB signal, providing 20dB of processing gain. Processing gain $G_p$ can be calculated by dividing the PN rate $R_{p(t)}$ by the data rate $R_{m(t)}$ of the message, equation 2. In figure 1’s example the data rate is sent at 10 bits per second while the PN rate is 1000 chips per second, which would equate to 20dB of process gain.

$$G_p = \frac{R_{p(t)}}{R_{m(t)}} = \frac{1000}{10} = 100 = 20dB$$  \hspace{1cm} 2.

2.3 **Pseudonoise Code Synchronization**

In general for a typical DSSS receiver there are two steps in obtaining synchronization between the two PN codes [5]. The first step is PN code acquisition where the receiver determines a rough alignment of the PN code usually to within one chip interval. The second step is to achieve fine alignment using a tracking method. Some of the most widely used methods for acquiring synchronization are full parallel search, serial search and hybrid search [5]. In all of these cases correlation algorithms are run to determine the maximum likelihood of the PN code, where a threshold is set to determine alignment. In the case of full parallel search the whole PN code is used to determine correlation. However, if PN codes are long this could take time to determine
the best fit. Therefore, serial search would be the next best option for long PN codes. Serial search takes a smaller set of observations to determine correlation against the PN code, which would decrease the amount of time but increase the possibility of misalignment when a threshold is met. Hybrid search methods test a small set of possible alignments in parallel and then repeats this test on another set of observations until the correct alignment is discovered.

Figure 4: Serial Search method [4]

Figure 4 [4] illustrates the serial search method for PN code acquisition, which is one of the most popular due to the simplicity in its structure. The integrator acts as a correlator or matched filter for a fixed examination interval. The examination interval is usually a small representation of the PN code. If the result of the correlator exceeds a defined threshold, acquisition is achieved. However, if the threshold is not met the search control block increments the PN code by a fixed amount, usually $\frac{1}{2}$ a chip, and repeats this process until acquisition is achieved. It should be noted that if the examination interval is a small representation of the PN code this process could take a long time.
2.4 Universal Software Radio Peripheral

The Ettus Universal Software Radio Peripheral (USRP) is the hardware component of a software defined radio (SDR) [1]. It includes an analog to digital and digital to analog convertors among other components. The USRP equipped with the wide band daughterboard allows for transmission of RF at frequencies up to 3GHz. The USRP interfaces with Mathwork’s Simulink software which allows for development of communications systems on a software defined level instead of hardware, shown in figure 5. In this project USRPs will be used to implement the design of the DSSS communication system. One major benefit of using the USRP is the ability to process data in Simulink at the baseband level. Therefore, the USRP hardware takes the baseband DSSS signal and converts it to a DSSS signal at 412 MHz for transmission.

![Diagram](https://via.placeholder.com/150)

Figure 5: Implementation of software defined radio S\textsubscript{c}AB\textsubscript{b}IR system

Two major benefits that the USRP provides for the S\textsubscript{c}AB\textsubscript{b}IR system is the ability to perform coherent frequency detection and frame based processing. If the transmitter and receiver’s carrier frequency are different, errors would occur during the demodulation process. The USRP has the ability to receive an external reference clock from a pulse generator that enables the transmitter and receiver to synchronize the
frequency component of the carrier wave. Since digital signal processing is heavily
dependent on processing time, Simulink and the USRP allow for frame based processing.
Frame based processing minimizes the amount of operations needed and speeds up the
processing time.
III. SYSTEM DESCRIPTION

A progressive approach will be taken to explain in detail the functionality of the S₅AB₀IR system. This section will cover four parts: looking at how a basic DSSS system works within a controlled environment such as Simulink, leading into an in depth analysis of the acquisition and tracking process for the S₅AB₀IR system and how it is implemented in Simulink. Finally, an explanation and demonstration of the S₅AB₀IR system on an over the air communication link will be presented.

3.1 DSSS Simulation

In this section a Simulink model for a DSSS simulation will be developed and verified. Some assumptions will be made when describing the DSSS model in Simulink. Below is a list of assumption for the DSSS Simulink model.

I. The transmitter and receiver will have a synchronized clock; therefore, timing recovery at the receiver is not an issue.

II. The transmitter and receiver’s frequency and phase are perfectly aligned.

III. The channel does not induce any delay.

Consequently in this DSSS model there is no need for determining PN synchronization.

3.1.1 Overview

The DSSS model is broken down into three parts, transmitter, receiver and channel. Figure 6 lays out the basic model for the DSSS system. The channel is represented by an Additive White Gaussian Noise block which will set the signal to noise ratio \( \frac{E_b}{N_0} \). The DSSS system will provide a baseline model of an ideal DSSS system. This model will be used to compare with the developed S₅AB₀IR system.
Figure 6: DSSS simulation model

The system parameters are defined below:

- Data rate = 10 bits/sec
- Chip rate = 1000 chips/sec
- PN code Length = 63 chips
- Carrier frequency = 10 KHz
- Sampling rate = 80K samples/sec

3.1.2 Transmitter

The DSSS transmitter is divided into two stages. Figure 7 depicts the two stages of the transmitter. In the first stage the product of the data message \( m(t) \) and PN code \( p(t) \) performs the spreading of the signal as described in figure 2 and figure 3. The second stage creates the product of the spread signal and the carrier wave, which is known as mixing. Figure 8 illustrates the spectrum of the modulated DSSS signal.
Figure 7: DSSS Transmitter Stages

Figure 8: DSSS Transmitted Spectrum

Figure 9 and figure 10 show the block diagrams of the data message and PN code subsystems. The data message is represented by a random set of 0’s and 1’s. The PN code generator generates a sequence of pseudorandom binary numbers using a linear-
feedback shift register. The unipolar to bipolar converter takes a 0 and 1 and converts it to a -1 and 1, respectively. This process is fundamental in using the BPSK modulation scheme. From equation 3 it can be seen that BPSK is equivalent to switching the amplitude of the carrier wave between -1 and 1.

\[ x(t) = A \cos(wt + \theta(t)), \text{where } \theta(t) = 0, \pi \]

\[ x(t) = A \cos(wt + 0) \text{ or } A \cos(wt + \pi) \]

\[ x(t) = \pm A \cos(wt) \]

The use of the rectangular pulse filter adds multiple data samples to a given input value. This ensures that when operations are performed with different data rates the correct number of data samples is being used.

![Simulink Data message generator](image1)

**Figure 9:** Simulink Data message generator

![PN code generator](image2)

**Figure 10:** PN code generator
3.1.3 Receiver

It can be seen from figure 11 that the receiver model is a mirror image of the transmitter. However, the receiver has an additional process, called the decision logic. When the receiver receives the signal it also receives noise inherent in the channel and system. Therefore, a decision must be made on whether a 0 or 1 was transmitted. Before a decision can be made, two operations must be accomplished. The first operation performed on the received signal is stage 2. The product of the received signal and the carrier wave shifts the signal back down to baseband. The product performed in stage 1 despreads the signal. If the PN code is synchronized perfectly the received signal data message is extracted. Finally, a decision can be made to determine if data is a 1 or 0.

![DSSS Receiver model](image)

Figure 11: DSSS Receiver model

The decision logic subsystem shown in figure 12 is based on the maximum likelihood rule. The Integrate and dump acts as the correlation process. Since only two symbols are transmitted the receiver only needs to determine if the received symbol correlates with one of the symbols. In the case of the DSSS receiver it is attempting correlation with symbol 1. It should be noted that the DSSS receiver knows when a bit begins and ends, making the integration period fixed and known. In the noise free case,
if a received symbol correlates with symbol 1 the resulting value would be greater than zero. If the received symbol does not correlate with symbol 1 the resulting value would be less than one. Therefore, a threshold is set to zero using the sign block shown in figure 12. The lookup table translates -1, 0 and 1 into binary data of 0 and 1. The correct operation of this simulator will be verified in section 4 of this report.

![Decision Logic Diagram](image)

Figure 12: Decision Logic

3.2 Impulse Response Based PN Code Acquisition

As discussed in section 2.1.3 the most common method for a PN code acquisition system is through the use of a correlation process and a delay locked loop. The $S_{AB_oIR}$ simulation system, shown in figure 13, produces an alternative approach to PN code acquisition. This approach uses the impulse response of the channel to extract the time difference between the locally generated PN code and the received signal's PN code. A delay between the two PN codes is introduced by the channel along with the synchronization error between the transmitter and receiver. This section will cover a basic overview of the simulation model, defining how the $S_{AB_oIR}$ system works. Section 3.3 will explain in detail the implementation of each subsystem within the simulation.
3.2.1 Calculated Impulse Response

Four subsystems construct the SABIR system: The Calculated Impulse Response, The Delay Locked Loop, The PN Code Replica and the Decision Logic. The keystone subsystem is the Calculated Impulse Response. This system computes the impulse response of the channel as shown in figure 14. From equation 4 it can be seen that the transfer function \( H(f) \) of the system can be calculated by dividing the Fourier transform of the received signal \( Y(f) \) by the Fourier transform of the locally generated PN code \( X(f) \) [6]. Since the received signal is essentially the transmitted signal delayed in time by \( t_o \), the Fourier transform time shifting property is applied with a resultant \( e^{-jwt_o} \) [6]. Taking the inverse Fourier transform of the transfer function will generate the impulse response of the system [6] [3]. Equation 4 shows that the impulse response of the channel is defined by a delta function shifted in time by the innate delay (\( t_o \)) between the PN code generator of the transmitter and receiver.
The Calculated Impulse Response subsystem computes this impulse response from the received signal and the local receiver PN code generator. The Fourier transforms required in this process are found using the Discrete Fourier Transforms (DFT) with Simulink. An important concept when using the DFT and inverse DFT (IDFT) is the number of samples used to represent the PN code and received signal. It would be ideal if the set of samples or length of the DFT/IDFT was sufficiently large so that the samples adequately represent the entire PN code. For this reason the PN code length is set low (63 chips) and the DFT/IDFT length is set high (1024 samples) to allow for proper resolution of the PN code spectrum and phase.

In the developed $S_{x}\text{AB}_{o}\text{IR}$ system the DFT/IDFT length allows for 64 chips instead of exactly 63 chips. Since the PN code is cyclic the PN code will wrap itself around the DFT/IDFT sequence, creating a one chip offset. This offset is induced by the mismatch between the DFT/IDFT length and PN code length, which creates a problem in the delay measurement. As the channel delay increases, more of the PN code will wrap itself around the DFT/IDFT sequence. The wrapped around PN code sequence generates an additional phase or time delay spike that includes a 1 chip offset. One of the spikes contains the correct phase while the other spike contains a 1 chip phase offset. The amount of wrapped around sequence is directly related to the magnitude of the spikes.
As more of the PN sequence is wrapped around the DFT/IDFT the 1 chip phase offset spike will increase in magnitude and decrease the correct delay spike. However, the two phase measurements are only noticeable when the channel delay is greater than 5% of the DFT/IDFT length. The measured delays that are less than 5% of the DFT/IDFT duration are characterized as the desired delay region. This region is desired because the magnitude of the correct delay spike is at its maximum and makes it easier for the receiver to measure the correct delay among noise. Figure 15 depicts the magnitude of the correct delay spike as the channel delay increases.

![Impulse Response vs Channel Delay](image)

Figure 15: Impulse Response amplitude relative to channel delay offset

The magnitude of the phase measurement depends on how many chip sequences are wrapped around the DFT/IDFT. It can be seen from figure 15 that as the channel delay approaches $\frac{1}{2}$ the length (512 samples or 32ms) of the PN code, the magnitude of the spike approaches 0.5. The magnitude is decreasing because portions of the sequence
are being binned in the adjacent bin due to the 1 chip offset. Figure 16 illustrates this process. The duration of the PN code is 63 ms long, however, the duration of the DFT/IDFT is 64 ms long. For this example the induced channel delay is \( \frac{1}{2} \) the length of the PN code or 32 ms. This is indicated by a spike at the 32ms time slot. There is also an adjacent spike at the 33ms time slot. This is due to the other \( \frac{1}{2} \) PN sequence wrapped around produced by the channel delay and cyclic nature of the PN code. The wrapped around sequence is delayed by an additional chip (1ms) due to the 1 chip offset in the sequence. To ensure that the correct delay is measured, a reference delay is introduced in the Delay Locked Loop subsystem to allow for more efficient acquisition and tracking of the channel delay \( (t_0) \). The purpose of the reference delay is to shift the measured delay to the desired delay region or 5\% of the DFT/IDFT length. Further explanation of the reference delay will be covered in the Delay Locked loop subsystem section.

Figure 16: \( h(t) \) with channel delay of 512 samples or 32ms
Another important characteristic of the S\textsubscript{AB}oIR system is how the frequency spectra of \(X(f)\) behaves. From equation 4 it can be seen that, as \(X(f)\) becomes very small, the transfer function will approach large values. This could cause false delay measurements and problems in acquiring and tracking the channel delay, especially in noise. Figure 17 illustrates the frequency spectrum of the PN code \((X(f))\). The magnitude of the spectrum has nulls and very small values at multiples of the chip rate (1kHz), which would excite the system.

![Frequency response of PN Code, X(f)](image)

In the S\textsubscript{AB}oIR system the magnitude of the transfer function is not as important as the phase. The phase information of the transfer function provides the time difference between the two PN codes or channel delay. The S\textsubscript{AB}oIR system will generate a new transfer function \((H(f)')\) to avoid the need to divide by the small values in the input spectrum. Since the output of the DFT is a complex number, complex analysis can be applied to determine the new transfer function \((H(f)')\), utilizing the phase difference of the
two signals. Equation 5 proves how the phase difference is extracted without exciting the system with very small values in the denominator. By multiplying the numerator \( Y(f) \) and denominator \( X(f) \) by the complex conjugate of the denominator \( X^*(f) \) it is shown that the phase of \( H(f) \) is the same as the phase of \( Y(f)X^*(f) \), which eliminates the need to divide by \( X(f) \). The S\textsubscript{AB}\textsubscript{0}IR system ignores the magnitude value of the original transfer function and sets the magnitude of the new transfer function, \( H(f)' \) to a constant of value 1. The angle of the new transfer function provides the time difference or phase information needed to determine the channel delay.

\[
\frac{Y(f)}{X(f)} = \frac{a + ib}{c + id} = \frac{Y(f)X^*(f)}{X(f)X^*(f)}
\]

\[
\frac{Y(f)X^*(f)}{X(f)X^*(f)} = \frac{(a + ib)(c - id)}{c^2 + d^2} = \left(\frac{1}{c^2 + d^2}\right)(ac + bd + i(bc - ad))
\]

\[
\angle \frac{Y(f)}{X(f)} = \tan^{-1}\left(\frac{bc - ad}{ac + bd}\right) = wt_o
\]

\[
H(f)' = 1\angle \tan^{-1}\left(\frac{bc - ad}{ac + bd}\right) = e^{-jwt_o}
\]

### 3.2.2 Delay Locked Loop

The Delay Locked Loop subsystem determines the delay needed to synchronize the received PN code and locally generated PN code. This is done by determining the sample number of the maximum peak within the impulse response sequence. Since figure 16 illustrates an ambiguity between which peak is the correct delay to synchronize the PN codes, the Delay Locked Loop introduces a reference delay which is applied to the locally generated PN code. By applying a reference delay to the locally generated PN code the spike \((t_o)\) is shifted to the left by the reference delay \((\tau)\) value as shown in equation 6. This shift allows for the maximum peak to reside in the desired delay region.
(figure 15) for measurement. Shifting the spike into the desired delay region allows for a minimal amount of sequence wrap around in the DFT/IDFT. Reducing the amount of wrap around increases the magnitude of the correct phase delay and decreases the magnitude of the 1 chip offset phase delay. The importance of this process will be illustrated in section 4, System Performance. Ensuring that the magnitude of the measured delay is at its maximum provides the optimal signal to noise ratio and allows for better detection of the signal.

\[ H(f) = \frac{X(f)e^{-jwt_0}}{X(f)e^{-jwt}} = e^{-jw(t_0 - \tau)} \]

\[ h(t) = \delta(t - (t_0 - \tau)) = \delta(t - t') \]

In order for the Delay Locked Loop to know how much reference delay is needed to shift the spike into the desired delay region two criteria must be meet. First the spike must be within the desired delay region. Second, the two PN codes must be synchronized. PN synchronization is evaluated in the Decision Logic subsystem while the Delay Locked Loop monitors the measured delay. Once the measured delay \((t' = t_0 - \tau)\) reaches less than 5% of the DFT/IDFT duration or is within the desired delay region and PN synchronization has been achieved the Delay Locked Loop will stop incrementing the reference delay. Therefore, by maintaining track of the measured delay and reference delay the Pseudonoise Code Replica subsystem can extract the channel delay and synchronize its replica to the received signals PN code.

3.2.3 Pseudonoise Code Replica

The PN code replica subsystem is responsible for shifting or delaying the PN code by the measured delay \((t')\) and reference delay \((\tau)\) found by the Delay Locked Loop. The subsystem outputs two PN codes, the channel delay \((t_0)\) shifted PN code and the reference
delay (τ) shifted PN code. The reference delayed (τ) PN code is used as the input to the Calculated Impulse response subsystem which generates a new measured delay (t’). The channel delayed (t₀) PN code is used to mix with the received signal, to despread the data. In order to extract the original channel delay, t₀, the measured delay is added to the reference delay, shown in equation 7.

\[ t' = t_0 - \tau \]  \hspace{1cm} 7  
\[ t_0 = t' + \tau \]

3.2.4 Decision Logic

The final subsystem of the process is the Decision Logic. This subsystem is very similar to that of section 3.1.3 Decision Logic. However, this block introduces an additive measure of determining when the S\textsubscript{AB}\textsubscript{0}IR system has achieved synchronization. When the PN code is synchronized a 20dB processing gain is applied to the signal. The Decision Logic monitors the energy per bit and if the energy per bit meets the specified threshold, synchronization has been achieved. However, synchronization has not achieved a lock status until the measured delay reaches the desired delay region. Since the Delay Locked Loop is sequentially incrementing the reference delay, the Decision Logic will re-achieve synchronization every valid measured delay. During this process the data is not valid until the measured delay is within the desired delay region and the delay is locked into position. Once the delay is within the desired delay region a valid synchronization is achieved and data can be processed.
3.3 Simulation of a Direct Sequence Spread Spectrum Link with Impulse Response based PN Code Acquisition

To verify the functionality of the S,AB,IR system a communication link was created in Simulink. This section will take an in-depth look at the S,AB,IR Simulink model. Since the performance of baseband transmission of a DSSS system is equivalent to bandpass transmission, this model will use baseband transmission [4]. In essence, the DSSS signal will not be up converted to the carrier frequency in Simulink. It will be shown in section 3.4 that the up conversion will be implemented using the USRP hardware for the over the air link.

An important concept when performing digital signal processing is timing. The ability to process data before the next data point is received is crucial in recovering all of the data. Therefore, multiple input multiple output (MIMO) is desired to reduce the operation time required for processing. To allow for this, it will be seen in the Simulink model that a majority of the processing will be frame based rather than sample based. This has been shown to be an effective method for reducing the processing time of a Simulink model [10]. Section 3.3.1 – 3.3.3 will describe the three major subsystems of this link, the transmitter, the channel and the receiver.

3.3.1 Channel

As described in section 3.2 the channel will introduce a delay on the transmitted signal. Therefore, the channel is represented by a delay block and AWGN block, shown in figure 18. To exercise the efficiency of the S,AB,IR system, the delay block will be
varied to evaluate various delays. The AWGN block will also be used to determine performance of the system in a high noise environment.

![Channel representation](image)

Figure 18: Channel representation

### 3.3.2 Transmitter

The DSSS transmitter is very similar to that of the transmitter explained in section 3.1.2. Figure 19 depicts the block diagram used to implement the DSSS transmitter. Since the DFT/IDFT is performed using frames the frame rate was chosen to ensure that the complete PN code could be represented. Figure 20 shows the PN Sequence Generator block parameter used for the SABIR system.

This block creates a PN code of length 63 \((2^6 - 1)\). The Random Integer Generator represents the data sequence at a rate of 10 bits per second (bps). In order to match the PN code frame rate the data sequence is up sampled by the rectangular pulse filter block (in samples per bit) and buffered into frame based vectors of length 64. The reason the data sequence cannot be generated into frames initially is because the data rate is 100 times slower than the PN code. In order to achieve the same frame rate it must be sampled by the rectangular pulse filter first.
After mixing the data with the PN code, the final Ideal Rectangular Pulse Filter block up samples the entire DSSS signal by 16 samples per chip. Figure 21 shows the block parameters for the Ideal Rectangular pulse filter Tx. The final up sampling process
is done to ensure that the DFT/IDFT operation is performed on an entire PN code sequence. The sample time of the DSSS signal before the pulse filter block is 0.001 seconds, however, the sample time after the rectangular pulse filter block is \( \frac{0.0001 \text{sec}}{16} \) = 0.0000625sec. The DFT/IDFT length is 1024 samples resulting in 0.0000625sec \times 1024 = 0.064 sec, which is one chip duration longer than the complete PN code duration of 0.063 sec.

![Ideal Rectangular Pulse Filter Tx block parameters](image)

**Figure 21:** Ideal Rectangular Pulse Filter Tx block parameters

### 3.3.3 Receiver

The S\(S_{\alpha}A_{\beta}I\)R receiver is depicted in figure 13 with the four subsystems: The Calculated Impulse Response, Delay Locked Loop, PN code Replica and Decision Logic. An overview of these subsystems was given in section 3.2. This section will cover the details within each subsystem and how they are implemented in Simulink.
3.3.3.1 The Calculated Impulse Response

The Calculated Impulse Response subsystem is shown in figure 22. The received signal and PN code are passed through the Fast Fourier Transform (FFT) blocks. The FFT and Inverse FFT are mathematical algorithms that are generally used to compute the Discrete Fourier Transform [7]. Before the FFT can be applied, the data must be windowed [7] [8]. The Window function allows for continuity between the first data point and last data point in the sequence. If there is a discontinuity between these two points, additional frequency components will be added to the spectra of Y(f) and X(f). The output of the FFT series is a complex number. As described in section 3.2 the product of X*(f) complex conjugate and Y(f) determines the transfer function (H(f)). The Complex to magnitude-angle block determines the angle information of the transfer Function.

Figure 22: Calculated Impulse Response subsystem

The transfer function, H(f)’ is created using the calculated angle and a constant value 1 as the magnitude. The Magnitude-Angle to Complex block converts the given magnitude and phase information into a complex number or new transfer function (H(f)’).
The IFFT is then used to determine the impulse response of the channel. This impulse response is used by the next subsystem, the Delay Locked Loop.

3.3.3.2 Delay Locked Loop

The Delay Locked Loop is the most complicated of all the subsystems. The Delay Locked Loop contains the logic used to determine the channel delay, as shown in figure 23. In order to measure the delay innate in the impulse response the magnitude of its components must be examined. The Complex to Magnitude block converts the complex value of \( h(t) \) into the magnitudes. The Peak Finder block determines the largest peak above an established threshold and outputs the delay in samples and magnitude of the peak. Figure 24 shows the Peak Finder block parameters used to control the system. It should be noted that the threshold used in this block is an important characteristic when measuring the bit error performance of the system. The threshold for the system without noise is set to 0.5 as discussed in section 3.2 and figures 15 and 16.

Figure 23: Delay Locked Loop
Figure 24: Peak Finder Block parameters

The PN delay logic subsystem shown in figure 25 uses the Delay found from the peak finder and Valid Sync from the Decision Logic to determine how much reference delay (τ) and measured delay (t’) to apply to the local PN code. The delay in values generated by the Peak Finder block is in units of unit32. In Simulink, some blocks to not support data type unit32, therefore a data type conversion block is used to convert the incoming data to a double precision data type. In addition, the Peak Finder block outputs a single delay measurement at the incoming frame rate, which is 0.064 seconds. A Rectangular Pulse Filter is applied to up sample the delay measurements by 1024 to ensure proper timing with the other subsystems.
The Valid Delay Logic shown in figure 26 determines when the delay is zero or not zero. When the delay measurement is received the Sign block and lookup table converts delay measurements of zero and below to zero while all other delay values are output as one. These values will be used to determine the logic in the Delay Measurement and Reference Delay Measurement subsystems.

The switch block is monitoring the energy per bit in the decision logic and sets the switch to 1 when synchronization has been achieved based on the threshold set in figure 27. If the threshold is not met, the switch will output a zero.
The Delay Measurement subsystem shown in figure 28 describes the logic used to determine the measured delay. A question one might ask is why would there need to be an additional measured delay when the measured delay is already given by the peak finder? To answer this question an illustration of how the measured delay behaves with and without data is given.
Figure 29 shows the measured delay as the PN code is transmitted with and without data. When data is not transmitted the measured delay output is constant with no variation or drop outs. It can be seen from the top graph in figure 29 that drop outs occur when the data transitions from 1 to -1 or vice versa while within the frame. These drop outs will induce false delay measurements. To avoid these false delay measurements a sample and hold block is used to hold the sampled measured delay until another measurement should be taken, figure 28.

![Figure 29](image)

**Figure 29:** Measured delay valid with data (top), measured delay valid without data (bottom)

Two "AND" gate logics are used to tell the subsystem when to make the next sample. It was noticed during BER measurements of the system that, once the system had achieved lock, additional delay measurements where being taken and passed onto the PN Code Replica subsystem. This would cause misalignments and create BER
degradations. To ensure additional delay measurements were not taken by the second "AND" gate, logic 2, was introduced to prevent any additional delay measurement after synchronization is locked. Synchronization locked is represented by a zero and would force the output of the "AND" gate to zero.

While the system is not in a synchronized locked state the measured delay will be decided by the reference delay increment and valid delay of the system. The Sample and Hold block will sample the measured delay from the peak finder only when the delay is a valid delay and after the reference delay has shifted the local PN code. Since the Peak Finder outputs data at a slow rate of 0.064 sec (frame rate) the reference delay increase signal (input 3) must be delayed by a whole frame to allow for the shifted PN code to be measured by the peak finder block. To ensure that the sample and hold is not sampled on a transition an integer delay block is introduced to sample immediately after a transition.

The purpose of the reference delay measurement subsystem, figure 30, is to shift the spike of the impulse response to the desired delay region as discussed in section 3.2.1 and shown in figure 15 by incrementing a reference delay on the locally generated PN code. The incrementing is controlled by a Clock or Count up block. The clock generates a pulse every defined number of samples established in the block parameters. The concept of the clock is to increment the reference delay by 55 samples every time a pulse is generated. If a pulse passes through the Logic 6 "AND" gate the Running Sum block increments the value by 1 and multiplies the incremented value by a constant of 55. The resultant value is the reference delay used to delay the locally generated PN code. However, to stop the clock from propagating through the "AND" gate one of the additional two inputs must equal zero.
The top section or top input to Logic 6 of figure 30 determines when the spike is within the desired delay region. A Relay block is used to control this logic. Figure 31 shows the Relay block parameters. If the measured delay is above 56 the relay will output a value 1 which would propagate through the "OR" gate and allow the "AND" gate to pass the clock pulse through to the running sum. Once the measured delay reaches the delay region of 55 or less and the Delay is valid the "OR" gate will propagate a zero to the "AND" gate and not allow a pulse to pass through to the running sum.

However, as shown previously the delay valid is subject to drop outs and could still propagate a value of 1 to the "AND" gate. To turn the counter block completely off two criteria must be met.
1. PN Synchronization has been achieved.

2. The measured Delay is within the desired region.

To ensure that these two criteria are achieved a "NOR" gate, logic 4 is introduced. The only way for a "NOR" gate to output a value of 1, is for both inputs to equal zero. Thus, a Trigger Locked block and "NOT" gate are used prior to the "NOR" gate. Once PN synchronization is set the Trigger Locked block locks in the value of 1 with the use of a feedback "OR" gate, shown in figure 32. Due to the "NOT" gate, the value is actually locked to zero. This permits the ability of the measured delay to completely turn off the clock once the measured delay reaches the desired region. When both criteria have been meet another Trigger Locked block and "NOT" gate sets the input value of the "AND"
gate to zero and disables the propagation of the clocks pulse. The output of "NOT" gate logic 5 is defined as the synchronized locked state.

![Logic Diagram](image)

Figure 32: Trigger Locked Subsystem

3.3.3.3 PN Code Replica

The reference delay and measured delay from the Delay Locked Loop are then both fed back into the PN code Replica subsystem. This system is depicted in figure 33. As discussed in section 3.2.3 the channel delay is calculated by adding the measured delay ($t'$) and the reference delay ($\tau$). This is implemented by using the Variable Integer Delay block and sequentially delaying the PN code by the reference delay and measured delay. The channel Delayed PN code is then used to despread the received signal. The reference delayed PN code is used to calculate the impulse response.
Figure 33: PN Code Replica

3.3.3.4 Decision Logic

Figure 34 depicts the block diagram of the Decision Logic subsystem. The goal of this subsystem is to determine the despread data sequence. In doing so one of the major problems the receiver has to face is determining when a bit begins and ends. A Data Recovery block\(^2\) that measures the bit transitions and recovers the timing of each bit was used. This process was developed using a sample based system. Therefore, the Unbuffer block is used to convert a frame based sequence into a sample based sequence. The Data Recovery block is susceptible to noise and will take longer to find the bit edges if the signal to noise ratio is low. To help reduce the amount of noise and increase the signal to noise level a low pass filter is used to pass frequencies at or below 10Hz (data rate) through to the Data Recovery Block. Once the Data Recovery block has determined the bit edges, it will accumulate the voltage level within each bit period. The accumulative voltage level will be used by the Data Sample block. The purpose of the Data Sample block is to sample the accumulated voltage every 0.1 seconds (bit duration).

\(^2\)This subsystem was developed by professor James Flynn of California State University, Northridge
The Data Sample block is enabled by a trigger which is designed to sample right before the bit transition and convert the data back into binary 0s and 1s. The Data Recovery block also measures the energy of each bit period. As discussed in section 3.3.3.2 and figure 25 the energy per bit is used to determine if the system has achieved synchronization in the Delay Locked Loop. The performance measurements made on this simulator are presented in section 4.

Figure 34: Decision Logic Subsystem

3.4 Design of an over the Air DSSS Link with Impulse Response based PN Code Acquisition

To demonstrate the application of the $S_5$AB$_3$IR system, the Simulink model previously discussed in section 3.3 will be used in conjunction with the USRP Tx/Rx hardware. The USRP has the ability to interface with Simulink using a TCP/IP connection over Ethernet. The USRP will provide the $S_5$AB$_3$IR system with two important functions.

1. Analog to Digital and Digital to Analog conversion.

2. RF daughterboard for up and down conversion of the baseband DSSS/BPSK signal.
In order for data to be processed correctly between Simulink and the USRP, the sampling rate in and out of the USRP must equal 100 M samples per second [9]. Therefore, the sampling time and frame rate of the S\textsubscript{s}A\textsubscript{B}_oIR simulation model will be altered to interface with the USRP. The USRP treats these samples as complex numbers, which provides In-phase and quadrature phase components of the signal. The In-phase and quadrature phase information will be used in correcting for the phase difference between the transmitter and receiver carrier wave. To correct for the frequency difference between the transmitter and receiver an external timing source will be used to synchronize the USRP's clocks and allow for coherent detection.

### 3.4.1 Transmitter

The transmitter for the S\textsubscript{s}A\textsubscript{B}_oIR system is very similar to that of the Simulink model found in section 3.3.2, figure 19. The only difference between these two systems is the sampling rate of the DSSS signal. Since the USRP requires the sampling rate equal to 100 M samples per second, the sampling rate of 16,000 samples per second will not work because 16,000 samples per second is not a multiple of 100 M samples per second. The transmitter in figure 19 is modified by the Rectangular Pulse Filter Tx block up sampling factor from 16 to 10 up samples. This results in a sample time of 0.00001 seconds or 10,000 samples per second for the DSSS signal. Figure 35 shows the S\textsubscript{s}A\textsubscript{B}_oIR transmitter system which converts the DSSS signal into a complex number and up samples the signal by a factor of 20 to achieve 200,000 samples per second. From there the USRP will do the rest.

Figure 36 shows the USRP parameters used in the S\textsubscript{s}A\textsubscript{B}_oIR system. The carrier frequency is set to 421 MHz and the amplifier gain is set to 24 dB. The Interpolation
value works the same way the Rectangular Pulse Filter does by up sampling the incoming data by 500 samples, which changes the 200,000 samples per second into the required 100 M samples per second. The resulting DSSS/BPSK signal is transmitted using a standard omni-directional antenna attached to the USRP SMA daughterboard connection.

Figure 35: S_{x}AB_{o}IR Transmitter

Figure 36: USRP Transmitter block parameters

### 3.4.2 Receiver

Figure 37 depicts the S_{x}AB_{o}IR receiver system. It should be noted that the receiver implements the use of an enabled subsystem. Since the data between the USRP and Simulink is transferred over TCP/IP, the data is sent in packets and not captured in
real time [1] [9]. The USRP must notify the subsystem when data is present. The enabled subsystem allows for this process by only processing data when data is present.

Figure 37: SsABoIR Receiver

Figure 38 shows the block parameters for the USRP. Due to the decimation factor or down sampling of 500 samples the output sample rate is 200,000 samples per second. In order for the frame rate to match that of the transmitter and the Simulink model, the frame length parameter is set to 12,800 samples per frame. Resulting in a sample time of \( \frac{1}{200,000} \) seconds at 12,800 samples equals 0.064 seconds per frame.
The enabled $S_sAB_oIR$ subsystem is shown in figure 39. The signal is downsampled to the sample rate of 10,000 samples per second. This allows for the DFT/IDFT in section 3.3.3.1 to represent the received signal and local PN code completely. Even though the frame duration is now 0.1024 seconds or 1.6 times longer than the PN code duration the Impulse Response spike will behave similarly to that described in section 3.2.1.

The BPSK Phase Recovery system recovers the phase difference presented by the transmitter and receiver's phase offset. By recovering the phase allows for proper measurement of the energy per bit and enables greater stability in the thresholds chosen in the $S_sAB_oIR$ system. The output of the signal is then propagated to the described $S_sAB_oIR$ Simulink model found in section 3.3.3 or figure 13.
Figure 39: Enabled $S_{AB,IR}$ subsystem
IV. SYSTEM PERFORMANCE

In section 3 the development and explanation of the DSSS/BPSK simulation, S\textsubscript{s}AB\textsubscript{o}IR simulation and S\textsubscript{s}AB\textsubscript{o}IR over the air link were covered. In this section the performance measurements of each system will be explained and compared. When speaking directly about DSSS systems, Bit Error Rate (BER) and acquisition and tracking timing are used to define the performance of the system. Bit Error Rate is the measure of bits received in error divided by the total number of bits received. It is a function of the signal to noise ratio of the channel. The acquisition and tracking timing is the measure of how long it takes to acquire and track the received signal’s PN code with respect to the signal to noise ratio of the channel. For the DSSS/BPSK simulation, explained in section 3.1, no delay was introduced into the channel therefore synchronization was instantaneous. Thus, acquisition and tracking timing will not be discussed in this section for that model.

The S\textsubscript{s}AB\textsubscript{o}IR system’s performance is highly dependent on two parameters manually entered within the system. The first is the Maximum Peak Finder threshold (figures 23 and 24). This is an established threshold within the Impulse Response subsystem that determines the measured delay of the maximum peak. The second parameter is the reference delay incrementing clock. The reference delay clock determines the amount of time the spike (peak) in the impulse response should remain at its position for measurement before the reference delay is incremented. The longer the spike maintains its delay position the better the data recovery block will perform at determining the bit edges or bit transitions and recover the data and determine synchronization.
It was noted during the performance measurements of the SABIR system there were several limitations and constraints involved. In addition to the performance measurements discussed in this section some of the limitations and constraints found in the SABIR system will be explained.

4.1 Bit Error Rate

This section will discuss the BER of each system presented in section 3. The modulation type used for each system is BPSK. BPSK is defined to have a theoretical bit error rate described by equation 8 [4]. Therefore, the optimum BER that can be achieved for each system is defined by equation 8.

\[
BER_{BPSK} = Q \left( \frac{2E_b}{N_0} \right)
\]

4.1.1 DSSS/BPSK simulation

Since the channel in the DSSS/BPSK simulator described in section 3.1 does not introduce a delay, PN synchronization does not need to be performed. The only aspect of the DSSS/BPSK simulation that differentiates it from standard BPSK modulation scheme is the spreading/despreating (stage 1) of the data. Figure 40 depicts the theoretical BER for BPSK and the measured BER for the DSSS/BPSK simulation. It can be seen that the spread/despread of the data does not add to the degradation of the BER when perfect synchronization has been achieved. However, it should be noted if the PN codes are not aligned the BER will degrade significantly.
4.1.2 $S_sAB_oIR$ System Simulation

Since the BER of the $S_sAB_oIR$ system is highly dependent on the Maximum Peak Finder threshold and the Reference Delay Clock, the optimal BER was measured by setting the Reference Delay Clock to generate a pulse every 100,000 samples or 6.25 seconds. Setting the Reference Delay Clock very slow allows for the $S_sAB_oIR$ system to achieve a delay measurement and allow for the Data Recovery block to synchronize to the data rate. The Maximum Peak Finder block threshold is varied depending on the amount of noise is introduced into the system. The relationships between the clock speed and threshold with respect to noise will be discussed in section 4.3.

During the measurements of the BER for the $S_sAB_oIR$ system it was noted that the minimum energy per bit to noise density ratio, $\frac{E_b}{N_0}$, that allowed the Maximum Peak
Finder block to detect the correct delay was 20dB. This difference is due to the fact that the \textit{SsABoIR} system attempts to measure the time difference of the two signals prior to despreading. Thus, the performance improvement due to the processing gain is not seen.

In a typical DSSS receiver, the receiver uses the processing gain to detect the timing difference of the two signals. When the two PN codes correlate, the process gain allows for the receiver to detect synchronization or close to synchronization and obtains a timing measurement. In the case of the \textit{SsABoIR} system, the receiver attempts to measure the timing difference through the use of the impulse response and then uses the measured timing difference to synchronize the PN codes. Hence, the \textit{SsABoIR} system does not utilize the processing gain to establish a timing measurement. Therefore, the signal detection is impacted by 20dB or the amount of processing gain used in the system.

As shown from the DSSS/BPSK simulation, the spreading/despreading of the data does not degrade the BER if the PN codes are synchronized. Consequently, once the \textit{SsABoIR} system detects the time difference and synchronizes the PN codes the BER performance of the \textit{SsABoIR} system will be equivalent to the theoretical BPSK BER. This could not be verified due to the limitations of the large number of bits that would be needed to be collected in order to verify the theoretical BER at an $\frac{E_b}{N_0}$ of 20dB. The calculated theoretical BER for 20dB is $1.0447 \times 10^{-45}$. This means that the number of bits that would need to be collected to achieve only one error is $1 \times 10^{45}$. In order to verify the \textit{SsABoIR} system is performing to the theoretical BPSK BER the simulation was run for 3 hours and collected 499,800 total bits with zero errors. Therefore, it can be assumed that in this preliminary testing the \textit{SsABoIR} system has reached proper synchronization.
4.1.3 SsABoIR System Over The Air Link

It was discovered during this project that a major limitation to the over the air SsABoIR system is the processing power and random access memory of the computer operating Simulink. When measuring the performance of the SsABoIR system, data must be stored in the workspace or logging location for BER calculations. This process created a memory overload and caused Matrix Laboratory (MATLAB) to interrupt and shutdown. The maximum number of bits that could be collected before MATLAB shutdown was around 200 bits. 200 bits is not sufficient to calculate the BER of the system. Therefore, an alternative approached was used to verify the functionality of the SsABoIR system over the air. The transmitter was set up to transmit a repetitive sequence of \([0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1]\). A small portion of data was collected prior to the memory overload and visual inspection was used to determine that the above sequence was received. Figure 41 is a plot of the data collected by the receiver. It can be seen that the SsABoIR system over the air was able to synchronize and decode the transmitted data correctly.

![Figure 41: Data pattern of received bit sequence of the SsABoIR system over the air](image)

50
4.2 Acquisition and Tracking Time

The SsABoIR system, unlike the typical DSSS receiver, can measure the time difference of the two PN codes by processing one frame, while the typical DSSS receiver must attempt several iterations to align the two PN codes. However, due to the wrap around portion of the DFT/IDFT sequence the SsABoIR system must shift the spike of the impulse response to the desired delay region for stable measurements, see section 3.2.1. The shifting characteristic of the SsABoIR system behaves very similarly to that of the iterative process of the typical DSSS correlation method. Shifting the spike of the impulse response to the desired delay region increases the time required to perform synchronization. Because of the shifting process the acquisition and tracking time can be broken down into two cases. Case 1: if the channel delay happens to fall into the desired delay region and Case 2: if the channel delay falls outside the desired delay region. The time required to acquire synchronization for case 1 is just the processing time of one frame, which is 64 ms. Figure 42 shows the minimum amount of time for Case 2 that would be needed by the SsABoIR system to synchronize the two PN codes with respect to $\frac{E_b}{N_0}$. In addition, the maximum amount of time for Case 2 is depicted in figure 43.
Figure 42: Minimum synchronization time with respect to $\frac{E_b}{N_0}$

Figure 43: Maximum synchronization time with respect to $\frac{E_b}{N_0}$

It should be noted from figure 42 and 43 that as $\frac{E_b}{N_0}$ decreases, more time is required by the $S_2$AB$_3$IR system to achieve a valid delay measurement and bit synchronization. Figure 43 is the same plot of figure 42, however, the time is scaled by a
factor of 18. This scaling factor is due to the size of the DFT/IDFT. The DFT/IDFT is represented by a total of 1024 samples or phase measurements. In the $S_{AB,ir}$ system the desired delay region is set to 55 samples and below. Hence, if the channel delay falls at the end of the DFT/IDFT sequence it would take 18 iterations for the impulse response spike to reach the desired delay region. In summary the $S_{AB,ir}$ system ranges from 0.064 seconds to 16.9 seconds to acquire and track the received signals PN code.

4.3 Limitations and Constraints

Throughout this project several observations and discoveries were made about the nature of the $S_{AB,ir}$ system. This section reviews some of the major limitations and constraints that affect the performance of the $S_{AB,ir}$ system.

4.3.1 Impulse Response and Noise

It has been shown that the $S_{AB,ir}$ system cannot acquire the time difference of the two PN codes when the $\frac{E_b}{N_0}$ is below 20dB. One of the limitations of this system is how noise affects the impulse response. Figure 44 shows the difference between an impulse response with and without noise. From the plot on the right ($\frac{E_b}{N_0} = 30dB$) of figure 44 it can be seen that the magnitude of the spike is dramatically reduced and additional phase components are created. When noise is added to the system the frequency components of the received signals PN code begin to become lost or distorted by the noise. This makes it extremely difficult to determine a phase measurement when parts of frequency components within the PN code are no longer recognizable. Only the frequency components with larger magnitude will be distinguishable and able to achieve a phase measurement.
4.3.2 Data Recovery Block and Noise

The Data Recovery block is responsible for determining the bit edges of the received signal in order to evaluate the data. This requires the Data Recovery block to determine the bit timing or bit transitioning. One of the key components in determining the bit timing is the measurement of the zero crossings and synchronizing a Voltage Controlled Oscillator to each transition. However, as noise is introduced to the signal the zero crossings tend to drift and become erratic. Furthermore, additional zero crossings could be introduced creating a longer synchronization time for the Voltage Controlled Oscillator.

In the S\textsubscript{o}AB\textsubscript{o}IR system the Delay Locked Loop subsystem determines when to stop incrementing the reference delay. One of the criteria for stopping the reference delay from incrementing is that the system must achieve synchronization. In order to achieve synchronization the Data Recovery block must determine bit synchronization. In summary, the more noise added to the system the longer the Data Recovery block will
take to determine PN synchronization thus requiring the reference delay clock to slow down. This explains why (figures 42 and 43) the synchronization time increases as the $\frac{E_p}{N_0}$ decreases.
V. CONCLUSION

This project was established to develop and demonstrate an alternative approach to a PN code acquisition and tracking system. Through the use of Simulink and the USRP technology, the S<sub>c</sub>AB<sub>o</sub>IR system was developed and tested, figure 45. A progressive approach was used to verify the functionality of the S<sub>c</sub>AB<sub>o</sub>IR system. First, a basic DSSS/BPSK simulation link was created in Simulink to verify the functionality and performance of a DSSS/BPSK system. Second, the S<sub>c</sub>AB<sub>o</sub>IR simulation system was created in Simulink to verify the ability to acquire and track the received PN code by using the impulse response of the channel. Finally, the S<sub>c</sub>AB<sub>o</sub>IR system over the air link was demonstrated with the implementation of the USRP hardware.

Figure 45: Picture of S<sub>c</sub>AB<sub>o</sub>IR system

Through this project it was discovered that the S<sub>c</sub>AB<sub>o</sub>IR system is able to acquire and track the received signal's PN code. The initial implementation of this system achieves acquisition and track for values of $\frac{E_b}{N_0}$ greater than or equal to 20dB. Once the
system is able to acquire and track the received PN Code it was shown that the BER performs to the theoretical performance of a typical DSSS/BPSK system.

It was noted during this project that further areas of improvement and research could be conducted on the S_sAB_oIR system. Several methods were used in an attempt to reduce the amount of noise in the impulse response, which would enable acquisition for lower signal to noise ratios. This includes using a running average of the impulse response to lower the noise floor and allow the spike to become recognizable. However, additional logic will be needed to ensure proper measurements of the impulse response are taken.

An additional area of improvement would be to make the S_sAB_oIR system completely autonomous. When measuring the performance of the system the reference clock timing and Maximum Peak Finder parameters were manually entered into the system. It would be beneficial if these parameters were dynamically changing based on the noised received by the system. Creating a noise detection system, with additional logic, would allow for the S_sAB_oIR system to become a completely self-governing system.

In conclusion, the S_sAB_oIR system was shown to provide an alternative method for acquiring and tracking the received PN code. Even though the detection threshold for a time difference measurement is high, additional methods could be used to increase the detection performance of the system.
Bibliography


[Accessed October 2012].