NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

DESIGN OF SCALABLE RECEIVERS FOR LOW PROBABILITY OF DETECTION COMMUNICATIONS SYSTEMS

by

Frank R. Cowan IV

September 2009

Thesis Co-Advisors: Frank Kragh
Herschel H. Loomis

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
September 2009

3. REPORT TYPE AND DATES COVERED
Master’s Thesis

4. TITLE AND SUBTITLE Design of Scalable Receivers for Low Probability of Detection Communications Systems

5. FUNDING NUMBERS

6. AUTHOR(S) Frank R. Cowan IV

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

Conventional approaches for a distributed low probability of detection communications system with a large number of unique transmitters and a single or a few receivers, require receiver complexity proportional to the number of transmitters. To improve efficiency in terms of receiver complexity, two alternative designs are analyzed and compared to a reference receiver, whose complexity grows linearly as the number of transmitters increases. The first alternative system groups the transmitters into clusters whose pseudorandom noise codes have some chips in common. The resulting receiver would then perform two stages of processes: identification of the transmitting cluster and received bit detection. The total number of processes required for any given transmitter would be substantially less than the traditional receiver. The second alternative design would utilize a common long spreading code and a shorter cyclically shifted spreading code in each transmitter. The receiver utilizes the cyclic shift property of the fast Fourier transform to recover efficiently both the identity of active receivers and the data sent using a single branch. The complexity of the two proposed systems is compared to that of the reference system.

14. SUBJECT TERMS Direct sequence spread spectrum, binary phase shift keying, Fourier transform, pseudo-random noise, cyclic, low probability of exploitation, low probability of intercept, low probability of detection.

15. NUMBER OF PAGES
83

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

THIS PAGE INTENTIONALLY LEFT BLANK

DESIGN OF SCALABLE RECEIVERS FOR LOW PROBABILITY OF DETECTION COMMUNICATIONS SYSTEMS

Frank R. Cowan IV
Lieutenant Commander, United States Navy
B.S., U.S. Air Force Academy, 1999

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2009

Author: Frank R. Cowan IV

Approved by: Frank E. Kragh
Thesis Advisor

Herschel H. Loomis
Co-Advisor

Jeffrey B. Knorr
Chairman, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Conventional approaches for a distributed low probability of detection communications system with a large number of unique transmitters, and a single or a few receivers, require receiver complexity proportional to the number of transmitters. To improve efficiency in terms of receiver complexity, two alternative designs are analyzed and compared to a reference receiver whose complexity grows linearly as the number of transmitters increases. The first alternative system groups the transmitters into clusters whose pseudorandom noise codes have some chips in common. The resulting receiver would then perform two stages of processes: identification of the transmitting cluster and received bit detection. The total number of processes required for any given transmitter would be substantially less than the traditional receiver. The second alternative design would utilize a common long spreading code and a shorter cyclically shifted spreading code in each transmitter. The receiver utilizes the cyclic shift property of the fast Fourier transform to recover efficiently both the identity of active receivers and the data sent using a single branch. The complexity of the two proposed systems is compared to that of the reference system.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Large distributed networks often have more transmitters than receivers, like the uplink in a single cellular network cell. Such a system can also use direct sequence spread spectrum (DSSS) in order to achieve a low probability of detection (LPD) and multiple access. DSSS systems achieve LPD performance using pseudo-random (PN) codes. Such coding also improves anti jamming (AJ) performance and mitigates interference between transmitters.

All designs considered in this thesis have transmitters that use a unique PN code for LPD, AJ, and multiple access. In order for a receiver to process all signals, it must have memory to store the reference PN code(s), a mixer to despread the signal, a matched filter to extract decision data, a sampler to select discrete data for analysis, and a decision algorithm in hardware or software to make bit decisions. In a traditional receiver, this requirement results in significant growth in complexity as the number of transmitters increases, making large systems unwieldy to field and difficult to scale for increased traffic loads.

One potential solution involves sub-dividing the received signal set into smaller groups using a mask of common PN code points. For example, a system with 32 transmitters requires 32 copies of the receiver processing hardware, or a software radio equivalent, including a demodulation algorithm run 32 times. Dividing the set into four subsets of eight related signals improves the efficiency of processing, as a single signal requires only a portion of the receivers for demodulation. The receiver would first determine which

subset contains the received signal, then attempt to demodulate all signals in that subset. The total number of processes would be 12 vice 32—four initial detection decisions to determine the subset and eight demodulations. This is potentially efficient if the number of active transmitters is a small fraction of the total transmitters at any given moment. However, in the case of all transmitters sending nearly simultaneously, the coding mask receiver is less efficient as 32 demodulations and four detections require 36 processes, while the linear-growth receiver still only requires 32. Additionally, the probability of error in the coding mask system will be worse than for the reference receiver as the mask detection stage introduces additional error.

A second proposed solution involves using short PN codes related to a single code by cyclic shift for multiple access. To provide for code division multiple access (CDMA), each transmitter is preprogrammed with a predetermined shifted form of a base PN code with duration of a single bit. Every transmitter then uses a common longer PN code to provide transmission security by mixing the two codes together prior to mixing with data. The longer PN code has duration greater than a single bit and potentially never repeats within the duration of the entire message thereby ensuring a unique PN code for each transmitter and LPD.

This proposed receiver also removes the common PN code through mixing before filtering and sampling of the combined received signal. It compares the discrete Fourier transform (DFT) processed signal received against a reference DFT of the short unshifted PN code, extracts the transmitter iden-

tity through an inverse DFT (IDFT) and completes bit decisions. This process achieves the same functionality as the matched filter in the reference receiver, but with less components required.

The IDFT output identifies which transmitters sent data, as well what data was sent. Increasing the number of transmitters requires additional processing of the IDFT output, but this growth is smaller as the system needs fewer copies of shift size and bit detection components. Finally, the cyclic PN code system’s improved scalability is not at the expense of error performance. It has equivalent error performance to the reference receiver for binary phase shift keying (BPSK) modulation within an additive white Gaussian noise (AWGN) channel.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

This thesis would not have been possible without the patience and support of my wife, Kaoru. Emigrating from your homeland is always tough, more so when school takes up all of your husband’s time.

I would also like to thank Professor Frank Kragh, for his commitment to this thesis and patience with my endless questions, and Professor Herschel Loomis, for his insight on this paper and all the knowledge I gained from our meetings.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

This thesis examines potential solutions to the problem of scalability for a widely distributed direct sequence spread spectrum (DSSS) communications system. This system must have low probability of detection (LPD) characteristics, which prevent the exploitation of its signals by an interceptor, as well as prevent interference between transmitters. Each transmitter, therefore, must have a unique pseudo-random noise (PN) code, and that in turn requires the receiver to process all possible system PN codes simultaneously.

Adding extra copies of the processing hardware for each additional transmitter quickly leads to an unwieldy system. There will be growing hardware costs and physical space requirements, as well as potential increased delays in the signal processing. Throughout this thesis, the system that requires an additional set of hardware for each additional transmitter will be referred to as the “reference receiver” and its detailed analysis is contained in Chapter III. The reference receiver will provide the error performance benchmark for comparison of scalability solutions. The reference receiver and all solutions to the scalability problem will use DSSS with unique PN codes in each transmitter to maintain LPD and multiple access. Additionally, each analysis will use an additive white Gaussian noise (AWGN) channel with no forward error correction coding.

Low probability of interception (LPI) and low probability of exploitation (LPE) are sometimes used interchangeably with LPD. LPD in the context of this thesis refers to structuring the signal such that it is relatively immune from being detected [1], and this term will be used from now on in discussing preventing the exploitation of the signal.

B. OBJECTIVE

This thesis will examine and validate the reference receiver, as well as analyze two potential solutions to the scalability problem by comparing their error performance and scalability. One solution is the “cyclic PN code receiver.” It uses two PN codes to provide multiple access and identification of transmitters, as well as LPD. A common receiver with a single branch of hardware processes all signals simultaneously using discrete Fourier transforms (DFTs), which act as a matched filter with a single bit decision block.

An alternative solution is the “coding mask receiver.” It uses the same hardware as the reference receiver for bit decisions but sub-divides them into groups with similar PN codes. Each subset then has a detection branch to reduce the total number of processes required for a single received signal. This solution speeds up processing of communications by requiring fewer processes for a single signal received. However, the same physical hardware is required to support the full system, so scalability is still potentially an issue.

C. RELATED WORK

PN codes with different lengths in conjunction with the discrete Fourier transform provide a method to efficiently emulate a set of matched filters and maintain bit error rate performance. The cyclic PN code receiver solution is similar to cyclic code shift keying (CCSK), a technique used in the Joint Tactical Information Distribution System (JTIDS) [2], except that CCSK uses the cyclic shift to indicate a data value while in this thesis, the cyclic shift provides multiple access and identifies the source transmitter at the receiver [3]. The coding mask receiver solution uses unique PN codes with points of similarity to perform the initial signal detection and then separate subgroups of receivers for demodulation. In terms of the signals, this is the same as a typical CDMA [4], but with prescribed correlation between portions of the spreading PN codes.

Additionally, the cyclic PN code receiver’s shorter PN code repeats at the bit rate and provides code division multiple access (CDMA) similar to the methods described by Lee and Miller and separately by Ha as used in the IS-95B cellular standard. There, the long PN code scrambles the data while two short codes, with different shifts, maintain multiple-access within and between cells while minimizing interference [4], [5].

The next chapter provides an overview of the design points of the various receivers investigated in this thesis. Chapter III then provides a detailed analysis and performance simulation of the reference receiver. Chapter IV follows with a detailed description, theoretical calculations, and simulation of the cyclic-PN code receiver, followed in

Chapter V by a similar analysis of the coding mask receiver. Chapter VI provides a summary of all data and compares error rate and scalability performance. Finally, conclusions are provided in Chapter VII.

II. SYSTEM DESIGN

In order to operate as a distributed communications system, a proposed design must meet three requirements. It must maintain low probability of detection (LPD). This means preventing the detection of the signal by an interceptor through the design of the signal's waveform and does not refer to protecting the data through encryption. Next, a proposed design must have error performance similar to uncoded BPSK in an AWGN channel. Superior scalability achieved through an increase in the bit error rate is not acceptable. Finally, each valid solution must have superior scalability to the reference receiver, which is expected to have linear growth as the number of transmitters increases.

A. LOW PROBABILITY OF EXPLOITATION DESIGN

Direct sequence spread spectrum (DSSS) addresses issues of LPD by spreading a signal's spectrum to decrease the probability of detection [1]. Additionally, using multiple pseudo-random noise (PN) codes allows for increased protection as well as improvements in demodulation as each receiver can use unique inner codes.

1. Non-recursive Coding

Effective PN codes must have a random appearance. Each transmitter must therefore use a non-recursive code, i.e., one that does not repeat for the duration of each transmission and ideally never use the same code in subsequent transmissions. Mixing two PN codes achieves this as long as one of the codes is non-recursive [8].

A random number generator, as simulated in MATLAB, can generate an infinite length code for this purpose. In a simulation, a common code generator can provide codes for the transmitter and receiver. In a real system, synchronizing the generators with a common seed achieves the same [9], [10].

2. Low Power

An LPE communications system typically uses less power in its transmitters in order to reduce the chance of detection. However, without forward error correction coding, this will also reduce error performance and range. For the models in this thesis, the energy per bit remains constant while the noise power varies, allowing for the testing of a range of signal to noise ratios.

3. High Bandwidth

Spread spectrum signals require more bandwidth than non-spread signals to transmit at the same data rate [5]. The proposed solutions control mutual interference through PN codes. Future design work should explore forward error control coding to support binary phase shift (BPSK) or quadrature phase shift keying (QPSK). This thesis' simulations used BPSK modulation for simplicity. However, DSSS designs using QPSK have superior LPE performance, since the signal squared does not reveal the carrier frequency [1].

B. SCALABILITY

Examining a system with N transmitters, for a linear-growth receiver to processes the received signals, it must repeat the filtering and bit decisions for all signals N times, resulting in an approximate total number of multiplication processes of N^2. However, the fast Fourier transform (FFT) algorithm, a means to implement the discrete Fourier Transform (DFT), requires $0.5N\log_2 N$ complex multiplication operations [6]. The proposed receiver will have one FFT and one inverse FFT (IFFT) each, requiring a potential total of $N\log_2 N$ complex multiplications, which increases at a rate less than the reference receiver as the system grows in size. A detailed scalability discussion is included Chapter V, section B. The next chapter describes the reference receiver in detail.

THIS PAGE INTENTIONALLY LEFT BLANK

III. REFERENCE RECEIVER

The reference receiver uses a unique PN code for each transmitter. Its receiver has some common components that process all received signals together, but it relies on separate de-spreading, filtering, and bit decision blocks for each transmitter. The reference receiver is representative of a straightforward design approach, and therefore provides a good benchmark for comparisons. Determining the reference receiver's BPSK error performance and validating theoretical calculations with simulation are the basis for comparison and analysis of the two alternative designs. The block diagram for this reference design was based on Nicholson's hypothetical DSSS transmitter with a single PN code and carrier frequency oscillator [1].

A. REFERENCE RECEIVER DESIGN

The reference transmitter and receiver use a single spreading code and upconversion to create the spread spectrum waveform. The receiver complexity grows linearly as the number of transmitters increases, but it provides the benchmark for error performance.

Its probability of bit error is $Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$, where E_b is energy per bit and ${N_0}/{2}$ is the two-sided noise power spectral density, which is identical to a binary phase shift keyed (BPSK) system [7] and derived in equations (3.16) and (3.17). This uncoded BPSK error performance is as expected for a DSSS system because the signaling is antipodal [8].

1. The Reference Transmitter

The reference transmitter initially transforms the polar binary data stream $\{d_i\}$ into an antipodal non-return-to-zero (NRZ) binary signal $m_p(t)$ at the bit rate $R_b = 1/T_b$, where $p_{Tb}(t-iT_b)$ is a pulse function for the i^{th} bit time with duration T_b, and subscript p in $m_p(t)$ is the transmitter number as shown in equation (3.1).

$$m_p(t) = \sum_{-\infty}^{\infty} d_i p_{Tb}(t-iT_b) \tag{3.1}$$

The data mixes with the transmitter's unique pseudorandom noise (PN) code, $c_p(t)$, and carrier wave, $\cos(2\pi f_c t)$. Each transmitter uses the same carrier frequency (f_c), which is much higher than the bit rate. The amplifier applies a preset gain of A before the signal enters the channel as $s(t) = Am_p(t)c_p(t)\cos(2\pi f_c t)$, as shown in Figure 1.

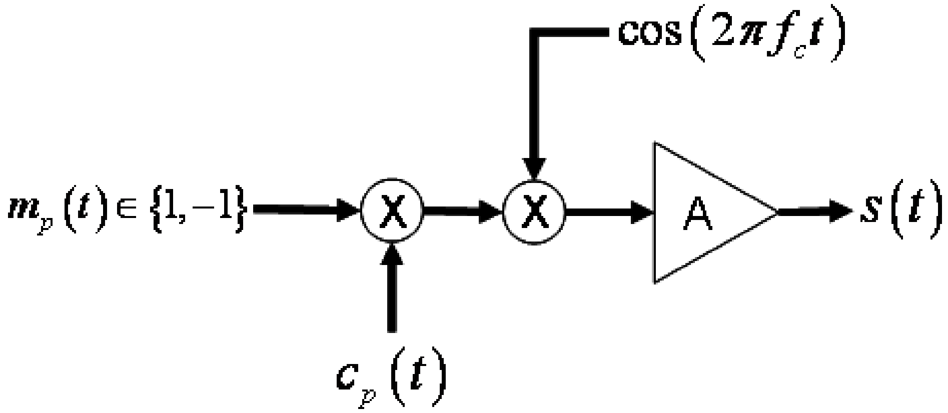

Figure 1. p^{th} Reference Transmitter

2. The Reference Receiver

The channel for this model is assumed to be additive white Gaussian noise (AWGN), adding a white Gaussian noise factor $n(t)$ to the combined signal, resulting in a received signal of $s(t)+n(t)$ as shown in equation (3.2). At the output of the LNA, the signal received still matches equation (3.2) for theoretical calculations.

$$r(t) = Am_p(t)c_p(t)\cos(2\pi f_c t) + n(t) \tag{3.2}$$

One complete receiver branch is required for each of N transmitters, resulting in a system as shown in Figure 2.

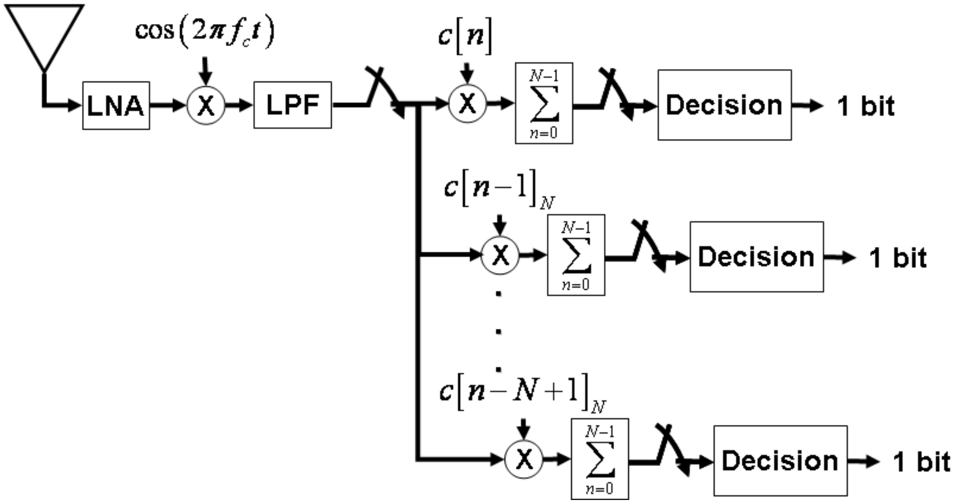

Figure 2. Reference Receiver System

Examining just the signal received from the p^{th} transmitter, the received signal passes through a low-noise amplifier (LNA) before mixing with the carrier and unique PN code matching its transmitter. The signal now has two components, the squared cosine function, and noise.

$$\begin{aligned} r(t) &= Am_p(t)c_p(t)c_p(t)\cos(2\pi f_c t)\cos(2\pi f_c t)+n(t)c_p(t)\cos(2\pi f_c t) \\ &= Am_p(t){c_p}^2(t)\cos^2(2\pi f_c t)+n(t)c_p(t)\cos(2\pi f_c t) \end{aligned} \tag{3.3}$$

Using the trigonometric identity $\cos^2(x)=\frac{1}{2}\left(1-\cos(2x)\right)$ [11], and the squared PN code equating to one, equation (3.3) simplifies, giving a signal with three components: a signal term, a double frequency term, and a noise term.

$$\begin{aligned} a(t) &= \frac{A}{2}m_p(t)\left(1-\cos(4\pi f_c t)\right)+n(t)c_n(t)\cos(2\pi f_c t) \\ &= \frac{A}{2}m_p(t)-\frac{A}{2}m_p(t)\cos(4\pi f_c t)+n(t)c_n(t)\cos(2\pi f_c t) \end{aligned} \tag{3.4}$$

Looking at only the signal term entering the matched filter, its output is

$$y(t)=h(t)*\frac{A}{2}m_p(t)=\frac{A}{2}\int_{-\infty}^{\infty}p_T(\alpha)m_p(t-\alpha)d\alpha\ . \tag{3.5}$$

With substitution, the time interval of the convolution integral changes from $-\infty<\alpha<\infty$ to $0<\alpha<T_b$. Using the new interval, the pulse function term, $p_T(\alpha)$, in the convolution integral equates to one, giving a new form of the integral.

$$y(t)=\frac{A}{2}\sum_{-\infty}^{\infty}d_i\int_0^{T_b}1p_T\left(t-\alpha-iT_b\right)d\alpha \tag{3.6}$$

Evaluating this integral at the sample instant, $t=kT_b$, where $k\in Z$, yields the signal portion of the decision statistic.

$$y(kT_b)=\frac{A}{2}\sum_{-\infty}^{\infty}d_i\int_0^{T_b}p_T\left((k-i)T_b-\alpha\right)d\alpha \tag{3.7}$$

In evaluating the convolution integral with $k,i\in Z$, there are three possible relations: $k-i\geq 2$, $k-i\leq 0$, and $k-i=1$. In the first instance, the integral is zero in $0<\alpha<T_b$. The second condition likewise solves to zero for

the same reason. In the third instance, evaluating the integral at $k-i=1$ results in the final form of the signal of interest at the output of the matched filter. The final value shown in equation (3.8) is the mean for the decision statistic normal random variable after sampling and prior to the bit decision.

$$\begin{aligned} y\left(kT_b\right) &= \frac{A}{2}\sum_{-\infty}^{\infty} d_i \delta_{k-1,i} \int_0^{T_b} p_T\left(T_b-\alpha\right) d\alpha \\ &= \frac{A}{2} d_{k-1} \int_0^{T_b} p_T\left(T_b-\alpha\right) d\alpha \\ &= \frac{A}{2} d_{k-1} T_b. \end{aligned} \qquad (3.8)$$

3. Double Frequency Term

The double frequency term, $\frac{A}{2} m_p\left(t\right)\cos\left(4\pi f_c t\right)$, convolves with the matched filter impulse response, $h\left(t\right)=p_T\left(t\right)$, to yield

$$\begin{aligned} y\left(t\right) &= h\left(t\right) * \frac{A}{2} m\left(t\right) \\ &= \frac{A}{2}\int_{-\infty}^{\infty} p_T\left(\alpha\right) m\left(t-\alpha\right)\cos\left(4\pi f_c\left(t-\alpha\right)\right) d\alpha. \end{aligned} \qquad (3.9)$$

Using the same substitution for the data signal as in the signal of interest above, the interval changes from $-\infty<\alpha<\infty$ to $0<\alpha<T_b$, giving a new convolution integral of the form:

$$y\left(t\right)=\frac{A}{2}\sum_{-\infty}^{\infty} d_i \int_0^{T_b} p_T\left(t-\alpha-iT\right)\cos\left(4\pi f_c\left(t-\alpha\right)\right) d\alpha\ . \qquad (3.10)$$

Evaluated at $t=kT_b$, where $k\in Z$, the integral becomes:

$$y\left(kT_b\right)=\frac{A}{2}\sum_{-\infty}^{\infty} d_i \int_0^{T_b} p_T\left(\left(k-i\right)T_b-\alpha\right)\cos\left(4\pi f_c\left(kT_b-\alpha\right)\right) d\alpha\ . \qquad (3.11)$$

Similar to the signal of interest, the double frequency convolution integral only produces non-zero output when $k-i=1$, resulting in the final form shown in equation (3.12).

$$\begin{aligned} y\left(kT_b\right) &= \frac{A}{2} d_{k-1} \int_0^{T_b} p_T\left(T_b-\alpha\right)\cos\left(4\pi f_c\left(kT_b-\alpha\right)\right)d\alpha \\ &= \frac{A}{2} d_{k-1} \int_0^{T_b} \cos\left(4\pi f_c\left(kT_b-\alpha\right)\right)d\alpha \qquad (3.12) \\ &= \frac{Ad_{k-1}}{8\pi f_c}\left[\sin\left(4\pi f_c kT_b\right)-\sin\left(4\pi f_c\left(\left(k-1\right)T_b\right)\right)\right] \end{aligned}$$

The term in square brackets in the solution has a max value of +2 and minimum of -2. Therefore, the absolute value of the solution must be:

$$\left|\frac{Ad_{k-1}}{8\pi f_c}\left[\sin\left(4\pi f_c kT_b\right)-\sin\left(4\pi f_c\left(\left(k-1\right)T_b\right)\right)\right]\right| \le \frac{2Ad_{k-1}}{8\pi f_c}\,. \qquad (3.13)$$

Accounting for $f_c >> 1$ and its location in the denominator, all possible results of this term are small compared to the possible results of the data signal in equation (3.8), so the double frequency term is ignored as insignificant in further analysis of the bit decision statistic.

4. Noise Calculations

The noise term enters the match filter as $n(t)c_n(t)\cos(2\pi f_c t)$. However, analyzing the resulting power of the noise is more useful. The power spectral density (PSD) of noise from the AWGN channel at the antenna is $S_{nn}\left(f\right)=\frac{N_0}{2}$. Mixing with the PN code has no effect on the resulting PSD. The instantaneous power of a cosine function is $\cos^2\left(x\right)$, which has an averaged of $\frac{1}{2}$. Mixing the noise with the carrier wave results in a noise process with a PSD with half the power density or $S_{nn}(f)=\frac{N_0}{4}$, which is the PSD at the input of the matched filter. The output of the matched filter has power spectral density

$$S_{out}(f)=S_{nn}(f)\left|H(f)\right|^2=\frac{N_0}{4}\left|H(f)\right|^2. \qquad (3.14)$$

To integrate equation (3.14), Parseval's theorem [12] allows the time-domain impulse response function, $h(t)=p_T(t)$, to be used instead. Therefore, the power of the noise at the output of the matched filter is

$$\begin{aligned}\sigma^2&=\int_{-\infty}^{\infty}\frac{N_0}{4}\left|H(f)\right|^2 df\\&=\frac{N_0}{4}\int_{-\infty}^{\infty}\left|h(t)\right|^2 dt\\&=\frac{N_0}{4}\int_{-\infty}^{\infty}\left|p_T(t)\right|^2 dt\\&=\frac{N_0}{4}T_b\end{aligned} \qquad (3.15)$$

Since $p_T(t)=1$ for $0<t<T_{bs}$ and has zero value otherwise, equation (3.15) resolves to the noise power at matched filter output.

5. Reference Receiver Error Performance

Using the output of the matched filter, the signal of interest and noise enter the decision block where the receiver makes bit determinations. Using the signal of interest input, $y(kT_b)=\frac{A}{2}d_{k-1}T_b$, the decision statistic is a Gaussian random variable, $N\left(\frac{Ad_{k-1}T_b}{2},\frac{N_0T_b}{4}\right)$[1]. This is used to determine the probability that the decision block output bit, $\hat{d}_{k-1}$, is in error when the sent bit is $d_{k-1}=-1$. Dividing by the standard deviation of the decision statistic, $\sqrt{\frac{N_0T_b}{4}}$, one finds the area under the right-hand tail of the Gaussian distribution, resulting in

$$\begin{aligned}\Pr\left(\hat{d}_{k-1}=1\,|\,d_{k-1}=-1\right)&=\Pr\left(y\left(kT_b\right)>0\right)\\&=Q\left(\frac{A\sqrt{T_b}}{\sqrt{N_0}}\right)\end{aligned} \tag{3.16}$$

Using the equivalence $A^2T_b=2E_b$, where E_b is the energy per bit, the resulting probability of bit error is identical to BPSK, as expected since the signal is antipodal [7].

$$\begin{aligned}\Pr\left(d_{dec}=1\,|\,d_{k-1}=-1\right)&=Q\left(\frac{\sqrt{A^2T_b}}{\sqrt{N_0}}\right)\\&=Q\left(\sqrt{\frac{2E_b}{N_0}}\right)\end{aligned} \tag{3.17}$$

The theoretical error performance for an antipodal system plotted using a range of signal to noise (SNR) ratios or $\frac{E_b}{N_0}$ (bit energy to noise power spectral density) ranging from 0 dB to 10 dB is shown in Figure 3. This curve represents the theoretical best error performance of the reference receiver with uncoded BPSK modulation and was generated using MathCAD (v.14.0.0.163).

[1] The notation $N\left(\bar{X},\sigma^2\right)$ is used in this thesis to describe a random variable as a normal, i.e., Gaussian, random variable with mean $\bar{X}$ and variance σ^2.

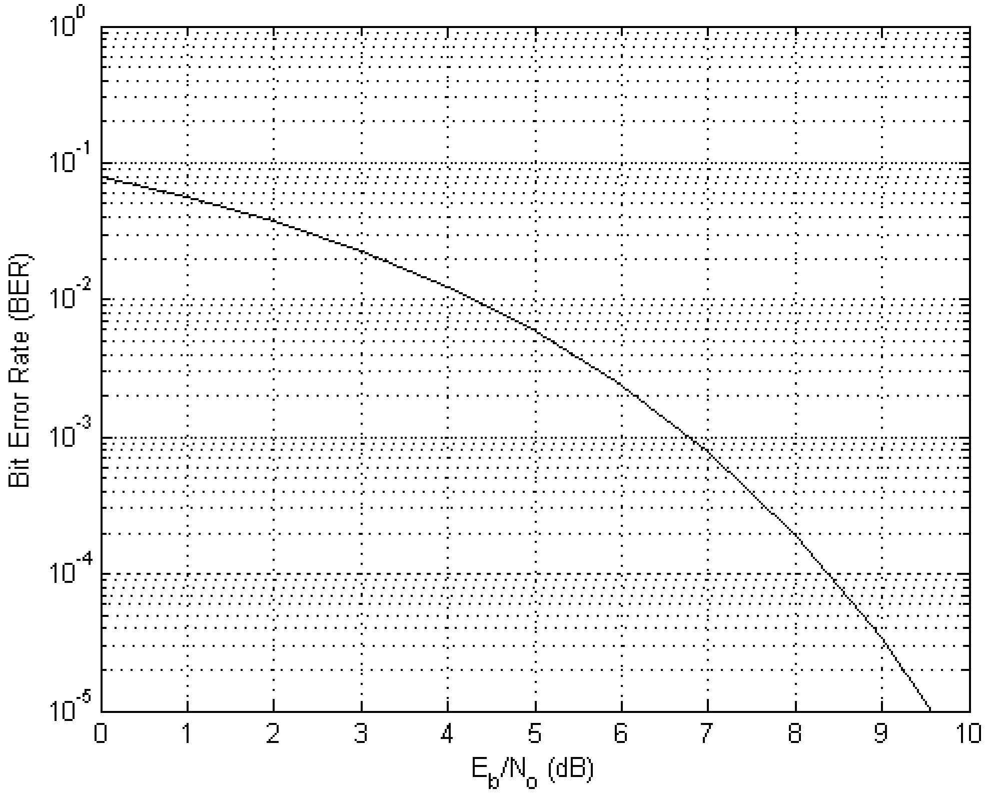

Figure 3. Optimum Bit Error Rate for a BPSK System over a Range of $\frac{E_b}{N_0}$

B. MODELING THE REFERENCE RECEIVER

The reference receiver was simulated with MATLAB and Simulink. The MATLAB file, shown in Appendix A, used the variable “sim_length” to control number of bits simulate. By setting this variable to three, the simulation ran for 100,000 bits for each SNR level over the range of 0 dB to 9 dB. The AWGN channel’s calculations were independent for each bit.

1. Simulation Results

The reference receiver’s simulation results are shown in Figure 4. The black line represents the BPSK theoretical bit error rate performance, and is the same curve from Figure 3. The simulation bit error rate results are the red circles on the plot, representing the bit error rate (BER) for each SNR level. The calculated error rate was the total number of errors divided by the total number of bits sent. The results of the simulation matched the general trend of the reference curve, indicating an accurate model.

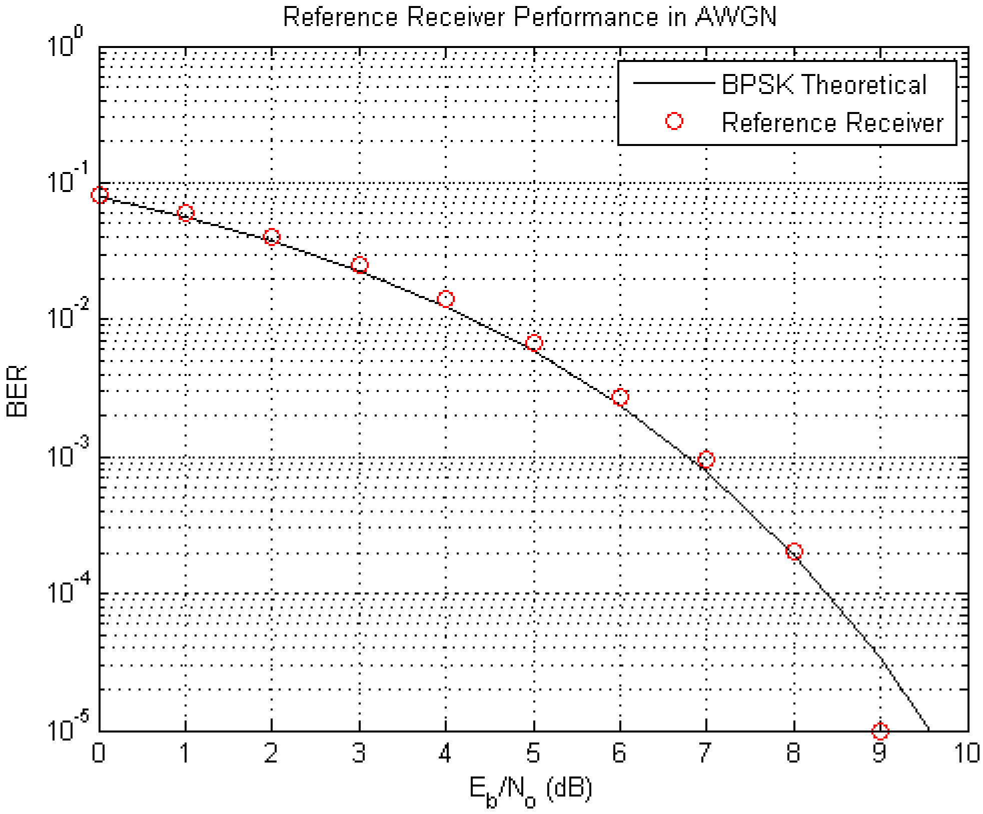

Figure 4. Reference Receiver Performance in AWGN Over a Range of SNR Values and Compared to BPSK

2. Transmitter Simulation Design

The reference transmitter simulation generated a single user’s Bernoulli binary random data stream at the bit rate and a 15-stage shift register maximal length PN sequence at the chip rate. The simulation mapped unipolar binary outputs to antipodal binary before mixing. The resulting chipped signal mixed with the carrier wave, a cosine function block with a carrier frequency of 1.2288 MHz. The resulting signal then passed through a fixed gain before entering the AWGN channel block. The simulation’s parameters are listed in Table 1 and are based on IS-95 parameters [5].

Table 1. List of Reference Transmitter Simulation Parameters

Parameter	Set Value
Chip Frequency R_c	1.2288 MHZ
Bit Frequency R_b	19.2 KHz
Carrier Frequency f_c	4.915200 MHz
Sample Time T_S	1.27E-08 seconds
User Gain A	1 (no units)

Simulink’s AWGN block provided the AWGN channel simulation using the parameters shown in Table 2, which the MATLAB file set prior to starting Simulink.

Table 2. List of AWGN Channel Simulation Parameters

Parameter	Set Value
Initial Seed	45 (default setting)
Mode	Signal to noise ratio
E_b / N_0 (dB)	0 to 9 dB (set by MATLAB file)
Number of bits per symbol	1
Symbol Period	$1/R_b$

The original data goes to the workspace for use in the receiver and error calculations. The combined signal then mixes with a cosine wave function with a carrier frequency that is four times the chip rate and sampling frequency that is 64 times the chip frequency. The user gain is unitary and has no effect on the model with the parameters as set. Integer relations between the chip and bit frequencies were arbitrary, but we require $f_c > R_c$ and $R_s >> f_c$.

3. Reference Receiver Model

The receiver shown in Figure 5 simulates the reference receiver in Figure 2. It consists of a downconverter and filter subsystem, a baseband processor, and an integration block that compares the demodulated signal to a threshold to determine individual bits. The error rate calculation block compares the threshold output against the original data generated by the Bernoulli binary sequence block in the transmitter. However, this visual output is for the user and a separate sink saves each bit to the workspace in a single vector for later error calculations and plotting.

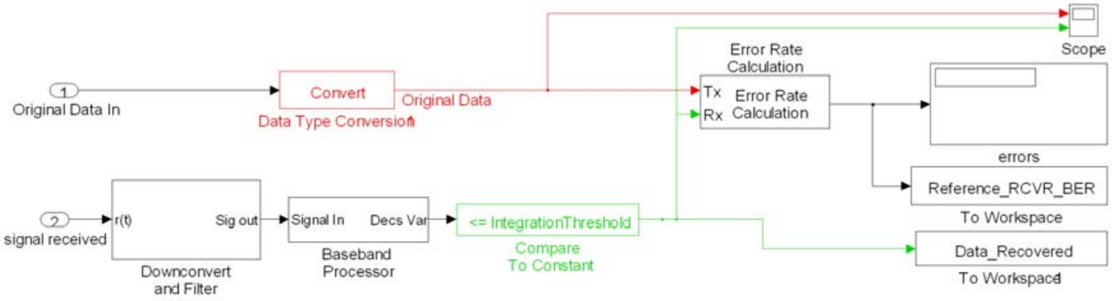

Figure 5. SIMULINK Reference Receiver Model

The downcoverter and filter subsystem mixes the signal with a cosine wave with the same parameters as in the transmitter. Next, the signal passes through a discrete-time low-pass filter (LPF) using a unitary pulse that is 64 samples in length to shape the output. This simulates the effect of the matched filter and eliminates the double frequency. Finally, the receiver downsamples the output with downsampling factor of 64 before proceeding to the baseband processor, which simulates the reference receiver’s sampler.

The baseband processor models the de-spreading, matched filter, and sampler from Figure 2, and mixes the received signal with a PN sequence that matches the transmitter. The resulting signal passes through a matched filter of 64 ones and downsamples 64 times. The processed signal output then goes to the decision block to determine bits. The integration block performs the threshold comparison and decision functions producing a unipolar binary signal output, and sends the results to the workspace for comparison. The receiver simulation matched the theoretical performance, so it provides a point of reference for performance comparison with the cyclic-PN code receiver in the next chapter.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. CYCLIC PSEUDO-NOISE CODE RECEIVER

The cyclic PN code receiver solves the scalability problem while maintaining the BPSK error performance of the reference receiver. Its use of DFTs implemented through the FFT algorithm eliminates the need for multiple matched filters to extract each transmitter’s data. Rather, a single receiver block performs all the same functions for all the transmitters except for bit decisions.

A. CYCLIC RECEIVER DESIGN

The cyclic PN-code design uses two spreading codes, with each transmitter using a cyclically rotated version of the same short code mixed with a longer common PN code. The shifted code repeats for each bit period and allows the receiver to identify the transmitter and demodulate data. When mixed with the long PN code, whose duration is much greater than a single bit, the resulting pseudo-noise sequence is unique and improves the LPE properties of the system. The DFTs allow a single receiver to analyze all transmitted signals simultaneously and determines the number of shifts of the short PN code in each signal as well as the data content. From this, the receiver determines transmitter identity and makes bit decisions. The theoretical bit error rate performance of this receiver is the same as the reference receiver, $P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$.

1. Cyclic PN-Code Transmitter

The cyclically shifted PN code transmitter mixes the long PN code, $b(t)$, with a cyclically shifted shorter code, $c(t,\gamma(p))$, where t is time, γ is the length of the shift, and p is the transmitter number. The transmitter number and quantity of shift are related by an integer factor k, resulting in a relationship of $\gamma = pk$. For this analysis, $k=1$, so the shift number will be the same as the transmitter's number, $\gamma = p$. The shifts in $c(t,\gamma(p))$ are not calculated by each transmitter but are preprogrammed prior to system emplacement. The block diagram for the transmitter is shown in Figure 6.

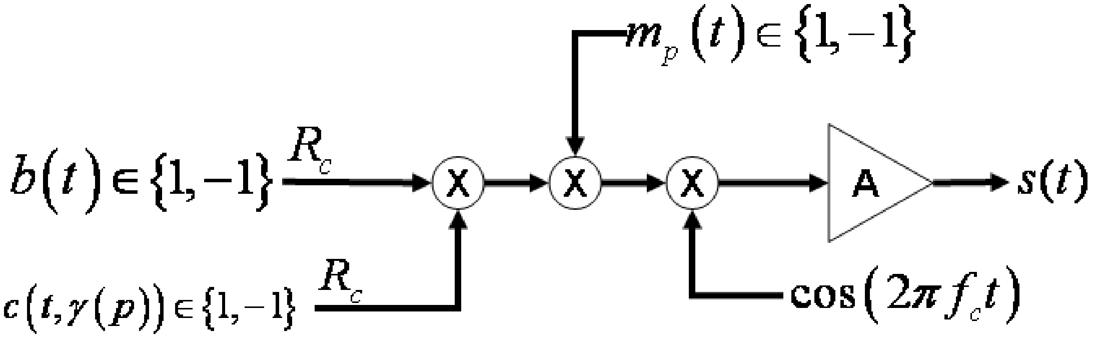

Figure 6. p^{th}Cyclic PN Code Transmitter

The cyclically shifted PN code repeats within each bit, allowing for demodulation and identification. It can be represented as $c(t,\gamma(p)) = \sum_{i=0}^{N-1} c[i-\gamma(p)]_N \, p_{Tc}(t-iT_c)$, where p is the transmitter number, t is time, $\gamma(p)$ is the number of shifts of the reference PN code for transmitter p, and $p_{Tc}(t)=1$ if $0 \le t \le T_c$ and $p_{Tc}(t)=0$ otherwise. The sequence $c[i-\gamma]_N$ is obtained by cyclically shifting the N chip sequence, $c[i]$, γ shifts to the right. Each $c[i]\in\{-1,1\}$ and p does not have to equal γ.

The longer PN code, $b(t)$, provides security by not repeating for the duration of the message. The two PN codes are generated and mixed at the chip rate R_c. Using two codes creates a unique spreading sequence for each transmitter.

The resulting unique PN code then mixes with the antipodal data signal m_p before upconversion to a BPSK transmission at a predetermined carrier frequency f_c. The antipodal data signal can be represented as $m_p(t)=\sum_{-\infty}^{\infty} d_i p_{T_b}(t-iT_b)$, where d_i is the i^{th} bit and $p_{Tb}(t-iT_b)$ is the pulse function for the i^{th} bit time. Finally, the transmitter amplifies the signal by a pre-determined factor A before transmission, yielding the transmitted signal

$$s(t)=Ab(t)c(t,\gamma)\cos(2\pi f_c t)\,. \qquad (4.1)$$

2. Cyclic PN Code Receiver

The signal in equation (4.1) travels through an additive white Gaussian noise (AWGN) channel and arrives at the receiver depicted in Figure 7 with the form shown in equation (4.2).

$$r(t)=A\sum_{-\infty}^{\infty} d_i c(t,\gamma)b(t)\cos(2\pi f_c t)+n(t)\,. \qquad (4.2)$$

The signal enters the downcoverter block and changes to the form shown in equation (4.3).

$$a(t) = A\sum_{-\infty}^{\infty} d_i p_T(t - iT_b) c_p(t,\gamma) b(t)\cos^2(2\pi f_c t) + n(t)\cos(2\pi f_c t) \qquad (4.3)$$

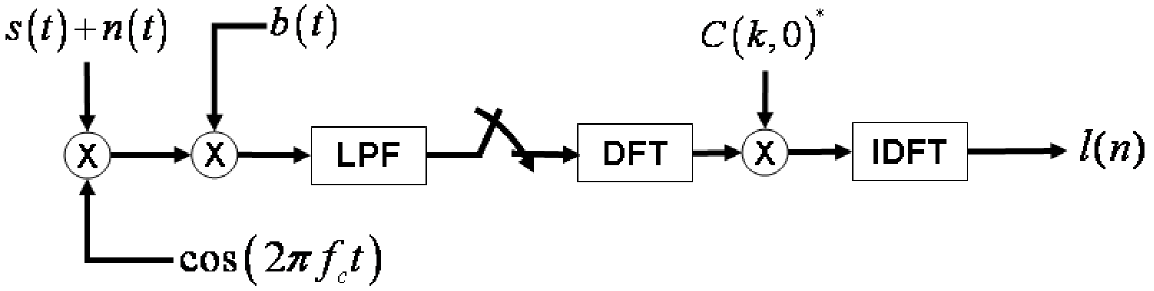

Figure 7. Cyclic PN-Code Receiver

Based on the trigonometric identity, $\cos^2(x) = \frac{1}{2}(1-\cos(2x))$ [11], this changes to the form shown in equation (4.4), with the signal received now in three parts – the desired signal term, the double frequency term, and the noise term.

$$a(t) = \frac{A}{2}\sum_{-\infty}^{\infty} d_i p_T(t - iT_b) c(t,\gamma) b(t) \quad + \frac{A}{2}\sum_{-\infty}^{\infty} d_i p_T(t - iT_b) c(t,\gamma) b(t)\cos(4\pi f_c t) + n(t)\cos(2\pi f_c t) \qquad (4.4)$$

Next, examining the desired signal term allows derivation of the mean of the decision statistic. Noise will be examined separately in the next subsection. Mixing with the long PN code effectively cancels the factor $b(t)$ from the first two terms on the right hand side of equation (4.4). The LPF then removes the double frequency term, giving the output

$$f(t) = \frac{A}{2}\sum_{i=-\infty}^{+\infty} d_i \sum_{j=0}^{N-1} c_{j,p}\Lambda\left[\left(t-(j+1)T_c\right)/T_c\right] \qquad (4.5)$$

where $\Lambda(x) = 1 - |x|$ if $|x| \le 1$ and $\Lambda(x) = 0$ otherwise.

The receiver then samples the signal $f(t)$ at the chip rate prior to it entering the DFT in order to analyze a single bit.

$$g[n]=\frac{A}{2}d_i c[n-\gamma]_N \tag{4.6}$$

Analysis of the signal-only term leads to the mean of the decision statistic. Starting from the output of the inverse DFT (IDFT), the output of the receiver is shown in equation (4.7). $C(k,0)^*$ represents the complex conjugate of the DFT of the short PN code with no shifts. Pre-calculating this result and storing it in memory rather than performing a real-time DFT and complex multiplication helps improve scalability.

$$l[n]=IDFT\left[\left\{C(k,0)^*\right\}DFT\left(g[n]\right)\right] \tag{4.7}$$

Multiplying the two DFT outputs is equivalent to circular convolution in the time domain [12]. The operator $\otimes$ indicates a circular convolution.

$$H(k)X(k)=DFT\left\{h[n]\otimes x[n]\right\} \tag{4.8}$$

The complex conjugate, indicated by *, in the frequency domain is equivalent to the time-reversed signal in the time domain [12]. In this instance, the complex conjugate for the discrete time domain signal in reverse is real, so the complex notation drops out.

$$\begin{gathered} x[n]\leftrightarrow X[k] \\ x[-n]\leftrightarrow X^*[k] \end{gathered} \tag{4.9}$$

A circular convolution is the same as an ordinary discrete time convolution, except that the time-shifted signal is circularly shifted, vice an ordinary shift [12].

$$h[n] \otimes x[n] = \sum_{m=0}^{N-1} h[m] x[n-m]_N \tag{4.10}$$

Applying equations (4.8), (4.9), and (4.10) to equation (4.7) yields an IDFT output of $l[n] = \frac{Ad_i N \delta_{n,\gamma}}{2}$ as shown in equation (4.11). This holds for $0 \le n \le N-1$, where $\delta_{n,\gamma}$ is the Kronecker delta and it is assumed that $c[n]$ has low correlation with $c[n-i]_N$ for all $0 \le i \le N-1$. The square of a shifted PN codes is one and the sum of that result over N-terms is N.

$$\begin{aligned} l[n] &= IDFT\left[\left\{C(k,0)^*\right\} DFT(g[n])\right] \\ &= g[n] \otimes c[-n] \\ &= \frac{Ad_i}{2} c[n-\gamma]_N \otimes c[-n] \\ &= \frac{Ad_i}{2} \sum_{m=0}^{N-1} c[m-\gamma]_N \, c[m-n]_N \\ &\approx \frac{Ad_i N \delta_{n,\gamma}}{2} \end{aligned} \tag{4.11}$$

When $n \ne \gamma$, using a circular convolution definition with a Kronecker delta gives a result of $l(n \ne \gamma) \approx 0$. Choosing $\sum_{m=0}^{N-1} c[m-\gamma]_N \, c[m-n]_N \approx N\delta_{\gamma,n}$ allows the result shown in equation (4.11). In the case of multiple transmitters' signals arriving simultaneously, $l[n]$ indicates whether transmitter n is transmitting (if $|l[n]|$ is large) and the value of the sent data bit ($\operatorname{sgn}(l[n])$). Thus $l[n]$ for $0 \le n \le N-1$ simultaneously indicates which of the N transmitters are active and the bits they have transmitted.

3. Noise Calculations

Noise at $r(t)$ is additive, white and Gaussian and has $\mu = 0$, $\sigma^2 = +\infty$, and a PSD of $\frac{N_0}{2}$. The downconversion changes the PSD to $\frac{N_0}{4}$ due to the average power of the squared cosine function, which is $1/2$.

After the LPF, the variance of the noise is no longer infinite, but limited by the bandwidth of the filter.

$$\begin{aligned}\sigma^2 &= \int_{-\infty}^{+\infty} \frac{N_0}{4} \left| H(f) \right|^2 df \\ &= \frac{N_0}{4} \int_{-\infty}^{+\infty} \left| h(t) \right|^2 dt \\ &= \frac{N_0 {R_c}^2 T_c}{4} \\ &= \frac{N_0 R_b N}{4}\end{aligned} \tag{4.12}$$

where $H(f)$ is the frequency response of the LPF and sampling does not affect white Gaussian noise. Therefore, the noise at $g[n]$, the input to the DFT, is still a Gaussian random variable with zero mean and $\sigma^2 = \frac{N_0 R_b N}{4}$, or $N\left(0, \frac{N_0 R_b N}{4}\right)$. The samples, $g[n]$, are independent because the input to the LPF is white noise, the LPF's impulse response has a duration equal to the chip duration, and the LPF's sample rate is equal to the chip rate. Therefore:

$$E\{g[m] g[p]\} = \frac{N_0 R_b N \delta_{m,p}}{4} . \tag{4.13}$$

The output of the mixer in the frequency domain is:

$$L(k) = G(k)C(k\,|\,0)^{*}\,. \tag{4.14}$$

The expected output of the receiver is then the IDFT of $L(k)$. The product of the two functions in the frequency domain is equivalent to a circular convolution in the time domain [12].

$$\begin{aligned} l[n] &= IDFT\left\{G(k)C(k\,|\,0)^{*}\right\} \\ &= g[n]\otimes c[-n] \\ &= \sum_{m=0}^{N-1} g[m]c[m-n]_N \end{aligned} \tag{4.15}$$

The expected value or mean of the noise component of this function is:

$$\begin{aligned} E\{l[n]\} &= E\left\{\sum_{m=0}^{N-1} g[m]c[m-n]_N\right\} \\ &= \sum_{m=0}^{N-1} E\{g[m]\}c[m-n]_N \\ &= 0 \end{aligned} \tag{4.16}$$

The variance of this function is the variance of the receiver output and therefore the variance of the signal (which allows probability of error calculations),

$$\begin{aligned} \sigma^2 &= E\left\{\sum_{m=0}^{N-1} g(m)c(m-n)_N \sum_{p=0}^{N-1} g(p)c(p-n)_N\right\} \\ &= \sum_{m=0}^{N-1}\sum_{p=0}^{N-1} E\{g(m)g(p)\}c(m-n)_N\, c(p-n)_N \\ &= \sum_{m=0}^{N-1} E\left\{g(m)^2\right\}\left[c(m-n)_N\right]^2 \\ &= \sum_{m=0}^{N-1} \frac{N_0 R_b N}{4} 1 \\ &= \frac{N_0 R_b N^2}{4}. \end{aligned} \tag{4.17}$$

4. Theoretical Error Performance

At the output of the IDFT block, the signal's decision statistic is $l[n]\sim N\left(\frac{Ad_iN}{2},\frac{N_0R_bN^2}{4}\right)$. This is used to find the probability of bit error, $Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$, the same performance as a BPSK system. Using the mean and variance calculated above in equations (4.11) and (4.17), and the equivalence $E_b=0.5A^2T_b$, the bit error rate is

$$\begin{aligned}P_b&=\Pr\left(l(n)>0\,|\,d_i=-1\right)\\&=\Pr\left(\frac{l(n)+\frac{AN}{2}}{\sqrt{\frac{N_0R_bN^2}{4}}}>\frac{\frac{AN}{2}}{\sqrt{\frac{N_0R_bN^2}{4}}}\right)\\&=Q\left(\sqrt{\frac{2E_b}{N_0}}\right).\end{aligned}\tag{4.18}$$

B. MODELING THE CYCLIC PN-CODE RECEIVER

The proposed receiver was simulated in MATLAB and tested for 100,000 bits at each of a range of signal to noise ratios $\left(E_b/N_0\right)$ from 0 dB to 9 dB. A single transmitter was simulated with a one chip cyclic shift ($\gamma=1$) in the base short PN-code. The simulation used an AWGN channel and BPSK modulation for the carrier. Perfect phase synchronization was assumed. The MATLAB code for the simulation is in Appendix B.

1. Simulation Results

The cyclic PN code receiver simulation results are shown in Figure 8. The black line represents the BPSK theoretical bit error rate, and is the same curve from Figure 3. The simulation bit error rate results are the blue squares on the plot, representing the BER for each SNR level. As with the reference receiver, the calculated error rate was the total number of errors divided by the total number of bits sent. The results of the simulation matched the general trend of the analytically derived curve, indicating that model's accuracy.

Figure 9 is an example of the IDFT output for a single bit from a single transmitter with a single shift of the reference short PN code, i.e., $\gamma = 1$. The signal from the transmitter of interest stands out clearly from the 64 possible signals received during this bit time.

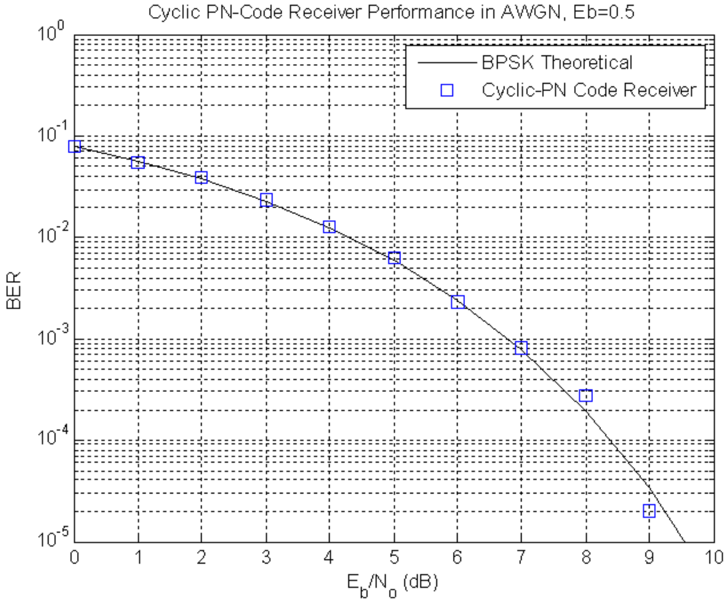

Figure 8. Cyclic PN Code Receiver Performance in AWGN over a Range of SNR

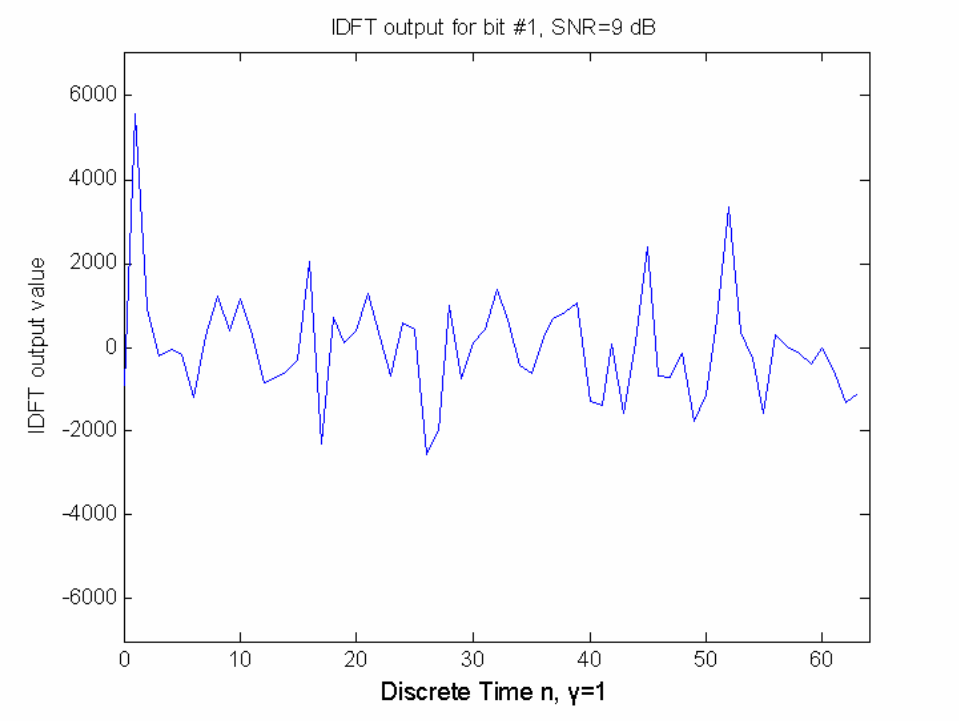

Figure 9. IDFT output for a single bit duration, at $E_b/N_0 = 9\text{ dB}$ with $\gamma = 1$

Figure 8 shows good concurrence between simulation and theory. Figure 9 shows that $l[n]$ at the receiver indicates reception of a +1 from transmitter number one and only noise from the other transmitters. A mechanism is still needed to distinguish an active transmitter's signal from noise only. This is not addressed in this thesis except to note that conventional methods such as using start and stop bit patterns or employing two thresholds to make "+1", "not transmitted" and "-1" can be used to address this issue [13].

2. Transmitter Simulation Design

The simulation of the cyclic PN code transmitter used a fixed $R_b = 1$ and then generated other variables from its multiples. Unlike the reference receiver, these differences were relative and not on an absolute scale. The simulated transmitter mixed the fixed data sequence with two separate PN codes, one of which was the cyclic rotation of the short PN code. The resulting signal was sampled at the chip rate $R_c = 64R_b$ and then upsampled with upsampling factor of 256 before mixing with the cosine vector at frequency, $f_c = 16R_c = 1024R_b$, and a fixed amplitude. The transmitter parameters are in 0.

Table 3. Parameters for Cyclic PN Code Transmitter Simulation

Parameter	Value
R_b (bit rate)	1
R_c (chip rate)	$64R_b$
f_c (carrier frequency)	$16R_c = 1024R_b$
R_s (modulated signal sample rate)	$16f_c$
T_s (sample time)	$1/R_s$
Data bits	50,000

The cyclic PN code simulation did not use the AWGN block in SIMULINK. Rather it generated noise as a vector of the same length as a single upsampled bit based on the SNR for a particular loop and the noise PSD using

$\text{Noise}=\sqrt{\frac{R_s N_o}{2}}\,\text{randn}\left(1,R_s\right)$. The noise PSD was then based on the energy per bit by $N_o=\frac{E_b}{10^{SNR/10}}$, with E_b fixed at 0.5. The MATLAB function “randn” generated a vector of random numbers with length equal to the sampling rate with a coefficient based on the noise PSD, allowing variable noise based on SNR and vector addition. The effect on the transmitted signal was equal to the effect of AWGN.

3. Cyclic PN Code Receiver Model

The cyclic PN code simulation downconverted the received signal with noise by multiplying the data vector by the same cosine vector used in the transmitter. To despread the signal with the long PN code, the simulation upsampled the PN code before mixing. A square pulse whose length was the chip duration performed the low pass filter function through convolution with the de-mixed signal. The simulation sampled the LPF output at the chipping rate, by downsampling using the ratio between sampling and chipping rates.

MATLAB’s Fastest Fourier Transform in the West (FFTW) algorithm, the “fft” function, acted as the DFT block proposed in the theoretical model. The simulation directly calculated the DFTs of the signal received and reference short PN code, the complex conjugate of the short PN code, and mixed the DFT outputs. The MATLAB function “ifft” acted as the IDFT block, and a loop made bit decisions based on the IDFT output.

The cyclic-PN code receiver simulation performed well and matched the theoretical BPSK error rate. In addition to the cyclic-PN code receiver, a coding mask receiver was proposed as a possible alternative. The next chapter analyzes the coding mask receiver and compares its theoretical error performance to the reference receiver.

V. CODING MASK RECEIVER

As an additional point of comparison, one possible alternative to the cyclic PN code receiver is a coding mask receiver. This design relies on similarities in the transmitters’ PN codes to perform a two-stage detection and demodulation. This receiver is not a viable solution as a system if many transmitters are expected to be simultaneously active as it does not improve scalability in this case and it has a greater error rate than the reference receiver. However, it requires fewer processes for single transmitters and could be a viable alternative design.

A. RECEIVER DESIGN

The coding mask receiver subdivides the full body of transmitters into groups, which will be called subsets for the remainder of this thesis. Each subset has related PN codes with some common chips to speed-up processing and this is the “mask.” The receiver’s signal detection circuit mixes the combined signal received with the base PN code of its subset and then its mask to determine signal presence. When a signal from a particular subset is detected, all receivers within that subset process in parallel to extract the signal received. This design has non-linear growth and while it is of similar complexity, efficiency is gained through savings in total number of processes. This savings would be most beneficial in a software defined radio implementation since the savings in processing is realized by simpler and faster algorithms and reduced requirements on the microprocessor, digital signal processor, or field programmable gate array (FPGA) responsible for the signal processing [14].

The coding mask receiver system would use the same transmitter as the reference receiver as shown in Figure 1 and generate a similar transmitted signal in the form $s(t) = Am_p(t)c_p(t)\cos(2\pi f_c t)$. Here, $m_p(t)$ is the antipodal data with the subscript p indicating the transmitter number and having the same definition as equation (3.1). The PN code, $c_p(t)$, is unique and antipodal for each transmitter and the subscript again indicates the transmitter number. The local oscillator, $\cos(2\pi f_c t)$, is also BPSK, and the coding mask transmitter also amplifies the transmitted signal by a pre-set gain A.

The signal then passes through an additive white Gaussian noise (AWGN) channel, which adds the noise term $n(t)$ to the combined signal. The signal received is, therefore, $r(t) = Am_p(t)c_p(t)\cos(2\pi f_c t) + n(t)$. This signal then enters the coding mask receiver shown in Figure 10.

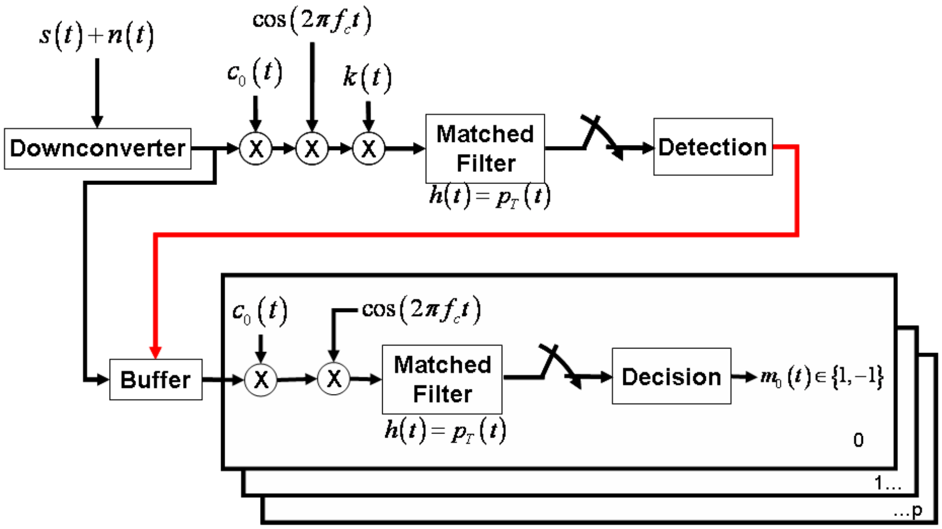

Figure 10. Code Masking Receiver Diagram

In Figure 10, a receiver for a single subset is shown. The PN codes of this subset have some common points allowing the mask to be pre-calculated and preprogrammed into the detection branch. The first PN code in this subset is $c_0(t)$, where the subscript 0 indicates the ordinal numbering of the transmitters in the subset. The first stage accomplishes signal detection by mixing the downconverted signal with the first PN code in the subset $c_0(t)$, the carrier $\cos(2\pi f_c t)$, and the mask $k(t)$ before entering the matched filter. The mask represents the common points of all PN codes in a subset, with 1 used to indicate they are the same and 0 used if even a signal code is different at a particular chip. The results is $k(t)\in\{1,0\}$ and not $\in\{1,-1\}$. The signal received at this point is similar as in the reference receiver with three components, a signal of interest, a double frequency, and noise [7].

$$r'(t)=\frac{A}{2}m(t)k(t)-\frac{A}{2}m(t)k(t)\cos(4\pi f_c t)+n(t)c_0(t)k(t)\cos(2\pi f_c t) \tag{5.1}$$

The matched filter’s impulse response $h(t)$ is a pulse function $p_{Tb}(t-iT_b)$, where the value of the function is 1 during the bit interval and 0 otherwise. The resulting output is the signal component of interest output [7].

$$\begin{aligned} y(t) &= \frac{A}{2}m(t)k(t)*h(t) \\ &= \frac{A}{2}\int_{-\infty}^{\infty} m(\alpha)k(\alpha)h(t-\alpha)\,d\alpha \end{aligned} \tag{5.2}$$

However, the integrand is only non-zero when $t-T_b<\alpha<t$, changing the integration interval. A substitution can be made for the modulated signal, whereby the antipodal data, $m(t)$, can be represented as a single bit at a discrete time instance rather than as the sum of bits and the pulse function, as was shown earlier: $m(t)=\sum d_i p_T(\alpha-iT_b)=d_{\ell-1}$. Additionally, the impulse response of the matched filter taken at a discrete time instance where $t=\ell T_b$ results in the substitution $h(lT_b-\alpha)=p_T(lT_b-\alpha)$. Using these two substitutions changes, the matched filter output in the detection branch to a different form of the mean of the signal received at that point.

$$\begin{aligned} y(t) &= \frac{A}{2}\int_{t-T_b}^{t} m(\alpha)k(\alpha)h(t-\alpha)\,d\alpha \\ y(lT_b) &= \frac{A}{2}\int_{(\ell-1)T_b}^{lT_b} d_{\ell-1}k(\alpha)\,p_T(\ell T_b-\alpha)\,d\alpha \\ &= \frac{A}{2}d_{\ell-1}\beta T_b \end{aligned} \tag{5.3}$$

This result is identical to the reference receiver’s signal of interest with the exception of β, the fraction of T_b for which the mask is one, i.e., $\beta = \frac{1}{T_b}\int_0^{Tb} k(t)\,dt$. Since the matched filter is a low pass filter, the double frequency term is filtered out.

The PSD of the noise at the receive antenna is $S = \frac{N_0}{2}$. As in the reference receiver, when this noise is mixed with the carrier, the resulting PSD is $S = \frac{N_0}{4}$. However, the mask has two possible states, $k(t) = 0$ or $k(t) = 1$. In the former case, the resulting mean (μ), variance (σ^2), and PSD (S) of the noise would be zero. In the latter case, $\mu = 0$, $\sigma^2 = +\infty$, and $S = \frac{N_0}{4}$, which is white Gaussian noise. However, to account for the fraction of the mask that equals 1 in a given period of T_b, the term β can likewise be used as in the signal of interest, giving a resulting input PSD of $S_{in} = \frac{N_0\beta}{4}$. The noise PSD at the output of the matched filter is therefore $S_{out}(f) = \left|H(f)\right|^2 \frac{N_0\beta}{4}$. Its integral is the noise power at the matched filter output $\sigma^2 = \int_{-\infty}^{\infty} \left|H(f)\right|^2 \frac{N_0\beta}{4} df = \frac{N_0\beta}{4}\int_{-\infty}^{\infty} \left|H(f)\right|^2 df$. Applying Parseval’s theorem [12], changes this to $\sigma^2 = \frac{N_0\beta}{4}\int_0^{Tb} h^2(t)dt$. And this in turn solves to the noise power or variance

$$\sigma^2 = \frac{N_0 \beta T_b}{4}. \qquad (5.4)$$

B. PERFORMANCE ANALYSIS

Using the sampled matched filter output for the signal of interest and noise, the decision statistic for signal detection is $N\left(\frac{Ad_{\ell-1}\beta T_b}{2}, \frac{N_0 \beta T_b}{4}\right)$. The probability of bit error is the same as the probability of deciding a “1” was sent when a “-1” was sent. As shown in equation (5.5), the probability of error for a “-1” sent is equivalent to the probability that the mean of the signal plus the noise N is greater than zero.

$$\begin{aligned}\Pr\left(d_{dec} = 1 \mid d_{\ell-1} = -1\right) &= \Pr\left(\frac{-A\beta T_b}{2} + N > 0\right) \\ &= Q\left(\sqrt{\frac{2E_b \beta}{N_0}}\right)\end{aligned} \qquad (5.5)$$

In terms of signal power required, the best possible condition would be $\beta = 1$, when all PN codes are the same. However, this condition would make multiple access impossible—the receiver would be unable to distinguish between the signals.

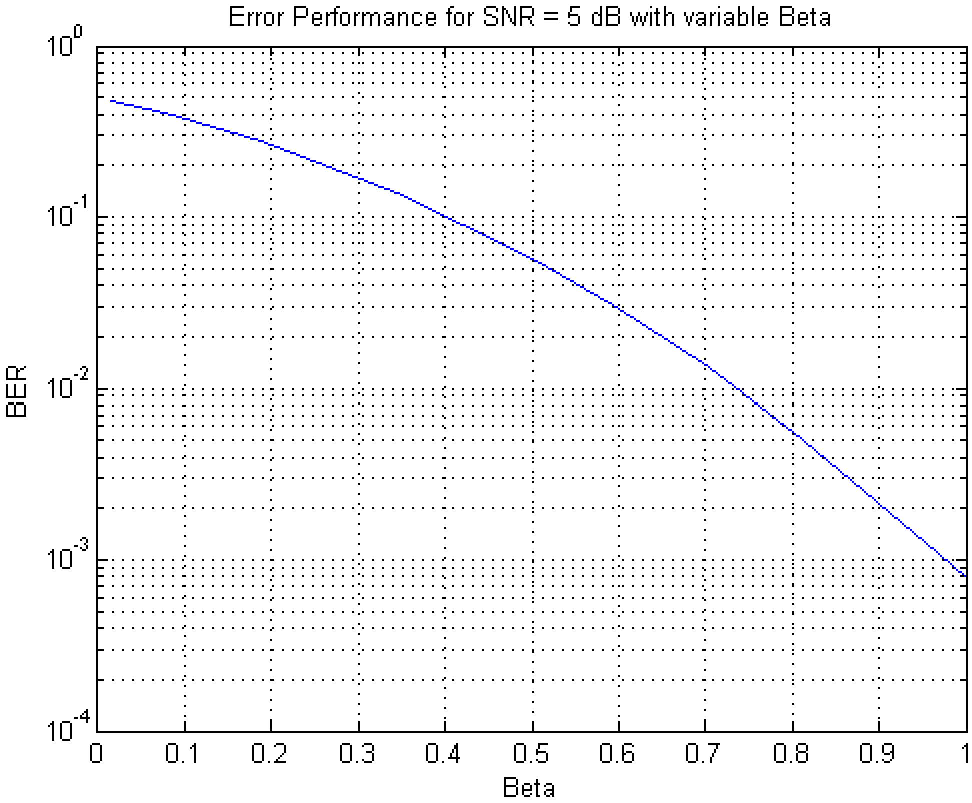

Figure 11. BPSK Error Performance for Fixed SNR and variable β

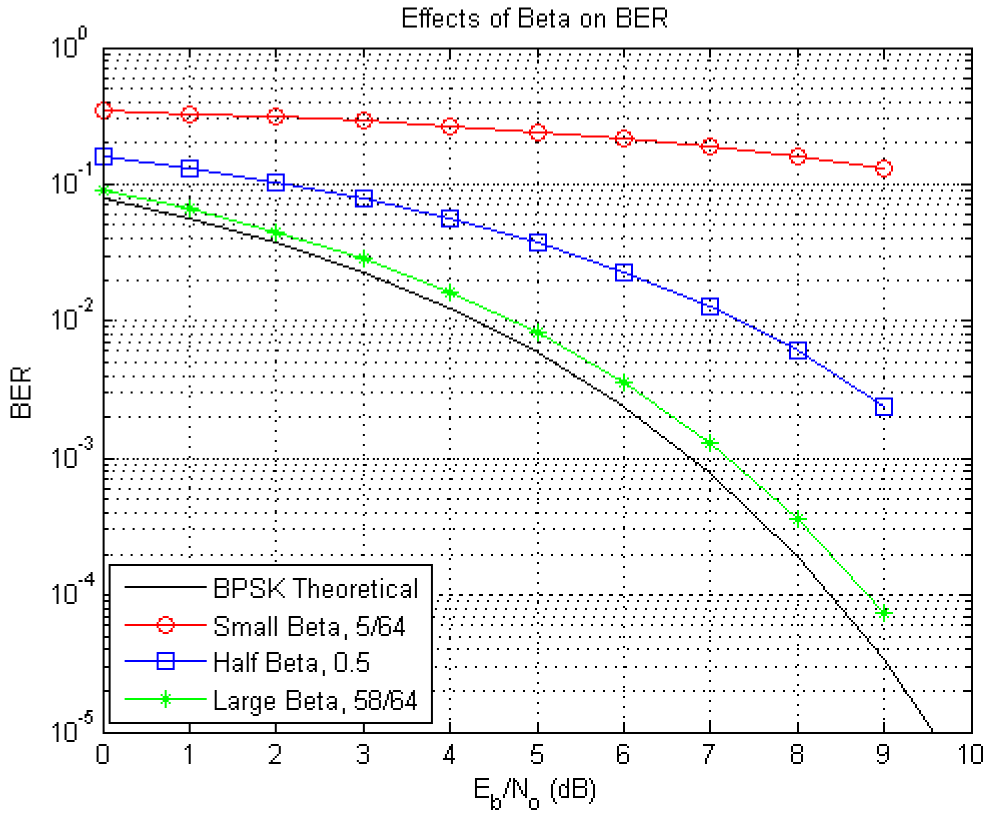

Figure 12. Comparison of Theoretical BPSK and Three Values of β

Fewer chips in common only increases the bit energy required to achieve a similar bit error rate as the reference receiver. The resulting system probability of bit error is worse than reference receiver for the same signal to noise ratio. The first Q function is the error of the detection branch where β represents the effect of the mask. The second Q function is the demodulation error of the receiver branches in the subset. The result is a performance worse than $2Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$.

$$\mathrm{P}_b\left(sys\right)=Q\left(\sqrt{\frac{2E_b\beta}{N_0}}\right)+Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad (5.6)$$

This receiver was not simulated due to time constraints and its significantly poorer BER performance as compared to the receiver in Chapter IV. Additionally because of its inferior theoretical error performance, the overall system performance was not modeled and compared to the reference or cyclic PN code receivers. Chapter VI, therefore, does not include the mask receiver’s performance in its comparison and analysis.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. RESULTS AND COMPARISONS

The reference receiver and cyclic PN code simulations ran were of the same duration, 100,000 bits per dB level. Their results both closely matched the theoretical un-coded BPSK performance in an AWGN channel, as expected. Of note, both simulations did not address fading or synchronization issues.

A. BIT ERROR RATE PERFORMANCE

The reference receiver and cyclic PN code DFT receiver have identical BER performance, as shown in the analysis in Chapters III and IV, and confirmed in the simulation results shown in Figure 13. In both cases, a SNR of approximately 7 dB was required to achieve an error rate of 1 in 1000 and approximately 8.5 dB for 1 in 10000.

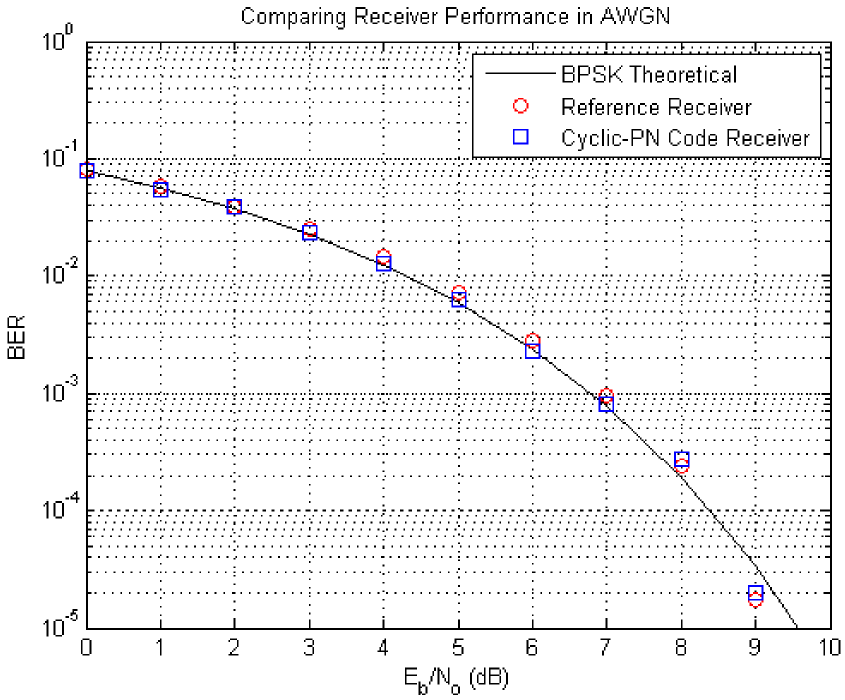

Figure 13. Comparison of BPSK Theoretical (curve), Reference Receiver (circles), and Cyclic PN Code Receiver (squares)

B. SCALABILITY COMPARISON

Examining the reference receiver system with N transmitters, it initially processes the received signals with a common antenna, low-noise amplifier, local oscillator, and low pass filter. From there, such a receiver requires a unique matched filter, sampler, and bit decision block for each transmitter in the system.

The cyclic PN code design in Figure 7 and the reference receiver in Figure 2 have similar components from antenna to LPF. Looking at the number of multiplications required allows for a scalability comparison. For the reference receiver, there are N branches from the LPF forward. Each branch must analyze N signals, with despreading, filtering, sampling, and bit decisions. Therefore, a good estimation of the scalability is total number of multiplications required for despreading in all the branches. Sign multiplications (recall $c[n] \in \{-1,1\}$) are not as complicated as other multiplications, so a coefficient, $\alpha < 1$, is used to indicate the ratio of the complexity of a sign multiplication to the complexity of a complex number multiplication. For each branch in the linear-growth receiver, there are N sign multiplications per bit duration because there are N chips per bit. Therefore, the total effective number of complex multiplies per bit duration is

$$\begin{aligned} C_{reference} &= N(\alpha N) \\ &= N^2 \alpha \end{aligned} \tag{6.1}$$

This value is accepted as the measure of complexity of the reference receiver.

For the cyclic PN code receiver operating in the same scenario, it must complete N sign multiplications per bit duration to despread the long PN code, N complex multiplies in the last mixer, 1 DFT, and 1 IDFT. The FFT and IFFT each require $0.5N\log_2 N$ complex multiplies [6]. Combining these gives the total number of complex multiplies per bit duration for the proposed receiver, which is accepted as its measure of complexity.

$$\begin{aligned} C_{cyclic-PN} &= \alpha N + {}^{N}\!/_{2} \log_2 N + N + {}^{N}\!/_{2} \log_2 N \\ &= N\left(\log_2 N + 1 + \alpha\right) \end{aligned} \tag{6.2}$$

The ratio of the complexity of the reference receiver to the complexity of the cyclic PN code receiver is $\frac{N\alpha}{\log_2 N + 1 + \alpha}$, where the value of α depends on the bit resolution of the numbers multiplied. The complexity of one complex multiply with real and imaginary parts each represented with r bits is equivalent to the complexity of $4r$ sign multiplications. If the number of transmitters exceeds approximately 300, the proposed receiver is a more efficient implementation, assuming $\alpha = \frac{1}{32}$, which corresponds to eight bit resolution. The relative growth in the size of the two systems is shown in Figure 14.

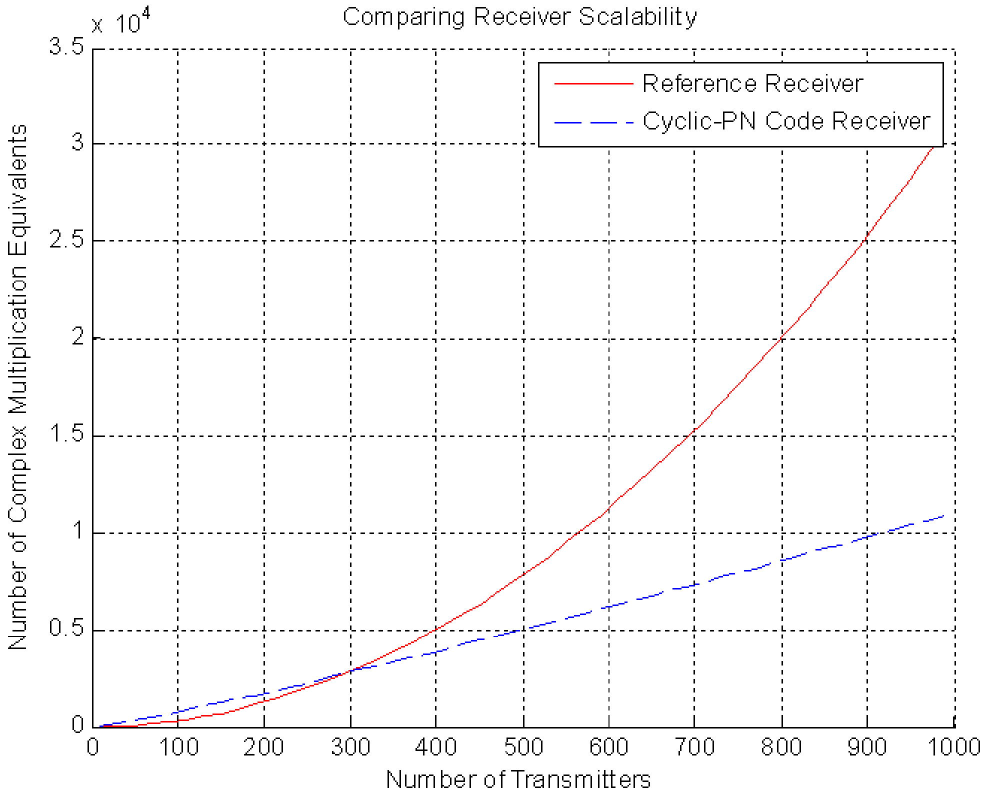

Figure 14. Scalability Comparison for $N = 1000$ and $\alpha = \frac{1}{32}$

VII. CONCLUSIONS AND RECOMMENDATIONS

As can be seen in Figure 8 and Figure 13, the proposed design matches theoretical uncoded BPSK performance (which also matches the reference receiver). The IDFT output, as shown in Figure 9 for a single transmitter, clearly indicates the shift number and bit value. The simulation used an $N = 64$-chip PN code, allowing up to 64 transmitters in the model system. For a full system, 64-bit decisions would be required for each T_b interval. $N = 64$ was chosen for illustration purposes only, and much larger values are practical. The cyclic-PN code receiver maintains the BER while improving scalability, making it a viable solution.

A. PERFORMANCE COMPARISON

1. Reference vs. Cyclic PN Code

Since the receivers had similar error performance for an uncoded signal in AWGN, it stands that the key difference is receiver complexity. In this, the cyclic PN code receiver has superior scalability when the number of transmitters is large, as shown in equations (6.1) and (6.2), as well as in Figure 14.

2. Reference vs. Code Mask

The theoretical bit error rate performance of the code mask receiver was inferior to the reference receiver. The code mask receiver’s advantage lies in the fewer required processes to receive a single signal, but its higher error rate offsets this advantage.

3. Code Mask vs. Cyclic PN Code

The cyclic PN code receiver has the same bit error rate performance as the reference receiver, but superior scalability and, therefore, is the better choice when the number of transmitters is large. Additionally, the DFT-based filtering in the cyclic PN code receiver was more efficient in terms of space, but potentially less efficient for a single signal as the total number of processes required is the same regardless of the number of transmitters in the system. However, for a large number of transmitters, the coding mask receiver is inefficient, as it requires two stages of processes, resulting in more overall complexity than either the reference or cyclic PN code receivers. The coding mask receiver’s theoretical BER, even under the best conditions would be twice that of the cyclic PN code receiver.

B. FUTURE WORK

1. Improving the Cyclic PN Code Model

The models presented were simulated in an AWGN environment. Simulating the same models in a fading and jamming environment would better match potential real-world scenarios. Large networks of sensors deployed in a mountainous or urban environment would suffer fading or jamming effects in addition to AWGN. Validation of the cyclic PN code receiver’s error performance in these environments is important.

MATLAB’s “randint” function and Simulink’s equivalent block generated the simulations’ random unipolar binary sequences. Future researchers could develop a truly orthogonal cyclically shifted PN code and validate its performance, especially the prevention of inter-transmitter interference. Additionally, future research should synthesize an effective forward error correction code using BPSK or QPSK for the cyclic PN code receiver.

Quantifying the LPE features expands the validity of the proposed communications system. This would require a larger model with multiple transmitters and a non-cooperative interceptor. Additionally, models of the most likely operating environments, which account for potential adversaries, are necessary in order to analyze potential system performance.

This thesis' models also need a synchronization analysis. Both the reference and cyclic PN receiver in this thesis relied on assumed synchronization to make bit decisions. Synchronization blocks or a non-coherent form of modulation might solve this problem.

2. Improving LPI and Data Rate Performance

Strong encryption would improve the systems defenses against exploitation of the intercepted signal. However, encryption would not improve overall LPE as adversaries could potentially exploit signal presence and geolocation through direction finding to localize transmitters.

The systems designed and tested in this thesis used BPSK modulation. Modulation techniques like QPSK, M-PSK, M-frequency shift keying (MFSK), and M-quadrature amplitude modulation (MQAM) were not tested. Other modulation techniques could achieve higher data rates with trade offs in transmitter power, error rate, and exploitation.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. REFERENCE RECEIVER SIMULATION CODE

All code in this appendix was generated using MATLAB v7.6 (R2008a). The SIMULINK file that supports this simulation is provided via electronic media.

```
% Name: LCDR Frank Cowan
% Title: Reference Transmitter and Receiver
% Simulation Project: NPS EE Thesis
% Date Created: 22 Jan 2009
% MATLAB/Simulink Version: 7.6.0, R2008a

% This file provides the variable inputs for a simulation of the
reference transmitter and receiver for my thesis.  All data is saved
to files and will be plotted using a separate m-file.

clc;  % clears the command window
clear all;  % clears workspace

% Transmitter parameters
Chip_Freq = 1.2288e6; % Rc, set at 1.2288 MHz
Bit_Freq = Chip_Freq/64; % Rb, set at 19200 Hz, 1/64 of Rc
Carrier_Freq = 4*Chip_Freq; % 4915200 Hz, frequency of carrier
CarrierSampleTime = 1/(16*Carrier_Freq);  %16 samples/cycle
TransmitterGain = 1; % A, for Arbitrary gain

%AWGN channel parameters
EbNo_array = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]; % the EbNo increments
InputSignalPower = (TransmitterGain^2)/2; % used in the AWGN channel
block for input signal power
AWGN_bps = 1; % used in the AWGN channel block for bps setting
AWGN_symbol_period = 1/Bit_Freq; % sets the symbol period for the
AWGN in the channel

% Reference Receiver parameters
IntegrationThreshold = 0; % required parameter for the reference re-
ceiver decision block

% Conduct simulation
Run_No = 1; % sets the Run_No to default setting of 1
sim_length = 3; % same simulation length used for each loop - based
on the time interval used, this works out to around 50,000 samples
per loop

for EbNo_idx = 1:10
    % Simulation
    Run_No % displays the run number in the work space to allow user
to observe progress
    EbNo = EbNo_array(EbNo_idx); % assigns the value of EbNo used in
the AWGN block in the channel
```

```
    sim('Reference_RCVR_Model_V1', sim_length); % calls the simula-
tion and assigns the sim length

    % Calculate the number of errors
    Bits_sent = length(Data_Transmitted)-3; % number of bits sent by
transmitter.  -3 is needed to account for delays in the system.
    Errors = sum(xor(Data_Transmitted, Data_Recovered))-1; % calcu-
lates the number of bit errors by XOR'ing the data transmitted and
data received.
    Error_Rate(EbNo_idx)=(Errors/Bits_sent); % calculates the error
rate by dividing the number of errors by the total number of bits
sent.  This is the data that will be plotted after the loop.

    % save the number of bits created for the simulation run
    Simulation_Length(EbNo_idx) = Bits_sent; % saves the value of
Bits-sent to an array for later analysis

    % increment counter
    EbNo_idx = EbNo_idx + 1; % increments the EbNo index and walks
through the EbNo array
    Run_No = Run_No + 1; % increments the counter for the loop
end

save('Error_Rate', 'Error_Rate') % Saves the Error_Rate variable in-
to a separate file
save('EbNo_array', 'EbNo_array') % Saves the EbNo_array variable in-
to a separate file


% End of Reference RCVR
```

APPENDIX B. CYCLIC PN-CODE RECEIVER SIMULATION CODE

All code in this appendix was generated using MATLAB v7.6 (R2008a).

A. CYCLIC PN CODE RECEIVER SIMULATION MATLAB CODE

```
% Name: LCDR Frank Cowan
% Title: Cyclic PN receiver Code
% File name: PN_Code_RCVR_VF.m
% Simulation Project: NPS EE Thesis
% Date Created: 06 May 09
% MATLAB/Simulink Version: 7.6.0, R2008a

% Linked functions: upsample2.m

% This files simulates a single transmitter system for the cyclic PN
code
% transmitter and receiver system detailed in my thesis.

clc;  % clears the command window
clear all;  % clears workspace

% create data array
Rb = 1; % bit rate
Rc = 64*Rb; % chip rate
alpha = 16; % ratio of fc to Rc
fc = alpha*Rc; % carrier frequency = alpha factor times Rc
beta = 16; % ratio of Rs to fc
Rs = beta*fc; % sample rate
Ts = 1/Rs; % sampling time - inverse of sampling rate
N = 49999; % length of vector factor for non random data vector
n = 0:N; % length of non-random data vector
Data_Array = (-1).^n; % alternating 1,-1 square wave - not random
Message_Length = length(Data_Array); % measures the length of the
random data vector - not used with the non-random data vector
Chipping_Matrix = ones(1,64); % 64 chips per bit - vector of 64 ones

% create the unique PN code for transmitter one - base code with one
shift
Short_PN = randint(1,length(Chipping_Matrix))*2-1; % base short PN
code with no shifts
Long_PN = randint(1,length(Chipping_Matrix))*2-1; % base long PN
code - this really should be longer than one bit duration
PN_Code1 = circshift(Short_PN,[0,1]).*Long_PN; % creating the unique
PN code for Transmitter #1s

% Energy per bit and amplitude calculations
```

```
Eb = 0.5; % energy per bit - this will remain unchanged each loop -
final problem should use 0.5, but 2 seems to be working better
A = sqrt(2*Eb*Rb); % peak amplitude

% this is the variable noise loop
SNR = [0,1,2,3,4,5,6,7,8,9]; % SNR in DB

for h = 1:length(SNR) % control for variable SNR loop
    No(h) = Eb/(10^(SNR(h)/10)); % No calculation - uncomment for
noise loop

    % this is the transmitter/receiver simulation loop
    for k = 1:length(Data_Array) % control for the simulation loop
        Chipped_Data = Data_Array(k)*Chipping_Matrix; % repeats the
data bit Rc times
        Mixed_Data = Chipped_Data.*PN_Code1; % mixes the chipped da-
ta with the combined PN code over one bit interval
        Sampled_Data = upsample2(Mixed_Data,alpha*beta); % upsamples
the data to the sample rate

        % build local oscillator
        time = (0:(length(Sampled_Data)-1))*Ts; % time vector
        Cosine_Array = cos(2*pi*fc*time); % cosine vector

        % build the modulated waveform that will be transmitted
        Modulated_Signal = A*Sampled_Data.*Cosine_Array; % mixes the
cosine array with the sampled data - result is the modulated signal
- prof Kragh changes.

        % simulate the AWGN channel and add noise to signal
        Noise = sqrt(Rs*No(h)/2)*randn(1,Rs); % Noise calculation -
must be repeated for each bit in order to be independently random
        Received_Signal = Modulated_Signal + Noise; % noise is added
to the signal

        % receive and process the signal
        Downcoverted_Signal = Received_Signal.*Cosine_Array; % mix
with same cosine
        LongPN_up = upsample2(Long_PN,(alpha*beta)); % upconverts
the long PN code to mix with the data received at the sample rate
        Demixed_Data = LongPN_up.*Downcoverted_Signal; % mixes the
data received prior to LPF input with the upsampled long PN code,
b(t)
        p = ones(1,Rs/Rc)*Rc/Rs; % averaging factor for downsampling
from Rs to Rc - change fc back to Rc later
        LPF_out = conv(Demixed_Data,p); % evenly spaces convolution
and averaging - final LPF output, must be downsampled still
        DFT_input =
LPF_out((0+(alpha*beta)):(alpha*beta):64*(alpha*beta)); % Downsam-
pling of the LPF output from sample rate to chip rate
        G = fft(DFT_input); % DFT of signal received
        C = fft(Short_PN); % Reference PN code DFT
        C0 = conj(C); % conjugate of reference PN code
        L = C0.*G; % mix DFT outputs
        l(k,:) = ifft(L); % IDFT of mixer output - matrix of data

    end % ends the transmitter/receiver system simulation loop

%---back in the outer loop

    % make bit decisions from IDFT output - idft_out_x_axis
    for m=1:length(Data_Array)
        if l(m,2) >= 0
            Bit_Decision(m) = 1;
        elseif l(m,2) < 0
            Bit_Decision(m) = -1;
        end
    end % ends the bit decision loop

    % convert both in and out data to unipolar binary
    Data_In = (Data_Array + 1)/2;
    Data_Out = (Bit_Decision + 1)/2;

    % determine error rate
    Error_Array = xor(Data_Out,Data_In);
    Total_Errors = sum(Error_Array);
    Error_Rate(h,:) = Total_Errors/length(Data_In); % change 1 back
to h after troubleshooting

end % ends the outer loop with counter k - for variable noise

% change Error rate into a horizontal vector - data array
Error_Rate_PN_RCVR = Error_Rate';

%export error rate
save('Error_Rate_PN_RCVR','Error_Rate_PN_RCVR') % Saves Error_Rate
variable into a separate file
save('SNR','SNR') % Saves EbNo_array variable into a separate file
save('Eb','Eb') % saves the value of Eb - it will be included on the
plot

% Error performance will be plotted in a separate file.


% End File
```

B. SUPPORTING FUNCTION ‘UPSAMPLE2’ MATLAB CODE

```
% Name: LCDR Frank Cowan
% Title: upsample2 - a function
% File Name: upsample2.m
% Simulation Project: NPS EE Thesis
% Date Created: 23 Apr 09
% MATLAB/Simulink Version: 7.6.0, R2008a

% This function upsamples a given data sequence by a user-entered
```

```
factor.
% It is designed as a component of PN_Code_RCVR_VF, the simulation
of the
% cyclic PN code for my thesis.

function [out] = upsample2(in,U)
    out = [];
    for k = 1:length(in)
        out = [out in(k)*ones(1,U)];
    end

% End File
```

LIST OF REFERENCES

[1] D. L. Nicholson, *Spread Spectrum Signal Design: LPE and AJ Systems*. Rockville: Computer Science Press, 1988.

[2] C. Kopp, *NCW101: An Introduction to Network Centric Warfare*. [electronic resource] Noble Park: Air Power Australia, 2008.

[3] G.M. Dillard, M. Reuter, J. Zeidler, and B. Zeidler, "Cyclic code shift keying: a low probability of intercept communication technique," IEEE Transactions on Aerospace and Electronic Systems, vol.39, no.3, pp. 786–798, July 2003.

[4] J. S. Lee and L. E. Miller, *CDMA Systems Engineering Handbook*. Boston: Artech House, 1998.

[5] T. Ha, *Digital Communication: Principles and Practice*. (to be published).

[6] R. D. Strum and D. E. Kirk, *First Principles of Discrete Systems and Digital Signal Processing*. New York: Addison-Wesley, 1989.

[7] B. Sklar, *Digital Communications: Fundamentals and Applications, 2nd Ed*. Upper Saddle River: Prentice Hall, 2001.

[8] M. K. Simon, J. K. Omura, et al., *Spread Spectrum Communications Handbook, Revised Edition*. New York: McGraw-Hill, 1994.

[9] The MathWorks, "Simulink 7 User's Guide," March 2009, http://www.mathworks.com/access/helpdesk/help/pdf_doc/simulink/sl_using.pdf.

[10] The MathWorks, "Communications Block Set 4 User's Guide," March 2009, http://www.mathworks.com/access/helpdesk/help/pdf_doc/commblks/usersguide.pdf.

[11] R. J. Tallarida, *Pocket Book of Integrals and Mathematical Formulas, 2nd Ed*. Boca Raton: CRC Press, 1992.

[12] R. Cristi, *Modern Digital Signal Processing*. Pacific Grove: Brooks/Cole-Thomson Learning, 2004.

[13] H. L. Van Trees, *Detection, Estimation, and Modulation Theory*. New York: Wiley, 1971.

[14] J. H. Reed, *Software Radio A Modern Approach to Radio Engineering*. Upper Saddle River: Pearson, 2002.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. LCDR Frank Cowan
   COMSEVENTHFLT, N39A
   Yokosuka, Japan

4. Assistant Professor Frank E. Kragh
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

5. Professor Herschel H. Loomis
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

6. Professor Phillip Pace
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

7. Donna Miller
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California