Spring 2-18-2010

PERFORMANCE OF SELF-ENCODED SPREAD SPECTRUM UNDER WORST-CASE JAMMING

Casey L. Deyle
University of Nebraska at Lincoln, cdeyle@gmail.com

Deyle, Casey L., "PERFORMANCE OF SELF-ENCODED SPREAD SPECTRUM UNDER WORST-CASE JAMMING" (2010). *Theses, Dissertations, & Student Research in Computer Electronics & Engineering*. 3.
https://digitalcommons.unl.edu/ceendiss/3

PERFORMANCE OF SELF-ENCODED SPREAD SPECTRUM UNDER WORST-CASE JAMMING

by

Casey Deyle

A THESIS

Presented to the Faculty of
The Graduate College of the University of Nebraska
In Partial Fulfillment of Requirements
For the Degree of Master of Science

Major: Telecommunications Engineering

Under the Supervision of Professor Lim Nguyen

Lincoln, Nebraska

November 2009

Performance of Self-Encoded Spread Spectrum Under Worst-Case Jamming

Casey Deyle, M.S

University of Nebraska 2009

Advisor: Lim Nguyen

Spread Spectrum Communications uses m-sequences (sometimes referred to as Pseudo Noise or PN sequences) modulated with a data signal to create a transmission signal that takes up more bandwidth than the original information signal. Self-Encoded Spread Spectrum (SESS) uses spreading codes generated by the transmitted signal, eliminating the need to synchronize m-sequences between the transmitter and receiver, thus making the channel more secure. This paper will discuss the performance of SESS system in Additive White Gaussian Noise (AWGN) and Rayleigh fading channels, as well as the use of an iterative detection to increase the performance of the system. Introduced in this paper is pulsed noise jammer (PNJ) to a SESS system, which is the worst-case jamming scenario for a SESS system, and possible ways to overcome these jamming conditions. The performance of the SESS system in this paper is analyzed using simulations that measure the probability of error (sometimes called Bit Error Rate or BER) vs signal-to-noise ratio (also called SNR or Eb/No).

Acknowledgements

I would like to point out the help that my advisor, Lim Nguyen, has given throughout not only my education while at the University of Nebraska Lincoln-Omaha, but also for the help, advice, guidance, ideas, and time that he has give me for this thesis (as well as the other members of my committee).

Certainly, all the individuals and professors that helped throughout my undergraduate and graduate career deserve credit for getting me to this point and providing irreplaceable memories and knowledge that contributed to this thesis.

Lastly, I need to thank my family for supporting me financially and mentally during college. And to my girlfriend for putting up with the long nights of homework, simulations, and reports needed to get my graduate degree.

TABLE OF CONTENTS

LIST OF FIGURES AND EQUATIONS

CHAPTER 1 – Introduction

1.1 History of Spread Spectrum Communications

Spread spectrum communications refer to any modulation scheme that creates a much wider bandwidth for the transmitted signal than the information bandwidth. At first glance, it would appear spread spectrum systems are wasteful as they require more bandwidth to transmit a signal. However, there are several benefits to spread spectrum systems including:

1 Rejects hostile jamming, as well as unintentional interference.
2 Lowers probability of intercept because its spread over larger bandwidth, making detection harder because signal is likely below the noise level.
3 Provides message privacy because any unauthorized listener who lacks prior knowledge of the system and the timing cannot demodulate the signal.
4 Provides a good resistance from multipath signals.
5 Offers a high degree for accuracy for measuring distance.
6 Like in Code Division Multiple Access (CDMA), it allows simultaneous signaling on the same frequency.

Although there is no real clear inventor of spread spectrum communication, all sources agree that the majority of advancement and research came from efforts during World War II to provide secure means of communication in hostile environments [1]. One of the most interesting and important milestones in the development of spread spectrum communications came from a patent filed in mid 1941 by Hedy Lamarr and George Antheil. Hedy, who move to the United States from Austria and later became a well-known movie star, help George developed a method for controlling a torpedo that implemented a frequency-hopping guidance system. The transmitter carrier would change frequency according to a randomized non-repeating code [2]. In 1948, the U.S. mathematician Claude Shannon published a Mathematical Theory of Communication as a monograph in the Bell System Technical. This paper is remarkable because of its elegant theorems, derived from statistical characterizations of both the information source and the channel effects [3]. This includes the theorem for perhaps which he is most well known, the communication capacity of the band-limited additive Gaussian noise channel. His theorem is:

(1)

$$C = W * log_2\left(1 + \frac{S}{N}\right)\ bits/s$$

where the W is the channel bandwidth in Hz, S is the signal power in watts, and N is the total noise power of the channel in Watts [4].

One of the early adoptions of direct sequences occurred for the purpose of ranging for the tracking-range radar systems at the Jet Propulsion Laboratory (JPL) for use in the Corporal guidance system link. Frank Lehan of JPL noted that radar signal correlation function was of prime importance in determining the accuracy of the range estimate. The pseudo-noise codes (sometime called m-sequence) were investigated starting around the 1950’s. Also in the 1950’s, Robert Price and Paul Green of Lincoln Laboratory developed a signal processing technique called Rake. The Rake processor uses the fine time-resolution capability of wideband signals to resolve signals arriving over different propagation paths, and inserts them into a diversity combiner to coherently construct a stronger received signal [3]. These ideas lead to developments in Code Division Multiple Access (CDMA) systems that would allow multiple users to gain access to the channel.

Up until the 1970's spread spectrum systems where mainly military developed and controlled. It was primarily used for satellite communications, and was developed by both the Western and Russian military forces. The first major spin off for commercial applications was the Global Positioning System (GPS) which uses CDMA based satellite technology. Before CDMA was used in mobile phones in the late 1980’s to early 1990’s, traditional mobile phones used FM and tone signaling with newly perfected microprocessor to enable automated calling by a few select users in a given city. The Improved Mobile Telephone Service (IMTS) allowed 10 to 25 channels for a given region, with mobile transmit powers running nearly 100 Watts ERP. This paved the way for a system developed at Bell Labs called Advanced Mobile Phone System (AMPS), which shifted the channel control and call processing to land-side process and divided coverage area into smaller cells, to increase coverage area [5].

In 1989, the first experiments using cellular CDMA at 800 MHz were conducted by Qualcom Inc. CDMA system could provide high quality and a capacity greater than ten times the capacity of the existing AMPS cellular system[6]. This lead to the development of the 2G standard called IS-95 wideband for 800 MHz cellular radio systems, which rested fully on the spread spectrum CDMA platform. This 2G standard was widely accepted throughout the 1990’s and helped set the platform for the major 3G mobile radio specifications UMTS and CDMA2000, whose variations are still used in modern cell phone networks [7]. The development of CDMA was pioneered at about the same time as other spread spectrum technologies like Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA). The main drawback in developing spread spectrum technologies for non-military use before the 1990's was the lack of enabling technologies. When high speed Digital Signal Processing (DSP) chips became readily available, there was a rapid acceleration in the development of spread spectrum based systems [8]. These rapid developments have led to the adoption of spread spectrum technologies into everyday life. Hybrid systems (systems that combine a few different spreading techniques) can be found in the ubiquitous IEEE 802.11 standard, hand held and car GPS navigation system, and virtually any other modern wireless communication device.

Spread spectrum refers to a telecommunications technique in which a signal is transmitted in a bandwidth considerably greater than the frequency content of the original information. The main signal structuring techniques are frequency hopping and direct sequence, but there are also many different forms of each as well as hybrids that combine multiple techniques. These techniques can be used for multiple access and offer multiple functions. First, they decrease the potential interference to other receivers while achieving privacy. Second, they increase the immunity of spread spectrum receivers to noise and interference. Lastly, spread spectrum makes use of a sequential noise-like signal structure to spread the normally narrowband information signal over a relatively wide band of frequencies. The receiver correlates the signals to retrieve the original information signal. In frequency hopping the signal power is spread over a wide band sequentially in time. This is accomplished by randomly hopping the carrier from one frequency to the next. In direct sequence, the signal power is instantaneously over a wider bandwidth. SESS is based off of direct sequence due to the ease of simulating in digital communications, but both direct sequence and frequency hopping techniques benefit from being able to use many different types of modulation techniques [7].

There are many different types of modulation schemes that can be used in conjunction with spreading techniques. The three main digital modulation methods are Amplitude-shift keying (ASK), Frequency-shift keying (FSK), and Phase-shift keying (PSK). ASK is a form of modulation that represents digital data as variations in the amplitude of a carrier wave. The amplitude of an analog carrier signal varies in accordance with the bit stream (modulating signal), keeping frequency and phase constant. The level of amplitude can be used to represent binary logic 0s and 1s.The ASK technique is also commonly used as the basic system behind digital data transmitted over optical fiber (light pulses). In FSK digital information is transmitted through discrete frequency changes of a carrier wave. An example of the simplest FSK would be binary FSK (BFSK). BFSK uses a couple of discrete frequencies to transmit binary information. Multiple frequency-shift keying (MFSK) is a more advanced form of FSK, and is used for VHF & UHF communications (radio and over the air television). PSK conveys data by changing, or modulating, the phase of a reference signal. Although frequency modulation and phase modulation are very similar, in practical signals, phase modulation is often considered superior [9]. As with any digital modulation scheme, it uses distinct signals to represent digital data. In PSK a finite number of phases are each assigned a unique pattern of binary bits. The demodulator determines the phase of the received signal and maps it back to the symbol it represents, and thus receiver is able to recover the original data. A convenient way to represent PSK schemes is on a constellation diagram, seen in Figure 1 below.

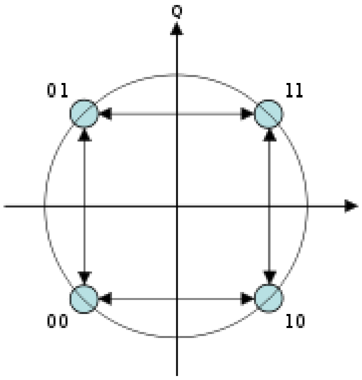

Figure 1 - Constellation Diagram of QPSK

Figure 1 represents the possible symbols that may be selected by a given modulation scheme as points in the complex plane for Quadrature PSK (QPSK), which is a variant of PSK modulation that uses 4 different values of the phase to transmit data. Variations of PSK can be found in numerous communications standards from 802.11 wireless standards (OFDM with QPSK) to cell phones (CDMA), digital television and modems (QAM) [10]. Binary PSK (BPSK), the simplest form of PSK, uses two phases which are separated by 180 degrees. Since BPSK is only able to modulate at one bit per symbol, it is unsuitable for high data-rate applications when bandwidth is limited. However, this modulation is considered the most robust of all the PSKs since it takes the highest level of noise, jamming, or distortion, when coupled with a correlation detector or matched filter, to make the demodulator reach an incorrect decision (QPSK has the same BER, but requires twice the energy since two bits are transmitted). Because BPSK does best in probability of error, it is the de facto standard in testing performance of spread spectrum systems and modulation techniques. Generally speaking, the transmission of a signal in digital form come much closer to the realization of the limit in Shannon's equation (1) han the transmission of signals in analog form [9]. For these reason, digital BPSK is used in this thesis as a control in order to measure the performance of SESS, which is based off of a direct sequence spreading technique with a digital BPSK modulation.

1.2 Motivation and Scope of Research

This thesis research is the extended work of the self-encoded spread spectrum (SESS) which is first proposed in [11]. Since their existence, spread spectrum techniques have used some variation of pseudo-noise (PN) or predetermined codes to achieve spreading. SESS eliminates this need by using spreading codes generated by the transmitted signal, thus also eliminating the need to synchronize PN codes between the transmitter and receiver. This makes the channel more secure, because PN codes both at the transmitter and receiver can be deterministic.

SESS system with iterative detector has been shown to have a 3 db performance gain over BPSK modulation in an Additive White Gaussian Noise (AWGN) channel. In a Rayleigh fading channel, it achieves a performance gain of 15 db with just the iterative detector and even greater when other methods are applied additionally increase the performance.

To facilitate future improvements SESS and its use for security in hostile environments it must be studied in jamming channels. Jamming in a communication channel makes the probability of error far worse than that of the standard noise or fading channel. So much, that the noise is often ignored as it is insignificant compared to the jamming. The worst method of jamming to any time varying spread spectrum system is known pulsed-noise jamming. Examining the performance of a SESS system under these conditions can help to show the versatility of SESS. As well as, give possible insight to future improvements that could add to the security and performance of SESS.

This thesis will provide a background, as well as, performance measures for binary PN sequences in chapter two. Then chapter three will outline the SESS system, that uses the no PN sequences, and analyze the performance SESS system in AWGN and Rayleigh fading channels. Also, introduced and analyzed in this chapter is an iterative detector that is used to help increase the performance of SESS. Chapter four establishes the jamming channel and sets up the worst-case jamming model. While chapter five analyzes effects of worst-case jamming on SESS, chapter five focuses on the effects of worst-case jamming on the iterative detector. Finally, chapter seven looks at possible solutions to help the worst-case jamming and examines the effects of jamming inside of AWGN and Rayleigh fading channels.

Chapter 2 – Binary PN Sequences

2.1 Introduction

By far the most widely known binary PN sequences are the maximum-length shift-register sequences. A maximum-length shift-register sequence, or m-sequence, has a length of n= 2^n – 1 bits, where m is the number stages in a shift register with linear feedback (as seen in Figure 2 below).

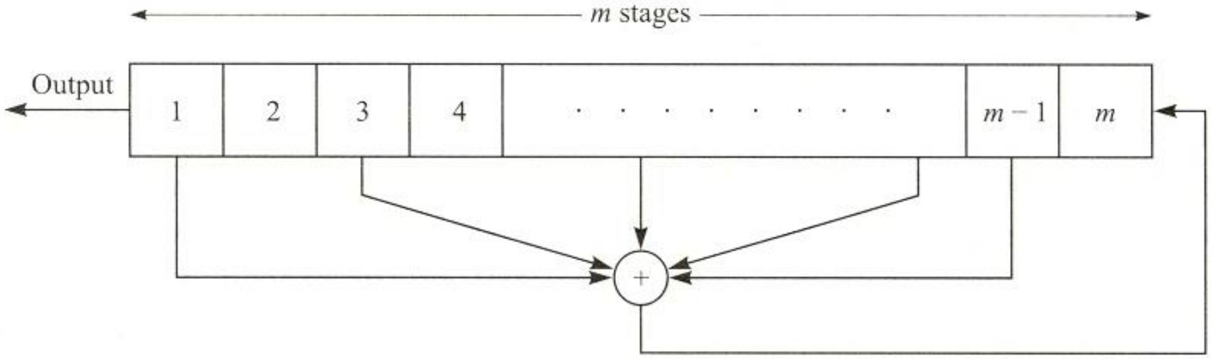

Figure 2 - M-stage Binary Sequence Generator [12]

2.2 Analysis Method – Autocorrelation

An important characteristic of a periodic PN sequence is the periodic autocorrelation function. Correlation refers to the relation is the mutual relationship between two or more random variables. Thus, autocorrelation is the correlation of a signal with itself. Autocorrelation is useful for finding repeating patterns in a signal, such as determining the presence of a periodic signal which has been buried under noise. Ideally, a pseudorandom sequence should have an autocorrelation function with the property that for $\phi(n) = 0$ and $\phi(j) = 0$ for $1 \leq j \leq n-1$. For m-sequences the autocorrelation function is written as:

(2)

$$\phi(j) = \begin{cases} n & (j = 0) \\ -1 & (1 \leq j \leq n-1) \end{cases}$$

For large m-sequences, the size of the off-peak values of the autocorrelation are relative to the peak value $\phi(j) / \phi(0) = -1/n$, which becomes small and irrelevant. Therefore, m-sequences are almost ideal when viewed in terms of autocorrelation function [12].

In anti-jamming applications of PN spread spectrum signals, the period of the sequence must be large in order to prevent the jammer from learning the feedback connections of the PN generator. However, this requirement is impractical in most cases because the jammer can determine the feedback connections by observing only 2n-1 chips from the PN sequence. This vulnerability of the PN sequence is due to the linearity property of the generator. To solve this, output sequences from several stages of the shift register or outputs from several distinct m-sequences are combined in a non-linear way to produce a non-linear sequence that is considerably more difficult for the jammer to learn.

The periodic autocorrelations functions for most of the Gold sequences are not as good as the periodic autocorrelations functions for m-sequences. Also, Gold sequences of period 2^n-1 can be generated by linear feedback shift registers with 2n storage elements, and so their periods are approximately the square root of the maximum periods for linear sequences generated with the same number of storage elements. However, if a large number of sequences is required for a given application, and if the cross correlation function is more important than the autocorrelation function, then the Gold and Kasami sequences are much better than m-sequences [13].

Gold sequences take a pair of m-sequences with sequences of length n are generated by taking the modulo-2 sum of one (called *a*) with the n cyclical shifted version of the other (called *b*). This generates $2^n + 1$ different sequences each period $2^n - 1$ and such that the cross-correlation function $\phi(a,b)$ of any pair of such sequences satisfies the equation [14]:

(3)

$$|\phi(a,b)| = \begin{cases} 2^{(n+1)/2} + 1, & for\ odd\ n \\ 2^{(n+2)/2} + 1, & for\ even\ n \end{cases}$$

Welsh showed that the maximum cross-correlation between any two sequences in a set length N sequences of cardinality M is lower bounded [15]. Specifically, he showed that the maximum cross-correlation between two sequences is lower bounded by $\sqrt{(\mathrm{M}-1)/(\mathrm{MN}-1)}$, where M is the number of codes in the set. For relatively large sets it can be concluded that the maximum cross-correlation is greater than $\sqrt{1/\mathrm{N}}$. By applying these bounds to Gold sequences ($M = 2^n + 1$, $N = 2^n - 1$), it can be seen that the max cross-correlation is [16]:

(4)

$$\phi_{max} \approx \begin{cases} \sqrt{2/N}\ , & for\ odd\ n \\ \sqrt{4/N}\ , & for\ even\ n \end{cases}$$

Gold codes obviously do not meet the meet the lower bound of $\sqrt{1/\mathrm{N}}$ derived by Welch. Kasami sequences use a similar method of to generate a smaller set of M = $2^{n/2}$ binary sequences of period N = 2^n -1 (for even n). Kasami sequences do so by taking a m-seqeunce (called *c*) and forming a second binary sequence from it by taking every $2^{n/2}$ + 1 bit (called *d*). Then *c* is added with a time shifted version of *d* using modulo two. The set which is created by taking all Kasami sequences generated by different time shifts of *d*, as well as the original *c* and *d* sequences, form the Kasami set of sequences. This set has is known to have $2^{N/2}$(M) different sequences of length 2^n-1 (N). Thus, the $\phi_{max} = 2^{n/2} + 1$, which satisfies the Welsh lower bound, making Kasami sequences optimal for cross-correlation [17].

2.3 Analysis Method – Merit Factor

A classical problem of digital sequence design is to determine those binary sequences whose aperiodic autocorrelations are collectively small according to some suitable measure. This is achieved by what is called the merit factor. It is used to determine whether coded signal is a good or poor spreading signal. For example, let a real sequence of length N be represented by S = $[x_0, x_1, \dots, x_{N-1}]$. The aperiodic autocorrelation function of sequence S of length N is:

(5)

$$A(k) = \begin{cases} \sum_{n=0}^{N-k-1} S_n S_{n+k}\,; & 0 \le k \le N-1 \\ \sum_{n=0}^{N+k-1} S_n S_{n-k}\,; & -N+1 \le k \le 0 \end{cases}$$

Golay in [18] defined the merit factor as the ratio of main lobe energy to side lobes energy of autocorrelation function of sequence S. The merit factor can be mathematically is defined as,

(6)

$$Merit\ Factor = \frac{A(0)^2}{2\sum_{k\neq 0}^{N-1}|A(k)|^2}$$

the denominator term represents the energy in the side lobes. The merit factor must be as large as possible for good sequences [19]. The larger the merit factor of a binary sequence that is used to transmit information by modulating a carrier signal, the more uniformly the signal energy is distributed over the frequency range; this is particularly important in spread-spectrum communication.

When the merit factor is applied to m-sequence, Gold, and Kasami sequences found in the previous section, [20] concluded that, through simulation, when the sequence length becomes large, the merit factors of the of the Gold sequence and Kasami sequence converge to a value of one. In fact, there have been studies that show the asymptotic merit factor of any maximal length shift register sequence is three and the asymptotic merit factor of a twin-prime, Legendre, and Jacobi (modified or not) sequences is six for the optimal shift [21]. So, from the standpoint of the merit factor, the Gold and Kasami sequences seem less appealing than m-sequence or other methods.

A merit factor of around six seems to, arguably, be the highest achievable value. Hodeln and Jensen in [21] even went as far as making the conjecture “that asymptotically the maximum value of the merit factor is and hence that offset Legendre sequences are optimal.” There has been some promising work done to show that merit factors slightly greater than six exists in binary sequences [22] but they are still far from the theoretical largest merit factor, found by Golay in [18], of approximately 12.32.

2.4 Summary

Factors like autocorrelation, cross correlation, and merit factor should be taken into consideration when developing a communication system that requires the use of binary PN sequences. PN sequences are still far from reaching the theoretical maximum that was once proposed. However, as discussed in the next section, SESS binary spreading system does not use PN sequences, it uses the randomness of the data being sent. Therefore each transmission would have different auto correlations and merit factors based of the data that is sent. There it does not apply to these complex factors by which conventional spreading system are judged.

Chapter 3 - Self Encoded Spread Spectrum

3.1 Introduction to Spread Spectrum Systems

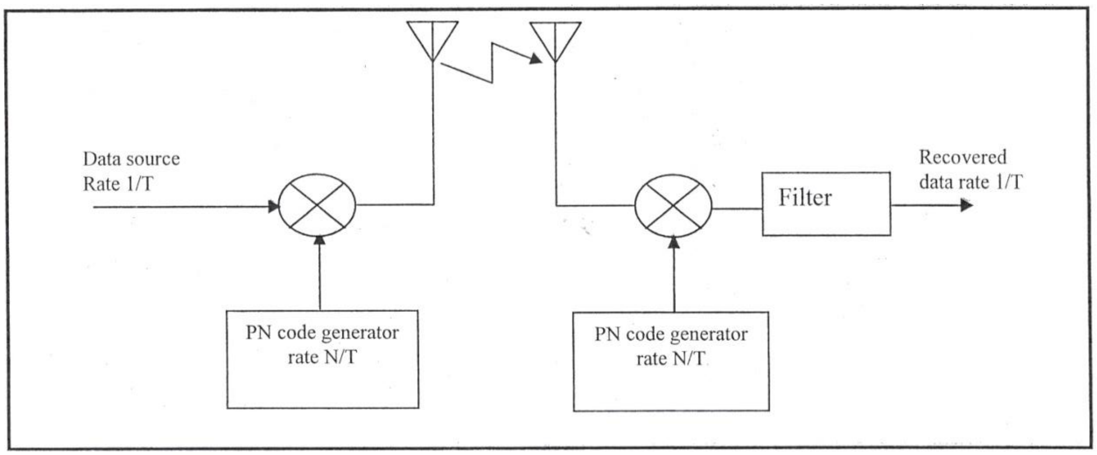

Figure 3 – Conventional Spread Spectrum System

A conventional spread spectrum system, like in Figure 3 above, employs Pseudorandom Noise (PN) code generators (or similar methods like m-sequence, Gold, Walsh codes, etc.) to spread the signal across a wider bandwidth. A visual on how the PN codes are used to spread a signal can be seen in Figure 4 on the next page. They present a practical implementation problem because data recovery by the intended receiver requires prior knowledge of the codes for signal detection. So, the PN codes have to be pre-assigned or be transmitted through the channel to the receiver. This brings up security issues as the PN codes may have pseudo-random properties; they also possess spectral lines and can be duplicated, thus potentially compromising the transmission security.

Self-Encoded Spread Spectrum is very similar to BPSK spread spectrum in that it uses a sequence of bits to encode the data before it is sent over the channel. The initial encoding and decoding sequences are the same as in BPSK, but there is one key difference, which impacts the design, reliability, and performance of the system. The encoding sequence is updated after the transmission of each bit to include the transmitted bit. Each time a bit is transmitted the encoding sequence is shifted and the previously sent bit is added to the sequence. So, in SSES the randomness of the current spreading sequence comes from previous bits transmitted. If the appropriate data compression methods are used to remove any redundant data, the binary data can be modeled as independent and identically distributed Bernoulli random variables. This smoothes out the spectrum of the signals and eliminates the spectral lines associated with PN sequences. As a result of not using PN codes, the detection of the digital data by an unintended receiver is practically impossible, resulting in ideally secure transmissions. The SESS provides a real world implementation of random-coded spread spectrum systems that previously have been thought to be impractical [11].

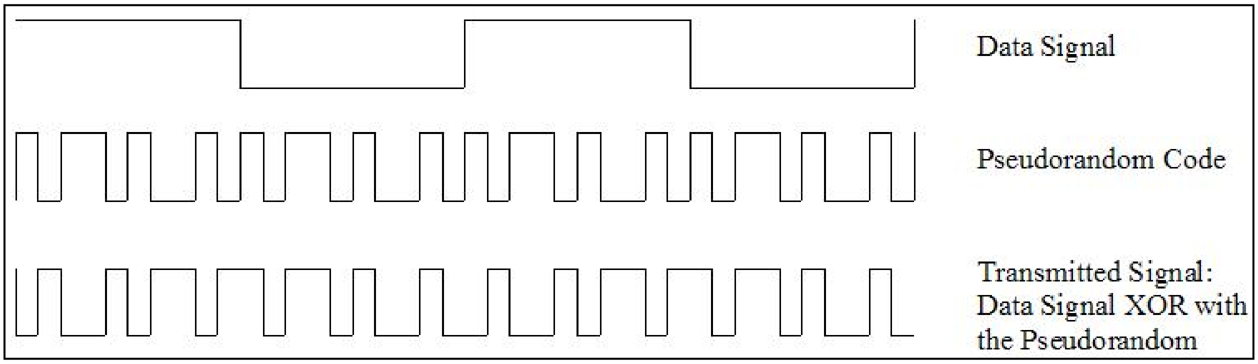

Figure 4 - How Pseudorandom Code is Used

3.2 Self Encoded Spread Spectrum Model

In a Self Encoded Spread Spectrum (SESS) system the traditional PN codes needed for transmitting and receiving are not required. Instead, the spreading codes are generated from the information being transmitted. SESS was first proposed in [23], and a figure of the system can be seen in Figure 5 below. At the transmitter, the delay registers are constantly updated from an N-tap delay of the data, where N is the code length. Thus each bit is modulated at a chip rate of N/T using the past N bits from the shift registers.

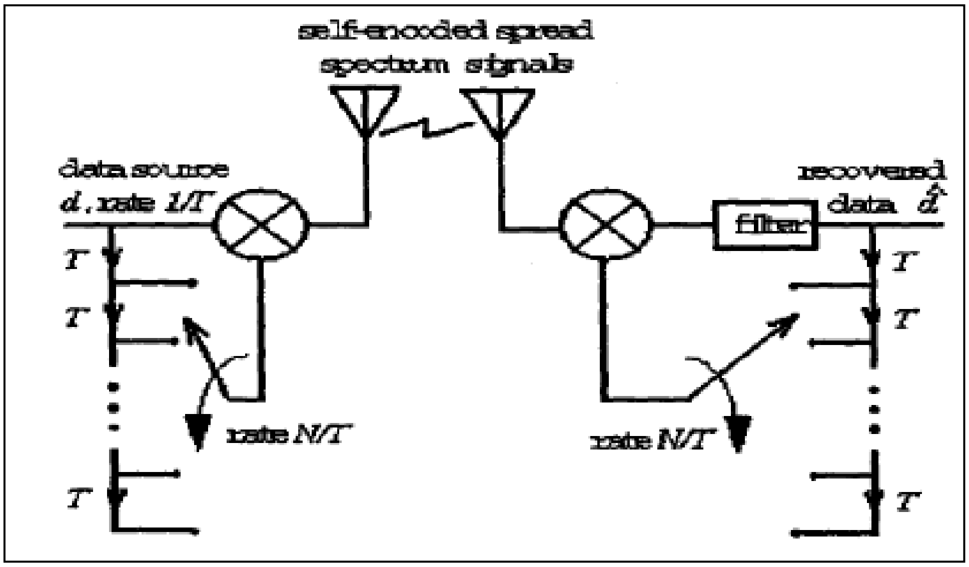

Figure 5 - Encoded Spread Spectrum System [11]

At the receiver, the feedback demodulator performs the reverse operation for symbol recovery by means of a correlation detector. The recovered symbols are fed back to the delay shift registers of N taps, where N is the number of bits in the decoding sequence, to provide an estimate of the spreading sequence required for signal de-spreading. The shift register contents at the transmitter and the receiver should be set initially to be identical.

3.3 SESS Performance Analysis

The issue with SESS is that the performance is impacted at low signal to noise ratios by error propagation. When the receiver detects a bit incorrectly that error is inserted into the decoding sequence and it continues to affect the decoding process until it is shifted out of the sequence N bits later. The signal attenuation depends on the chip length. This means for large N, a chip error would remain in the register longer, but would contribute to a smaller attenuation. Inversely, for a small N value a chip error would rotate out of the register quickly, but would contribute to a larger attenuation. Figure 6 below shows the effects of the sequence length.

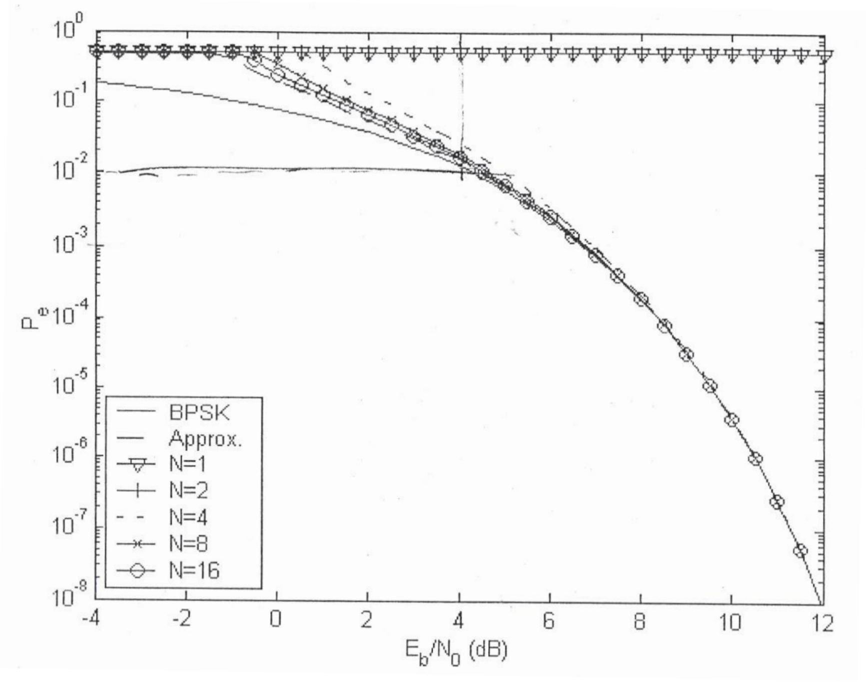

Figure 6 – Effects of Chip Length on SESS

Notice that the greater the sequence length the quicker the SESS system converges to BPSK in an Additive White Gaussian Noise (AWGN) channel. Conversely, for lower N values the BER stays closer to 0.5 longer.

This makes sense when you consider that if there is an error for a small N of 1 or 2, the next bit will be demodulated with the wrong spreading code leaving with either 100 or 50 percent of the wrong bits in the demodulating spreading sequence. This leads to the 50 percent error rate that can been seen in these two values on Figure 5. The chip errors in the receiver registers attenuate the de-spreaded signal strength. This can be regarded as a form of self interference introduced by self encoding. Let X be the random variable denoting the number of chip errors within the receiver register of length N. For l chip errors, the amplitude attenuation can be expressed as:

(7)

$$A|_{X=l} = \left|1 - \frac{2l}{N}\right|$$

Then the conditional probability of error becomes:

(8)

$$P_{e|l} = Q\left(\left(1 - \frac{2l}{N}\right)\sqrt{\frac{2E_b}{N_o}}\right)$$

The probability of error will be greater than 0.5 if the argument of the Q-function is negative; this corresponds to the situation that X > N/2 [24]. This suggests that there exists an undesirable situation caused by error propagation: a sufficient number of chip errors may accumulate in the receiver registers that exceed N/2. So, as N→∞ the probability for error approaches that of a BPSK system which is confirmed in Figure 6 on page 18 and with Equation (8 on the previous page.

The SESS system behaves similarly in a Rayleigh Fading channel as the AWGN channel as seen in Figure 6 below.

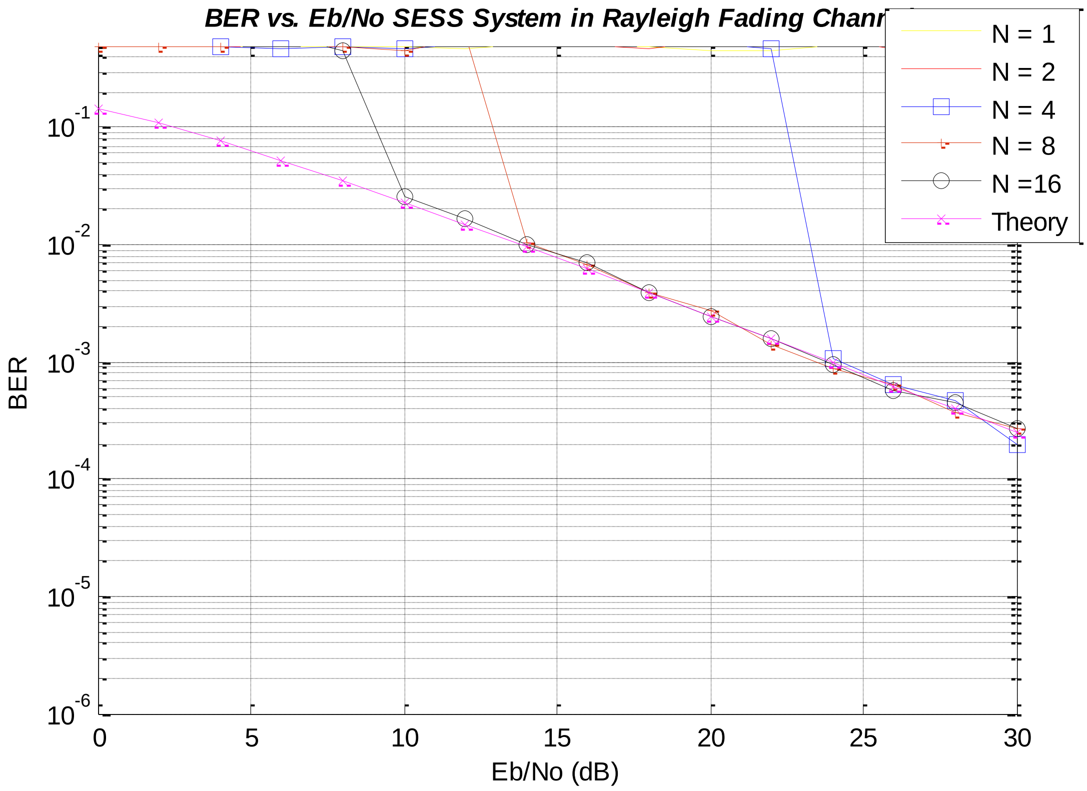

Figure 7 – SESS Performance in Rayleigh Fading Channel

Observe how when N gets bigger, the simulation converges toward the theoretical Rayleigh Fading Channel line. This is the same behavior as is see in the AWGN channel, in that, as N→∞ the probability of error approaches that of a BPSK system in that channel.

3.4 Introduction to Iterative Detectors

Iterative decoding can be described as a decoding technique utilizing a soft-output decoding algorithm that is iterated several times to improve the bit error performance of a coding scheme, with the goal of obtaining true maximum-likelihood decoding, with less decoder complexity [4]. Because there is memory within the SESS modulation, it is a natural candidate for the Maximum Likelihood Sequence Estimation (MLSE) detection based on the Viterbi algorithm. MLSE detection improves the system performance by estimating the sequence of the received signals. However, the number of states in the Viterbi algorithm decoder grows exponentially with the spreading factor, as can be seen in the trellis diagram of SESS when N = 4 in Figure 8 on the next page [24]. An iterative detection scheme can be used instead to reduce the complexity to a linear order of the spreading factor, which achieves performance very close to that of the MLSE detector. The iterative detector is also able to be improved in fading channels by adding a chip-interleaver as discussed in [25].

3.5 Iterative Detector Design

As describe in the previous section, the iterative decoder has a complexity linear to that of the spreading code. The design used requires N+1 storage of the received data bits. The definition of a SESS systems states that the spreading codes are generated from the information being transmitted. If we view the first bit after the encoder (called Bit 1), then we can write the future N+1 bits (N is the length of the spreading code) as the part of the Bit 1 at the receiver (see Figure 9 on the next page).

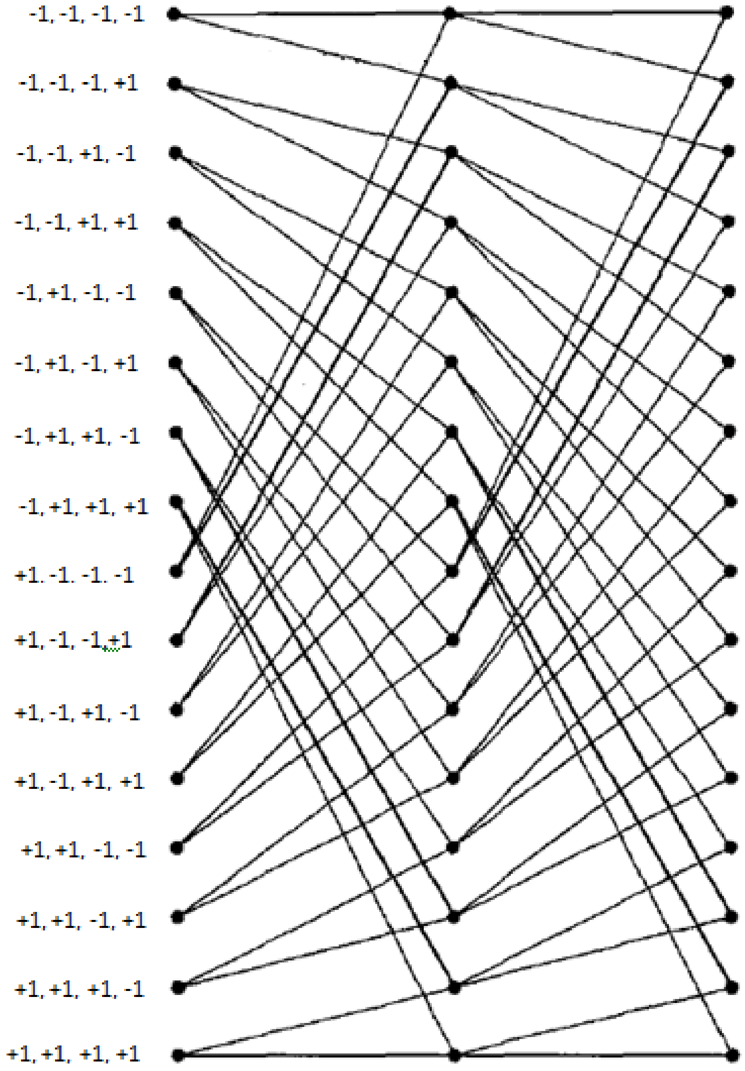

Figure 8 - Trellis Diagram of the Viterbi Algorithm for SESS of N=4 [24]

$$Bit_1 = \left[e_1 e_0\ ,\ \ e_1 e_{-1}\ , \dots ,\ e_1 e_{-N}, e_1 e_{-(N+1)}\right]$$

$$Bit_2 = \left[e_2 e_1\ ,\ \ e_2 e_0\ , \dots ,\ e_2 e_{-(N+1)}, e_2 e_{-(N+2)}\right]$$

$$Bit_3 = \left[e_3 e_2\ ,\ \ e_3 e_1\ , \dots ,\ e_3 e_{-(N+2)}, e_3 e_{-(N+3)}\right]$$

$$\vdots$$

$$Bit_N = [e_N e_{N-1}\ ,\ \ e_N e_{N-2}\ , \dots ,\ e_N e_1, e_N e_0]$$

$$Bit_{N+1} = \left[e_{N+1} e_N\ ,\ \ e_{N+1} e_{N-1}\ , \dots ,\ e_{N+1} e_2, e_{N+1} e_{1)}\right]$$

Figure 9 - Iterative Detector Signals in SESS

From Figure 9 above, it is easy to see that e_1 is not only related to the previous N+1 bits, but also related to N future transmitted bits. This means that N future bits contain information about e_1 , that can be used to help make the final decision on Bit1 should there be excessive channel noise or jamming on Bit1 that would normally cause an error. So by incorporating future transmitted signals together with previous detected bits, we expect to improve the performance over the feedback detector, which only estimates the current bits by correlating with N previous detected bits. How these future transmitted signals are incorporated into the final decision can be seen in Figure 10 on the next page. Also, for a step by step run through of the iterative detector see Appendix A on page 54. It should be noted that additional iterations are able to be run through this detector. Each additional iteration requires N more chips, so if there are M iterations then the storage of roughly N*M transmissions is required. The effect of the number of iterations can be seen in Figure 11 on page 25. Despite the number of iterations, the

BER eventually converges to a max. So, for simplicity reasons the detector in this paper only uses one iteration.

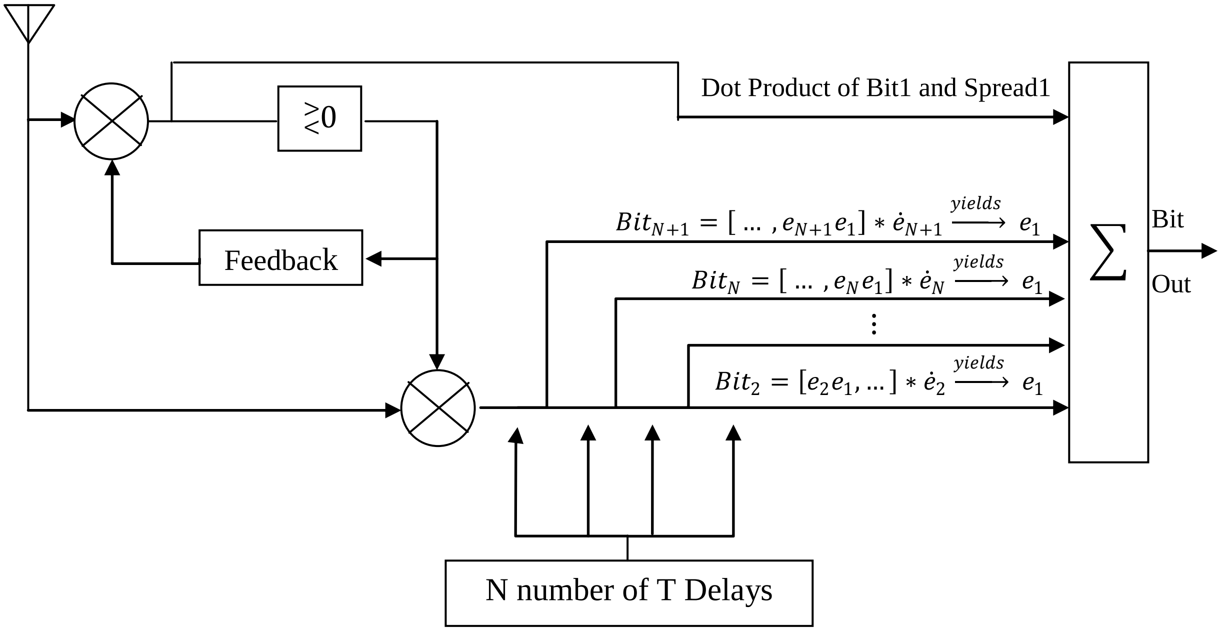
$\gtrless 0$
Feedback
Dot Product of Bit1 and Spread1
$Bit_{N+1} = [\ldots, e_{N+1}e_1] * \dot{e}_{N+1} \xrightarrow{yields} e_1$
$Bit_N = [\ldots, e_N e_1] * \dot{e}_N \xrightarrow{yields} e_1$
$\vdots$
$Bit_2 = [e_2 e_1, \ldots] * \dot{e}_2 \xrightarrow{yields} e_1$
$\sum$
Bit
Out
N number of T Delays

Figure 10 - Iterative Detector Model in SESS

3.6 Iterative Detector Performance

The performance of the iterative detector can be seen in Figure 11 and Figure 12 on the next page. Figure 11 shows that the detector in an AWGN channel performs at a BER of 10^{-4} nets a 3 db gain over the BPSK system. The real surprise comes from the performance in a Rayleigh fading channel, in Figure 12. It shows that at a lower 10^{-3} BER, the detector is able to improve by over the BPSK system by about 15db. In fact, it is almost able to achieve the

AWGN performance of BPSK in the Rayleigh fading channel. Further performance gains in the fading channel can be achieved by implementing a form of interleaving.

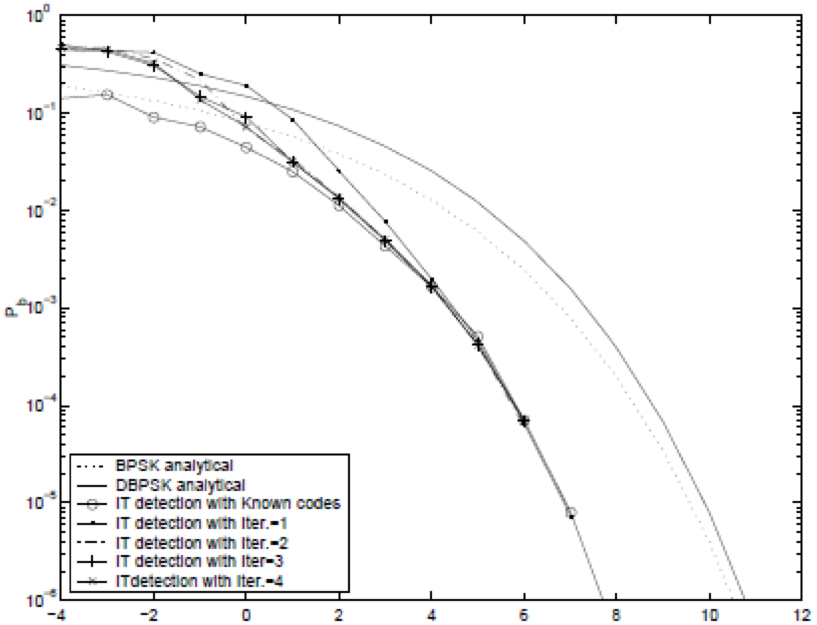

Figure 11 - Effects of Multiple Iterations

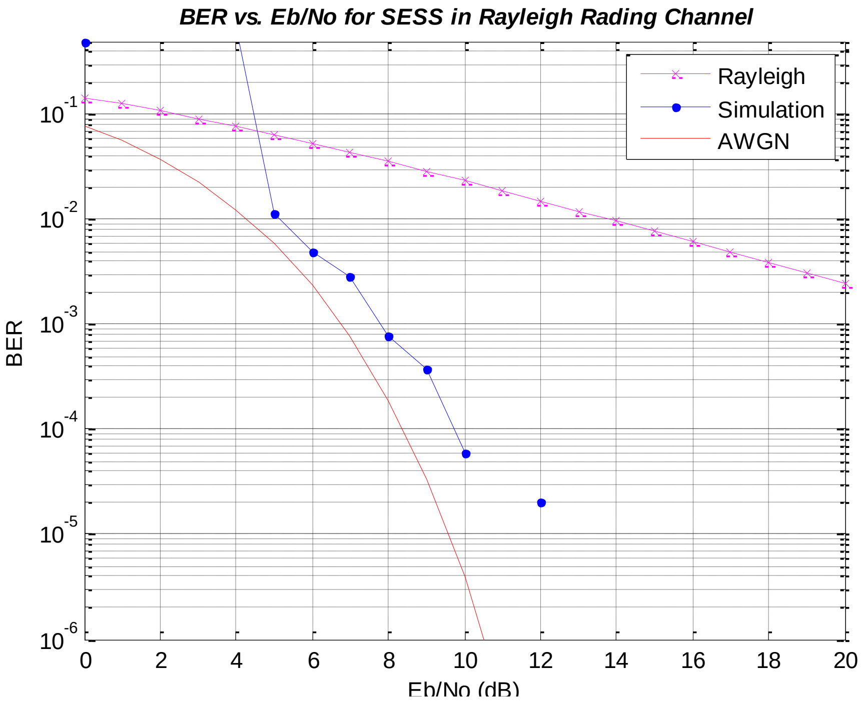

Figure 12 - Iterative Detector Performance in Rayleigh Fading Channel

Chapter 4 – Concept of Jamming

4.1 Introduction to Jamming

There are various different jamming models that can be seen in Figure 13 below. Figure 13a depicts the most benign jammer known as a barrage noise jammer. This type of jammer transmits band limited white Gaussian noise across the power spectrum that covers the same frequency range as the spreading signal. Figure 13c shows a single-tone jammer, which transmits an unmodulated carrier in the spread signal bandwidth. This jammer is quite effective and easy to implement in DSSS systems, however requires the tone to be placed at the center of the spread signal bandwidth to achieve maximum effectiveness.

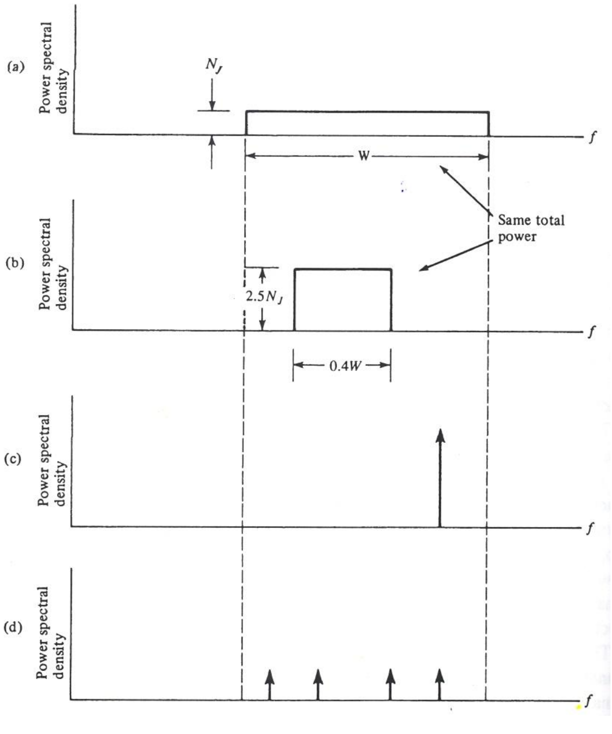

Figure 13- Jamming Models [26]

Figure 13d is a multi-tone jammer, which poses a better strategy against frequency hoping systems than the single-tone jamming. This jammer selects the number of tones so that the optimum degradation occurs when the spread signal hops to a tone frequency. The type of jammer that is used in this paper is known as the partial-band jammer for frequency hoping systems, and in time varying systems it is called a pulsed noise jammer. It can be seen in Figure 13b, and is talked about more in depth in the next section.

4.2 Pulsed Noise Jamming Introduction

Instead of just continuously jamming a communication channel, a pulsed noise jammer (PNJ) can be used to jam the channel at chosen times with a greater power. This proves to be a more effective way of jamming a spread spectrum system and is often used in electronic counter measure operations. A PNJ can be defined as a jammer that turns “on” with just sufficient power to degrade spread spectrum system performance significantly, but does not totally annihilate system performance when “on”. The PNJ transmits a pulsed band-limited white Gaussian noise signal whose power spectral density (PSD) just covers the spread spectrum system’s bandwidth (W). The duty factor for the jammer (ρ) is the fraction of time during which the jammer is “on”. When the jammer is “on,” the one-sided received jammer power spectral density can be expressed by $\dot{N}_j = J/\rho W$, where J is the jammer power, ρ is jammer duty cycle, and W is the bandwidth [26]. So, during the time that the jammer is “on”, the jammer voltage is $1/\sqrt{\rho}$ and when the jammer is “off” the voltage is zero. PNJ in these simulations assumes a jammer power amplifier is average-power limited rather than peak-power limited to simplify calculations and implementation. Note that in practice, the peak power would have limitations; this would affect the lower ρ in the simulations.

4.3 Pulsed Noised Jamming BER Analysis for DSSS

In coherent systems direct-sequence spread spectrum (DSSS) systems, the bit error rate is:

(9)

$$PE_b = Q\left(\sqrt{\frac{2E_b}{N_o}}\right)$$

Where PE_b is the probability for error or BER, Q( ) is the Q-function, E_b is the energy of each bit, and N_o is the one-sided noise spectral density. When the PNJ is added to the system the equation becomes the following:

(10)

$$\overline{PE_b} = (1-\rho)Q\left(\sqrt{\frac{2E_b}{N_o}}\right) + \rho Q\left(\sqrt{\frac{2E_b}{N_o + N_j/\rho}}\right)$$

Where (1 –ρ) represents the time that the jammer is “off”. If it is assumed that the noise is negligible with respect to the jamming level, this equation can be simplified to:

(11)

$$\overline{PE_b} \approx \rho Q\left(\sqrt{\frac{2E_b}{N_j/\rho}}\right) \; or\ can\ we\ rewritten\ as, \qquad \overline{PE_b} \approx \rho Q\left(\sqrt{\frac{2\rho PW}{JR_b}}\right)$$

It can be rewritten because $E_b = P / R_b$ where R_b= bit rate and $N_j = J/W$. This can be tested by running a simulation and calculating the BER for various ρ against E_b/N_j. This can be seen in Figure 14 below:

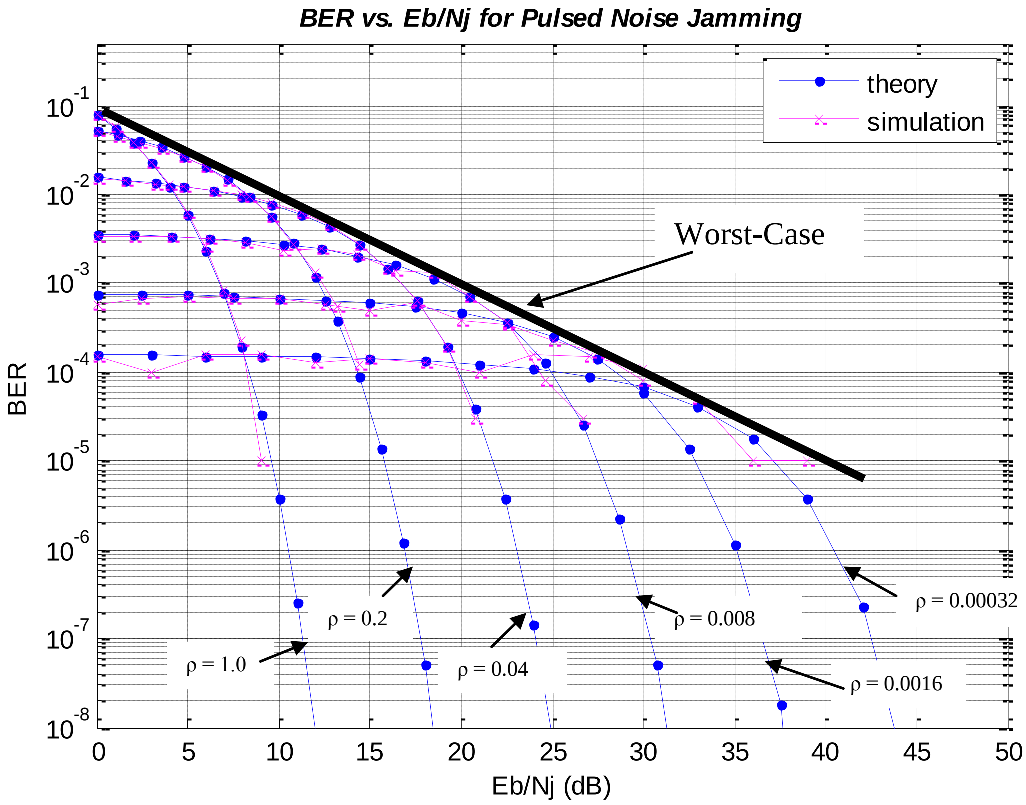

Figure 14 – Worst-Case Jamming for Pulsed Noise in DSSS

The worst-case can be approximated by the think black line is tangent to the curves of the various ρ values. Figure 14 shows that the optimal jamming duty cycle is dependent on the E_b/N_j of the transmitted signal. Assuming the E_b/N_j is at least 0.709, the worst-case line can be approximated by the following equation (see [26] for math):

(12)

$$\overline{(P_b)}_{max} \approx \frac{0.083}{(P/J)\,(W/R)}$$

4.4 Summary

This chapter introduced the various jamming models and established that worst-case jamming in SESS is pulsed noise. The probability of error of a DSSS system is used to derive the worst-case jamming equations for various ρ. By confirming these numeric results with the simulation, the results of the simulation can be used to generate a worst-case jamming line. From the worst-case jamming line, the equation for worst-case jamming (Equation (12) can be found from the inverse linear relationship of the envelope of the curves.

Chapter 5 – Worst-Case Jamming in SESS

5.1 Introduction

SESS has been shown to achieve same performance as DSSS in AWGN and Rayleigh fading channels. This chapter sets out to show through mathematical analysis and simulation that SESS remains the same as DSSS in worst-case jamming.

5.2 Analysis

As stated in Chapter 3.3 in Equation (8), the probability for error in a SESS system can be expressed as: $P_{e|l} = Q\left(\left(1 - \frac{2l}{N}\right)\sqrt{\frac{2E_b}{N_o}}\right)$. Where l refers to the number of errors in the receiver code, and N is the chip length. In order to find the worst case jamming, this probability for error must be used, in a similar fashion to DSSS, to the PNJ model. Equation 8 shows the probability of error of SESS in the pulse noise jamming model.

(13)

$$\overline{\overline{P_{e|l}E_b}} = (1-\rho)Q\left(\left(1 - \frac{2l}{N}\right)\sqrt{\frac{2E_b}{N_o}}\right) + \quad \rho Q\left(\left(1 - \frac{2l}{N}\right)\sqrt{\frac{2E_b}{N_o + N_j/\rho}}\right)$$

In the DSSS model, the assumption was made that the probability of error is dominated by the jamming. By making the same assumptions in the SESS model, the (1 – ρ) Q-function can be ignored as it represents the noise of the channel when the jammer is off, and getting rid of the conditional probability, the theoretical probability of error for the SESS becomes:

(14)

$$P_b = \rho Q\left((1-2P_b)\sqrt{\frac{2E_b}{N_j/\rho}}\right)$$

By using the upper bound of the Q-function.

(15)

$$P_b \leq \frac{\rho}{Y\sqrt{4\pi\rho X}} * e^{(-\rho X Y^2)}$$

Where X = Eb/ Nj and Y=(1-2*Pb). By setting the first derivative of equation (15) with respect to ρ equal to zero and solving for ρ, the worst-case jamming line can be found.

(16)

$$0 = \frac{e^{-pXY^2}}{Y\sqrt{4\pi\rho X}} - \frac{1}{2}\frac{\rho e^{-\rho XY^2}}{Y\sqrt{4\pi\rho X}} - \frac{-X\,\rho e^{-\rho x Y^2}}{Y\sqrt{4\pi\rho X}} \xrightarrow{yields} \rho = \frac{1}{2XY^2}$$

By substituting this back into the equation (14) and assuming a large N:

(17)

$$P_b = \frac{Q(1)}{2XY^2} \;\; or \;\; P_b(1-2P_b)^2 = \frac{Q(1)}{2E_b/N_j} \cong \frac{0.083}{(P/J)\,(W/R)}$$

It was shown in [24], and explained earlier in section 3.3, that as N approaches infinity Y approaches one and drops out and it becomes the same as Equation (12). Also note that under normal conditions Pb is very small and leads to Pb = Q(1)/(2Eb/Nj).

5.3 Results

Figure 15 on the next page shows that the optimal jamming duty cycle is dependent on the E_b/N_j of the transmitted signal. In Figure 15 on the next page the worst-case is approximated by the think black line, which is the tangent to the curves of the various ρ values. By recognizing the inverse linear relationship of the worst-case jamming line, it can be approximated by trail and error. When this approximation method is applied to the worst-case jamming to Figure 15 on the next page is the result:

(18**)**

$$\frac{0.083}{(P/J)\,(W/R)}$$

Notice that equation of the worst-case jamming line in equation **(**18**)** and theoretical line in equation (17) are the same. Figure 15 also matches up with the DSSS figure in the previous chapter. The effects of error propagation in the SESS explain the deviation from the worst case jamming line when the SNR values are below 2.5 db.

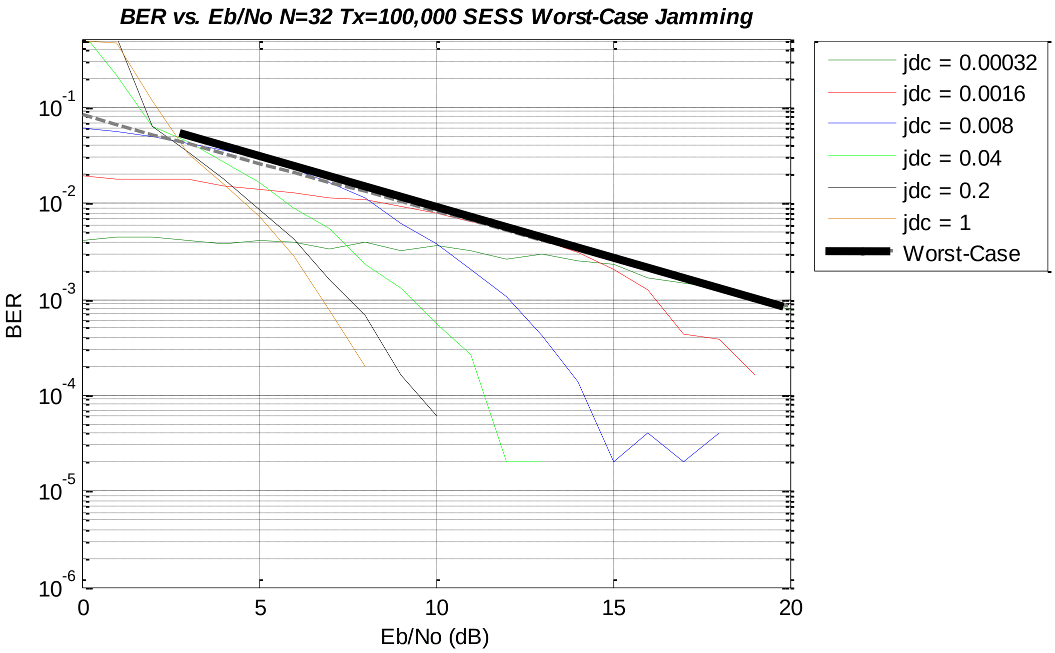

Figure 15 - SESS Worst-Case Jamming

Chapter 6 – Worst-Case Jamming in SESS w/ Iterative Detector

6.1 Introduction

In chapter three, the performance of the SESS in a noise or fading channel was improved by adding an iterative detector. In this chapter, an iterative detector was adding to the worst-case jamming of SESS proposed in chapter four in hopes to achieve similar performance gains.

6.2 Analysis

In the previous chapter, the worst-case jamming was found by taking the differentiating Equation (16) with respect to ρ and setting the result equal to zero. The derivative of the Q-function is not easily calculated when an iterative detector is attached to a SESS system. The worst-case can be calculated from the same method used to find Equation **(**18**)** on page 33, which is, using envelope of the family of curves to represent the worst-case for the probability of error.

6.3 Results

Taking the model used to generate Figure 15 on the previous page, we can apply SESS with the iterative detector to help achieve an improvement on the worst-case scenario. Figure 16 below, shows the effects of SESS with an iterative detector in the worst-case jamming conditions. This simulation assumes that the initial spreading sequences are synchronized and that noise of the channel is negligible.

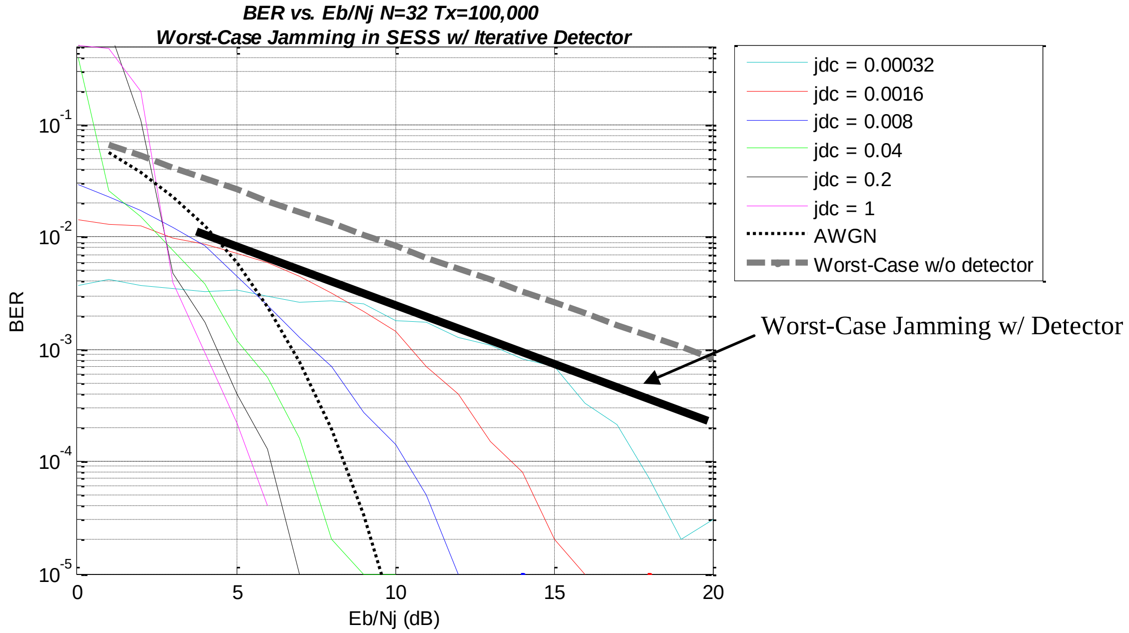

Figure 16 - Worst-Case Jamming SESS Simulation

As the same in DSSS, the envelope of the family of curves exhibits an inverse linear relation between Pb_{max} and the Eb/Nj can be seen. This makes Equation (19), the worst-case jamming in SESS with iterative detector, easily calculated by finding of the equation of the line in Figure 16.

(19)

$$\overline{(P_b)}_{max} \approx \frac{0.022}{(P/\,J)\;(W/\,R)}$$

Notice the performance in relation to the AWGN line in Figure 16. With the jamming duty cycle at 100 percent, the performance is the same as in an AWGN channel (See Figure 10 on page 24). The improvement of the system can be seen when comparing Figure 16 against Figure 14. The SESS with the iterative detector improves the worst-case jamming by 6 db at a BER of 10^{-3} (difference can also be seen in Equations 12 and Equation 13).

6.3 Discussion

The Rayleigh fading of SESS and the worst-case jamming are very close in terms of channel performance. The 6 db improvement of the iterative detector in worst-case jamming still lacks the 15 db improvement that was seen in the Rayleigh fading channel (Figure 12 on page 25). At a first glance it appears that since the PNJ uses band-limited white Gaussian noise, the performance of the iterative in the presence mocks that of the AWGN as they both use white Gaussian noise. However, by investigating the way the hard bit decision is made in the SESS systems can give insight to the possible difference in the BER performance.

In a DSSS system, the data is multiplied with the spreading sequence to effectively spread the signal over a larger bandwidth. To retrieve the data on the receiver side, the dot product, or summing the products of their respective components (For example if $\vec{u}= (a, b)$ and $\vec{v}= (c, d)$, the dot product is represented as: $\vec{u} \cdot \vec{v} = \ a * c + b * d$), is taken from the received signal and the spreading code. An example of this can be found by letting $\vec{x}$ = -1 stand for the bit to be transmitted and letting $\vec{y} = (1, -1,1,1)$ represent the spreading sequence. In order to spread the transmitted bit it is multiplied by the spreading sequence. So the transmitted signal through the communication channel becomes $\vec{x} * \vec{y} = (-1, 1, -1, -1)$. Assuming the spreading sequence at the receiver is the same as the transmitter, the dot product of the transmitted data and the spreading sequence can be taken $(-1, 1, -1, -1) \cdot \vec{y} = -1 * 1 + 1 * -1 + -1 * 1 + -1 * 1 = -4$. . The next step is to make a hard decision based on the dot product of the received signal and spread sequence. Since the dot product equals -4 and -4 is less than zero, the despread bit is -1, which matches the original bit we transmitted. The iterative detector model makes decisions in a similar matter, however, in addition the N components that are used in the dot product, it incorporates N more components from the SESS signal (as explained in section 2.4-2.6).

The difference between the channel performance of Rayleigh fading and worst-case channel when an iterative detector is present, can most likely be attributed to the difference in the channel that lead to the hard decision that is made. In the jamming channel for low ρ values, a single chip value may be jammed with a very large amount. This single chip is weighted on the hard bit decision so much that it dominates the decision. For instance, in the example in the previous paragraph, if one chip of the data at the receiver is heavily jammed the dot product can be represented as (where the 20 represents the jammed chip, as it changed from -1 to 20):

$$(-1, 1, -1, 20) \cdot \vec{y} = -1 * 1 + 1 * -1 + -1 * 1 + 20 * 1 = 17$$

This shows how the worst-case jamming can effect on a spreading system. Adding an iterative detector is certainly going to help by adding more chips to the hard decision, and it does by 6 db, but it does not come close the 15 db in a Rayleigh fading channel. It would appear that a greater the N, the greater the chance that the jamming will not affect the system. However, it has been shown that the length of the N, does not improve performance of the system without an iterative detectors, it just helps it converge to a max faster (see Figure 6 on page 18). By varying the N from 2 to up the 32 used in Figure 16 on page 36, the only change found was the lower SNR at values when they are beginning to that if N was very large. The likely reason that the increase N does not help the performance is that it also increases the chances that one of the chips in the sequence is going be jammed.

6.4 Summary

This chapter examined the worst-case jamming of a SESS system with an iterative detector. It achieved a performance gain of 6 db over the SESS system without the detector. The worst-case jamming line of a SESS without detector is close to that of Rayleigh fading, so the 6 db improvement of the iterative detector falls short of the 15 db it adds to Rayleigh fading. The difference in the performance can likely be attributed to the differences in jamming and fading models. The effects of jamming can be seen in the hard decision made on the receiver side. To help combat the worst-case jamming that occurs at low ρ, a method needs to be investigated that stops the jammed chip from heavily contributing to the final bit decision.

Chapter 7 – Jamming with Noise and Fading Channels

7.1 Introduction

This chapter investigates a sub-optimal method to improve the SESS under the jamming conditions. The method makes a soft decision on the chip values rather than the bit values like the systems in the pervious chapters. The only problem with this system is that in previous simulations, the noise of the channel was assumed to be irrelivate. By adding the noise and fading into the jamming channel the flaws in the new sub-optimal method are exposed.

7.2 Analysis

By looking at the way the PNJ works, we can develop methods to defeat the jammer. The method that proposed in this section makes decisions on each chip rather than each bit. Looking at the jammer at the lower jamming duty cycles, it is effective because it is jamming one chip a large amount, which then gets factored into the decision of the bit. If a decision is made at the chip level, then each chip is weighed evenly at the bit level. This means that when the jammer heavily jams one chip, the chip decision will be made and it will only effect the bit decision by the change on that one chip. In other words, only 1 of the N chips is going to be wrong. The final bit decision has a better chance of correcting the jammed chip since the jammed chip contributes a value of 1/N to the final bit decision, rather than the actual jammed amount. So, the chip values get changed from the receiver voltage to either -1 or 1. Normally this would happen at the bit level, however this lets the jamming have a greater effect on the system. Figure 17 shows how the decisions placed on the chip can improve the worst-case jamming of the SESS system with iterative detector when the noise of the channel is negligible.

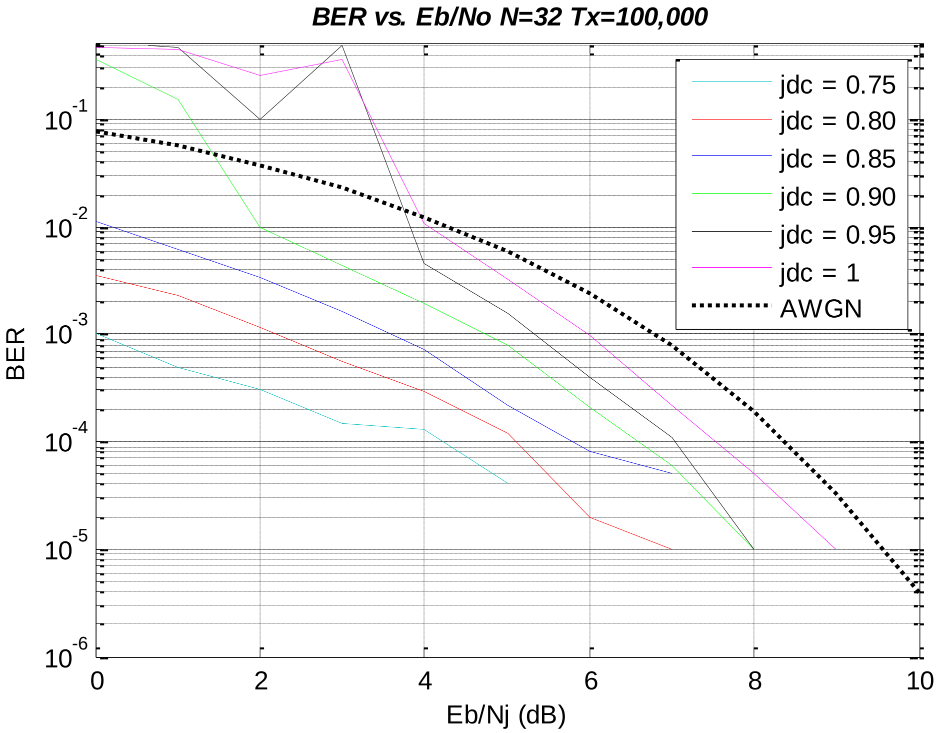

Figure 17 - Worst-Case Jamming SESS Chip Decision

It should be noted that the chip decision improves the worst-case jamming, and that the worst-case is when the jamming duty cycle is 100 percent (every bit is jammed or ρ =1). As the jamming duty cycle gets lower, the BER gets better, proving that it works. In fact, it improves the worst-case jamming by 6 db at a BER of 10^{-3} over the bit decision, which is 12 db over DSSS.

The success of the chip decision in the worst-case scenario can be attributed to the way it handles jamming at low ρ values. At the lower ρ values, chips have a lower chance of being jammed. When the decision is made at the chip level it weighs out effects of higher power jamming, as the jammed chip only attributes 1/N of the final hard decision (1/2N with iterative detector). Thus, at the lowest level of ρ, the majority of the data being sent is not jammed and sent through a noiseless channel. The noiseless channel makes it easy for the chip decision detector to recover from errors at lower ρ. This being the case, the jamming over the channel no longer dominates the noise of the channel. This reveals the need to add noise or fading back to the channel during the jamming simulations.

7.3 Results

There remain problems with the chip decision approach in the assumptions made of the simulations used thus far in the paper. The first problem is that the chip decision is actually lower performance under the standard AWGN channel and in a Rayleigh fading channel (can be seen in Figure 16 on the next page)

With ρ equal to one, chip decision is about 2.5 db worse than in the bit decision on both the AWGN and Rayleigh fading channels. Figure 17 shows that under jamming the chip decision does better, but Figure 17 shows in noise channels without jamming bit decision is better. This brings rise to the second problem; when jamming the assumption is that the noise of the channel is negligible.

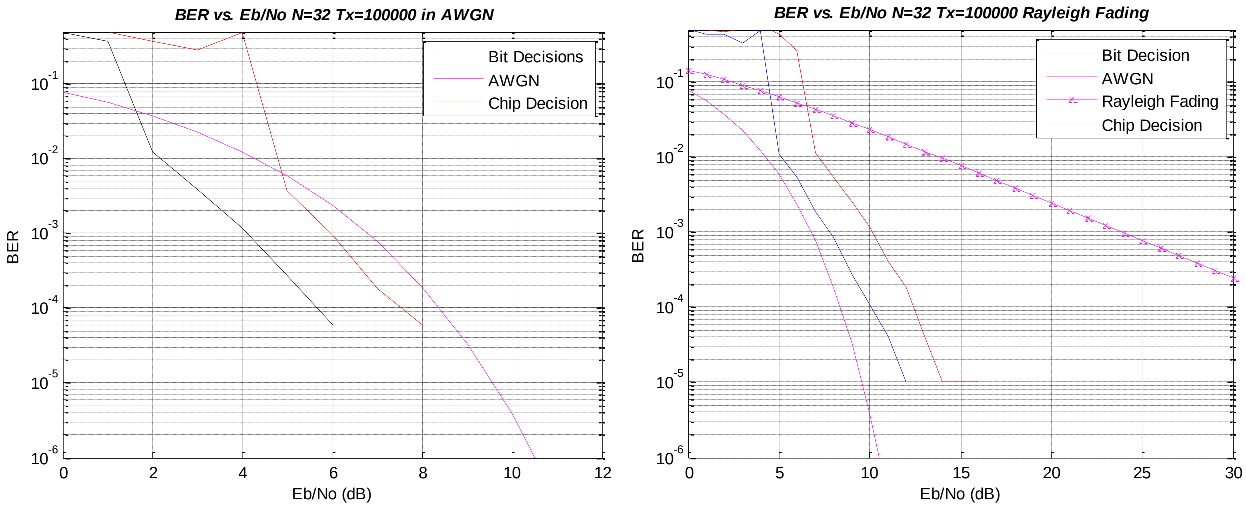

Figure 18 - SESS Chip Decision Vs Bit Decision (AWGN on right, Rayleigh fading on Left)

This performance hit in chip decision can be described by the model being fundamentally suboptimal and goes against the of correlation detection. However, it does offer a potential solution to improve the worst-case jamming.

If bit decision works better in noise channels and chip decisions work better in the worst-case jamming scenario, then what if there is a noise channel with jamming? Figure 19 and Figure 20 on the next page show the effects of jamming in AWGN and Rayleigh fading channels with bit decisions and chip decision respectively.

From these figures it can be seen that there are a few things about the differences between chip and bit decisions. The first is that the ρ in an AWGN channel doesn't affect the chip decision. It suffers from the 2.5 db worse performance than the bit decision when the ρ is one. The second is that it becomes advantageous to use the chip decision when ρ is less than 0.08. The results in the Rayleigh channel show similar results, that right around a ρ of 0.08 chip decisions become the better system. The channel noise was ignored in previous simulations because the jamming dominated the noise. This appears to remain the case for the higher SNR as the values for the worst-case jamming remain close to the same. The only part of the graphs where the channel noise or fading had any effect was the SNR values from zero to ten.

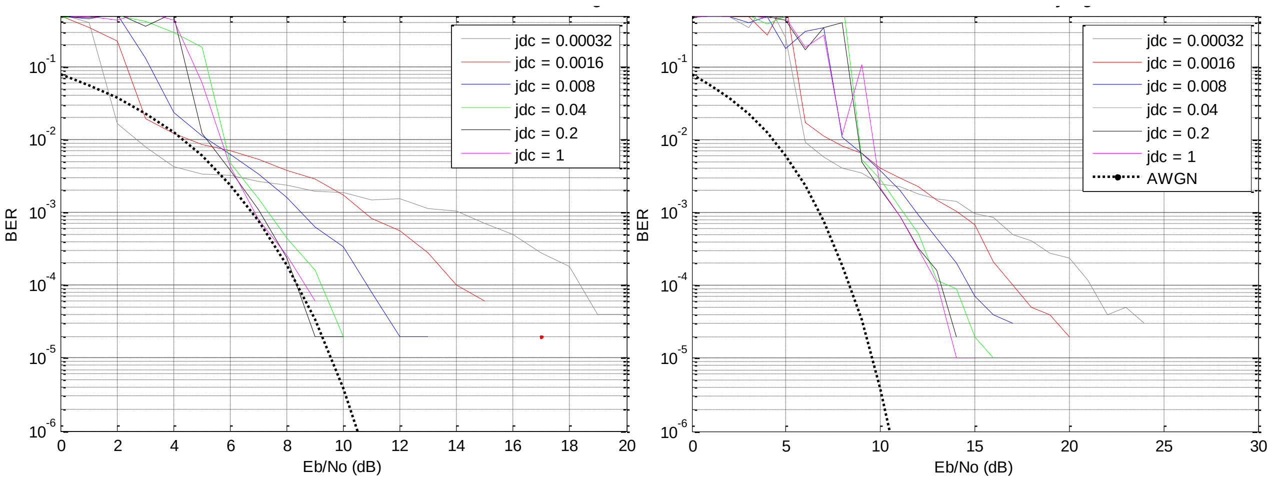

Figure 19 - Bit Decision Jamming in AWGN (left) and Rayleigh Fading Channels (Right)

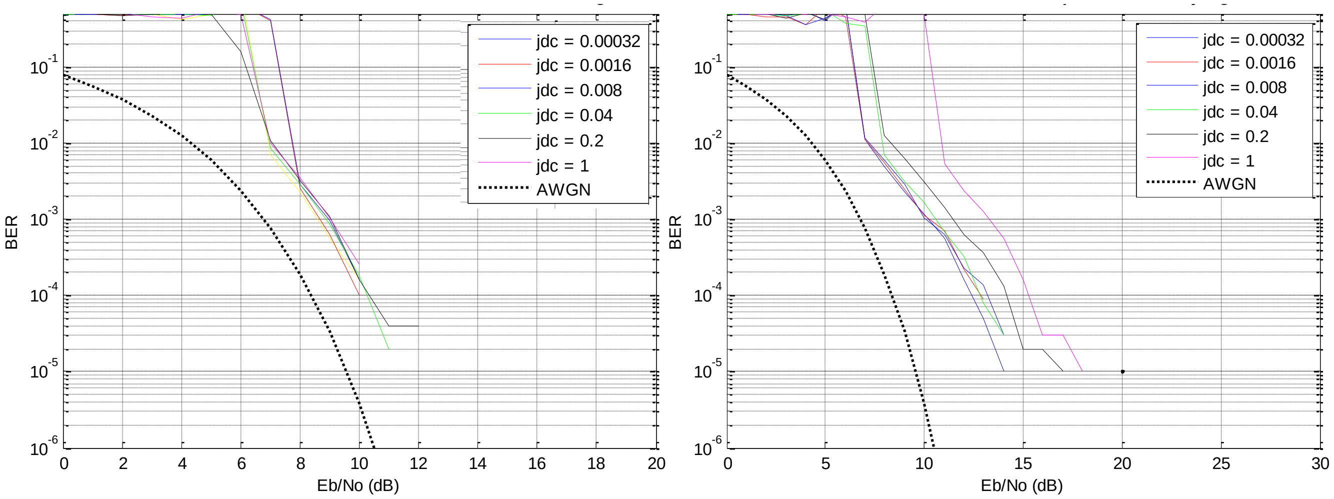

Figure 20 - Chip Decision Jamming in AWGN (left) and Rayleigh Fading Channels (Right)

Figure 19 and Figure 20 shows that chip decision does improve the worst-case scenario by as much as 8 db over bit decision (at BER of 10^{-4}) in a noise or fading channel with jamming. However, just like as the worst-case jamming depends on prior knowledge of the system to achieve the best jamming, the decision to use chip based decisions requires knowledge of the jamming. With knowledge of the ρ, a decision could be made to switch to chip based decision to help improve performance.

The final problem affects both systems and it is that in the simulations we assumed that the jammer had a limited average power and unlimited peak-power. With the ρ values low, the peak-power is affected because the voltage of the jammer is increased by $1/\sqrt{\rho}$. The voltage cannot be expected to continuously increase at this rate, as there will be a physical limit to the system and it would eventually begin to decline. It is especially bad for the chip decision, as it has been shown that the ρ values where the chip decision becomes most valuable is at the lower values.

7.4 Discussion

Perhaps, using a hybrid system of both bit and chip decision could help alleviate the problems that the bit decision system has at low duty cycles, while giving the performance of the bit decision at higher duty cycles. More performance could potentially utilized from the chip base decisions by adding weights to each chip decision. The way the system in the simulation runs is it changes the chips values of -1 or 1. By adjusting the values based on the original voltages, it could bridge the gap in performance between the chip and bit decision system. This sort of method that uses these weight adjustments that could incorporate the iterative detector can be looked as an iterative least mean square algorithm. In a least mean square algorithm the equalizer weighs by observing the error between the desired pulse shape and the observed pulse shape at the equalizer output. This error is based on the observing the sampling instants and then processing the error to determine the direction that the chip weights should be changed to obtain the optimum values [27]. Figure 21 below shows a least mean square algorithm for a baseband adaptive equalizer, that uses a line delay filter similar to that used on the iterative detector in the SESS system. It assumes that some form of pulse shaping has been utilized in the design, so it could be used in the SESS system. The equalizer weight adjustments may be achieved by observing the error between the desired pulse shape and the observed pulse shape at the equalizer output. There is many different ways that an error can be defined, and there are many papers that discuss the possibilities.

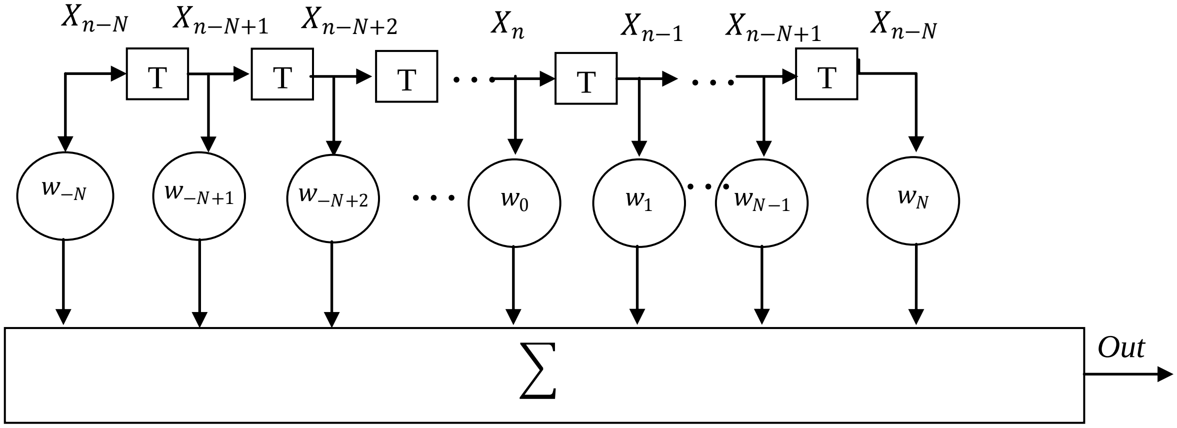

Figure 21 - Least Mean Square with Baseband Adaptive Equalization [4]

There has been work done with a similar weighting system in RAKE receivers in [28]. It concludes that by the maximum-likelihood RAKE receiver limits the effect of pulse jamming by weighting each bit by the inverse of the variance. This requires the variance to be measured on a bit by bit basis, which significantly complicates the receiver. This is similar to the chip decision used in SESS because a rake receiver uses a sequence of soft decision receiver outputs to make a bit decision. If any of these soft decision receiver outputs have a large variance, much like the bit decision used in SESS, it will greatly affect the output of the bit.

7.5 Summary

From this chapter it can be seen that noise or fading in a channel with jamming affects the BER at lower SNR greatly. The effect at high SNR is less seen, which skews the worst-case jamming, making it no longer an inverse relationship with the envelope of the curves. Chip decision is a sub-optimal way that can be used to help fight the worst-case jamming conditions. It does not always outperform bit decision, as in many conditions the bit decision remains the better system. The worst-case jamming depends on prior knowledge of the system, the decision to use chip based decisions requires knowledge of the jamming. With knowledge of the duty cycle, a decision could be made to switch to chip based decision to help improve performance. Another method employs using algorithms that readjust the weight of the chips at the equalizer, known as least mean square algorithms. If the peak power of the jammer is limited it is going to affect the jamming performance at low ρ in both chip and bit decision. The bottom line is that without knowing the specifics of a jamming channel (i.e. capabilities of jammer, current jammer state, etc.), it is very difficult to determine which model is superior.

Chapter 8 – Conclusions / Future Work

Self-Encode Spread Spectrum eliminates the need to synchronize m-sequences between transmitter and receiver, because it uses spreading codes generated by the transmitted signal. This also eliminates the security flaws associated with m-sequences, thus making the channel more secure. This paper discussed the performance of SESS system with an iterative detector in AWGN and Rayleigh fading channels. Introduced was the worst-case pulsed-noise jamming to a SESS, as well as, a potential solution for the jamming. The performance of the SESS system with the iterative detector outperformed the standard DSSS system by 6 db at a BER of 10^{-3}. This performance is great, but fails to come close to the performance of improvement seen in the Rayleigh fading channel. In an attempt to increase the performance of the worst-case jamming, modification of the SESS system was made. By making decisions based on the chips instead of the bit, the worst case performance can be increased dramatically. However, by looking into the performance of chip based decisions in noise and fading channels, then introducing noise into the jamming simulation made chip decisions less desirable. When taking physical limitations of jamming systems into consideration, the chip decision based system performance suffers more, but under absolute worst-case jamming conditions it could still offer improvement over the bit based decisions. Future work needs to be done to explain why the performance of the iterative detector in jamming differs from that in a fading channel. More work is also needed to improve the worst-case jamming in SESS with iterative detector by a defeating jamming by implementing an advanced form chip decision or other method.

References

1. **Scholtz, Robert A.** *The Origins of Spread-spectrum Communications.* No. 5, s.l. : IEEE Transactions on Communications, May 1982, Vols. Vol.com-30.

2. **Price, R.** *Further Notes and Anecdotes of Spread-Spectrum Origins.* January 1983, IEEE Trans. on Communications, Vol. COM-31, No. 1.

3. **Scholtz, Robert A.** *The evolution of spread-spectrum multiple-access communications* . s.l. : IEEE ISSSTA '94, 1994, Vol. Spread Spectrum Techniques and Applications. 10.1109/ISSSTA.1994.379623.

4. **Holmes, Jack K.** *Spread Spectrum Systems for GNSS and Wireless Communications.* London : Artech House, INC., 2007. 978-1-59693-083-4.

5. **Lee, Steve.** *Spread Spectrum CDMA IS-95 and IS-2000 for RF Communications.* s.l. : McGraw-Hill, 2002. 0-07-140671-9.

6. **Bacon, Peter.** *CDMA-myth or reality*. s.l. : Local Loop Fixed Radio Access, IEE Collequium, December 1995.

7. **Ipatov, Valery P.** *Spread Spectrum and CDMA Principles and Applications.* s.l. : John Wiley & Sons, Ltd, 2005. 0-470-09178-9.

8. **Tiedermann, E.G., Jr.** *CDMA for cellular and PCS.* s.l. : Electro/94 International Conference Proceedings. 10.1109/ELECTR.1994.472696 .

9. **Lathi, B. P.** *Modern Digital and Analog Communciation Sysytems.* New York : Oxford University Press, 1998.

10. **The Institute for Telecommunication Sciences.** Glossary of Telecommunication Terms. *Federal Standard 1037C.* [Online] 1996. [Cited: Novemeber 19, 2009.] http://www.its.bldrdoc.gov/fs-1037/fs-1037c.htm.

11. **Nguyen, L.** *Self-Encoded Spread Spectrum and Multiple Acess Communications.* New Jersey, USA : IEEE 6th Int. Symp. on Spread-Spectrum Tech. & Appli.,, 2000.

12. **Proaskis, John G.** *Digital Communications 4th Edition.* s.l. : McGraw-Hill Higher Education, 2001. 0-07-118183-0.

13. **Pursley, Michael B.** *Introduction to Digital Communications.* Upper Saddle River, NJ : Pearson Prentice Hall, 2005. 0-201-18493-1.

14. **Gold, R.** Optimal Binary Seqeunces for Spread Spectrum Multiplexing (Corresp.). *Information Thoery, IEEE Transactions on.* 1967, Vol. 13, 4.

15. **Welch, L.** Lower Bounds on the Maximum Cross Correlation of Signals. *Information Theory, IEEE Transactions on.* 1974, Vol. 20, 3.

16. **Burchrer, R. Michael.** *Code Division Multiple Access (CDMA).* s.l. : Morgan & Claypool, 2006. 1598290401.

17. **Kasami, T.** *Weight Distribution Formula for Some Class of Cyclic Codes.* Univ. of Illinois, Urbana : Report of Coordinated Science Lab., 1966.

18. **Golay, M.** *The Merit Factor of Long Low Autocorrelation Binary Sequences (Corresp.).* 3, s.l. : Information Theory, IEEE Transactions on, May 1982, Vol. 28. 0018-9448.

19. **N. Balaji, K. Subba Rao, M. Srinivasa Rao.** *FPGA Implementation of Ternary Pulse Compression Sequences with Superior Merit Factors.* 2, s.l. : International Journal of Circuits, Systems, and Signal Processing, 2009, Vol. 3.

20. **Toshitaka Date, Xuping Zhou, Ikuo Oka, and Chikato Fujiwara.** Evaluation of Pseudonoise Sequence in Time Spread Pulse Position Modulation / Code Division Multiple Access Systems. *Electronics and Communications in Japan (Part 1: Communications).* 1996, Vol. 79, 10.

21. **Jensen, J.M., Jensen, H.E. and Hoholdt, T.** The Merit Factor of Binary Seqeunces Related to Difference Sets. *Information Theory, IEEE Transactions on.* 1991, Vol. 37, 3.

22. **Kristiansen, R.A. and Parker, M.G.** Binary Sequences with Merit Factor > 6.3. *Information Theory, IEEE Transactions on.* 2004, Vol. 50, 12.

23. **Nguyen, Lim.** *Self-Encoded Spread Spectrum Communications.* s.l. : Military Communications Conference Proceedings, MILCOM IEEE, 1999. Vol. 1.

24. **Nguyen, and W. M. Jang.** *Self-Encoded Spread Spectrum Modulation with Differential Encoding.* **Y. Kong, L.** Prague, Czech Republic : IEEE 7th Int. Symp. on Spread-Spectrum Techniques, Sept. 2002.

25. **Youn Seok Kim, Won Mee Jang, Lim Nguyen.** *Chip-interleaved Self-encoded Multiple Access with Iterative Detection in Fading Channels.* s.l. : Journal of Communication and Networks,, Mar. 2007. vol. 9, no. 1, pp. 50-55.

26. **Roger L. Peterson, Rodger E. Ziemer, David E. Borth.** *Introduction to Spread Spectrum Communications.* Upper Saddle River, NJ : Prentice Hall, 1995. 0-02-431623-7.

27. **Frost, O. L.** *An Algorithm for Linearly Constrained Adaptive Array Processing.* s.l. : Proceedings of the IEEE, 1972.

28. **K. Kowalske, R.C. Robertson.** *Performance of a Noncoherent RAKE Receiver and Convolutional Coding with Ricean Fading and Pulse-Noise Jamming.* Monterey, CA : Military Communications Conference, 2003. MILCOM 2003. IEEE, 2003. 10.1109/MILCOM.2003.1290354.

Appendix A – Iterative Detector Run Through

Overview of the detector looks like this:

$$Bit_1 = \left[e_1 e_0\,,\;\; e_1 e_{-1}\,, \dots,\, e_1 e_{-N}, e_1 e_{-(N+1)}\right]$$

$$Bit_2 = \left[e_2 e_1\,,\;\; e_2 e_0\,, \dots,\, e_2 e_{-(N+1)}\;\; e_2 e_{-(N+2)}\right]$$

$$Bit_3 = \left[e_3 e_2\,,\;\; e_3 e_1\,, \dots,\, e_3 e_{-(N+2)}, e_3 e_{-(N+3)}\right]$$

$$\vdots$$

$$Bit_N = [e_N e_{N-1}\,,\;\; e_N e_{N-2}\,, \dots,\, e_N e_1, e_N e_0]$$

$$Bit_{N+1} = \left[e_{N+1} e_N\,,\;\; e_{N+1} e_{N-1}\,, \dots,\, e_{N+1} e_2, e_{N+1} e_{1)}\right]$$

Step 1:

Decode Bit 1 using the despreading code, this step is the same as you do in regular SESS or any other spreading system. Take the dot product of the spreading code with the spreading sequence. Then as in DSSS system, the sum of the received data times the spreading code. (Notice the soft decision made on e_1, indicated by the $\dot{e}_1$, a final decision isn't made till the last step)

$$Bit_1 = \left[e_1 e_0\,,\;\; e_1 e_{-1}\,, \dots,\, e_1 e_{-N},\;\; e_1 e_{-(N+1)}\right] \xrightarrow{\text{Decode}} \;\; \dot{e}_1\text{, this is a +1 or -1}$$

Step 2:

Using the soft decision made in Step1, update the spread sequence used in the 2nd sent Bit, in this case Bit 2. Then, Decode Bit 2 using the updated sequence:

$Bit_2 = \left[e_2\dot{e}_1 ,\ e_2e_0 ,\dots,\ e_2e_{-(N+1)},\ e_2e_{-(N+2)}\right] \xrightarrow{Decode} \dot{e}_2$, (this is a +1 or -1)

Then update the Bit 2 ‘s sequence replacing e_2 with the

Step 3:

Take the soft decision made in step 2 and multiply it with the first chip. By multiplying the first chip by the bit that was sent ($\dot{e}_2$), this gives the original value of e_1. (See below). This is needed to determine if the value e_1 is positive or negative.

$$Bit_2's\ e_1\ value\ =$$

$$\dot{e}_2 * \left[e_2e_1 ,\ e_2e_0 ,\dots,\ e_2e_{-(N+1)},\ e_2e_{-(N+2)}\right] = \left[\boldsymbol{e_1} , e_0 ,\ \dots , e_{-(N+1)}e_{-(N+2)}\right]$$

It should be noted that in Bit 2, e_1 is the first chip. In proceeding Bits it will be different.

Step 4:

Repeat steps 2 and 3, but use the proceeding Bits (3,4,5…) till Bit N+1.

$Bit_3 = \left[e_3\dot{e}_2 ,\ e_3\dot{e}_1 ,\dots,\ e_3e_{-(N+2)}, e_3e_{-(N+3)}\right] \xrightarrow{Decode} \dot{e}_3$ (this is +1 or -1)

$$Bit_3's\ e_1\ value\ =$$

$$\dot{e}_3 * \ \left[e_3e_2 ,\ e_3e_1 ,\dots,\ e_3e_{-(N+2)},\ e_3e_{-(N+3)}\right]\ =\ \left[e_2 , \boldsymbol{e_1}\ , \dots\ ,\ e_{-(N+2)},\ e_{-(N+3)}\right]$$

…..

Through Bit N+1

Step 5:

Take the sum of all the e_1 values from Bits 2 through N+1 and add it to the dot product of Bit 1 with Bit 1’s despreading code. This gives you the final decision for Bit 1, $\widehat{e_1}$.

$$\widehat{e_1} = \left(\sum_{x=2}^{N+1} Bit_x's\, e_1\ \ value\right) + dot(Bit_1, Bit_1's\ despreading\ code)$$

Step 6:

Update the spread sequence using the final decision for Bit 1.

$$Bit_2 = \left[e_2\widehat{e_1}\ ,\ \ e_2e_0\ , \dots ,\ e_2e_{-(N+1)}, e_2e_{-(N+2)}\right]$$

$$Bit_3 = \left[e_3e_2\ ,\ \ e_3\widehat{e_1}\ , \dots ,\ e_3e_{-(N+2)}, e_3e_{-(N+3)}\right]$$

$$\vdots$$

$$Bit_N = \ [e_Ne_{N-1}\ ,\ \ e_Ne_{N-2}\ , \dots ,\ e_N\widehat{e_1}, e_Ne_0]$$

$$Bit_{N+1} = \ [e_{N+1}e_N\ ,\ \ e_{N+1}e_{N-1}\ , \dots ,\ e_{N+1}e_2, e_{N+1}\widehat{e_1}]$$

Step 7:

Receive bit N+2 and repeat by shifting all bits up a spot. The new received bit N+2 gets moved to the Bit N+1 spot (bit 2 becomes bit 1, bit 3 become bit2, …, bit N+1 becomes bit N). And then repeat steps 1-6.

Appendix B – Matlab Code

```
%Casey Deyle
%Version 1 - proving underlying Direct Sequence Spread Spectrum works
%by comparing to BPSK system
%Version 2 - Implementing selfencoded spread spectrum on top of Spread
%Spectrum
%Version 3 - Adding the Jamming signal
%Version 4 - Adding selfencoded spread spectrum to jamming and jamming
%            each chip instead of each bit
%Version 5 - Adding irative detector to the self-encoded spread spectrum
%Version 8 - Final
clear;
%------------------- Declaring Varibles -----------------------------------
figure;
ch_bt=32; % Set the chips/bit
increment = 1; %How much you increment the SNR in the loop
Max_Eb_No_db=20; %Max SNR, changes graph window as well
Bits2Tx = 100000; %How many Bits to transmit
BER = zeros(1,floor(Max_Eb_No_db/increment)); %empty BER
Eb_No = zeros(1,floor(Max_Eb_No_db/increment)); %empty Eb/No
temp = zeros(1,ch_bt);
b = zeros(1,ch_bt+1);

jdcnt = [1] ;
%jdc=.02; % Set the jamming duty cycle (percent) 20% = .2, 100%=1, etc.

for z=1:length(jdcnt)
    jdc = jdcnt(z);

    %------------------- Setup ------------------------------------------------
    %figure; %crates new figure

    arraycounter = 0; %initialize to 0

    %for display in Matlab window
    fprintf('Chips per bit = %g\tBits Transmitted = %g\n',ch_bt,Bits2Tx);
    fprintf('Max Eb_No = %gdb\tIncrements = %gdb\n',Max_Eb_No_db,increment);
    fprintf('Jamming Duty Cycle = %g percent\n',jdc*100);

    for SNR1=0:increment:Max_Eb_No_db
        arraycounter = arraycounter+1;
        Eb_No(arraycounter) = SNR1;
        fprintf('SNR: %g (',SNR1); %for display in Matlab window
        SNR = 10*log10(ch_bt)-SNR1; %Adjusts for Processing Gain
        data_reg_ct = ch_bt+1;
        data_reg = zeros(ch_bt+1, ch_bt);
        spread_reg = zeros(ch_bt+1, ch_bt);
        %--------- Generate the initial bit pattern used for spreading ----
         s_sig=2.*round(rand(1,ch_bt))-1; %random # 1 or -1

        %------------------  Set the initial despreading signal -----------
        ds_sig=s_sig;
        Errors = 0;
        BitsTx = 0;
        hardbit = 0;
        hardbit1 = 0;
        %------------------------- Start Transmitting Bits ----------------
        while (BitsTx<Bits2Tx)
```

```
            BitsTx=BitsTx+1; %increment bits transmitted

            b(data_reg_ct)=(2*round(rand(1))-1);% Generate the next bit to
                                                %be transmitted
            t_sig=[b(data_reg_ct)*s_sig];% Multiply the pattern with the
            %g_sig = awgn(t_sig,-(SNR-3),0); %with spreading signal
            %fade=sqrt(randn^2+randn^2)/sqrt(2);
            %rchan_sig=fade*t_sig+(1/sqrt(2*10^(SNR1/10))*randn(1,ch_bt)*sqrt(ch_bt));

            j_sig = g_sig; %jamming signal = transmitted signal
                           %when no jamming present, or chip not jammed

            %------ Adding the jamming signal, jamming done on chip level
            for i=1:length(j_sig)
                if (floor(rand(1)+jdc)==1) %jdc percent chance of jamming
                    j_sig(i) = j_sig(i) + wgn(1,1,(SNR-3))*(1/sqrt(jdc));
                end
            end

            data_reg(data_reg_ct,:) = j_sig;
            spread_reg(data_reg_ct,:) = ds_sig;

            if data_reg_ct > 1
                %---------------- Update Spread Sequence ------------------
                s_sig=circshift(s_sig,[1,1]); %rotates in a 0
                s_sig(1)=b(data_reg_ct);     %and sets a new 1st bit
                ds_sig=circshift(ds_sig,[1,1]); %rotates in a 0
                ds_sig(1)=b(data_reg_ct);   %and sets a new 1st bit
                data_reg_ct = data_reg_ct - 1;

            else
                n= ch_bt+1;
                %-----------Bit Decision
                for n=1:ch_bt
                    if dot(spread_reg((ch_bt+2)-n,:),data_reg((ch_bt+2)-n,:)) < 0
                        bit = -1;
                    else
                        bit = 1;
                    end %n=1:(4,1),(3,2),(2,3),(1,4)
                    for p=1:ch_bt+1-n % n=2:(3,1),(2,2),(1,3)
                        spread_reg((ch_bt+2)-(p+n),p) = bit;
                    end
                    if(dot(spread_reg((ch_bt+1)-n,:),data_reg((ch_bt+1-n),:))<0)
                        bit2 = -1;
                    else
                        bit2 = 1;
                    end
                    temp(n) = data_reg((ch_bt+1)-n,n) * bit2;
                end

                d_sig = dot(data_reg(ch_bt+1,:),spread_reg(ch_bt+1,:)) + sum(temp);

%--------------------------------Chip Decision ----------------------------
%                  for n=1:ch_bt
%
%                     if dot2(spread_reg((ch_bt+2)-n,:),data_reg((ch_bt+2)-n,:)) < 0
%                         bit = -1;
%                     else
%                         bit = 1;
%                     end
%                     for p=1:ch_bt+1-n %n=1:(4,1),(3,2),(2,3),(1,4)
%                            %n=2:(3,1),(2,2),(1,3)
%                         spread_reg((ch_bt+2)-(p+n),p) = bit;
%                     end
%                     if(dot2(spread_reg((ch_bt+1)-n,:),data_reg((ch_bt+1-n),:))<0)
%                         bit2 = -1;
%                     else
%                         bit2 = 1;
%                     end
%
%                     temp(n) = dot2(bit2,data_reg((ch_bt+1)-n,n));
%                 end
%
%                 d_sig = dot2(spread_reg(ch_bt+1,:),data_reg(ch_bt+1,:)) + sum(temp);

                if d_sig>0 %gets value for decoded bit
                    cd_sig=1;
                else
                    cd_sig=-1;
                end

                if(b(ch_bt+1) ~= cd_sig) %bit transmitted is != decoded bit
                   Errors=Errors+1;
                end

                %-------------- Progress Bar for Matlab Window ------------

                if(mod(BitsTx,Bits2Tx/10) == 0)
                    fprintf('.');
                end

                %---------------- Update Spread Sequence ------------------

                for j=1:ch_bt
                    spread_reg(j,(ch_bt+1)-j) = cd_sig;
                end

                s_sig=circshift(s_sig,[1,1]); %rotates in a 0
                s_sig(1)=b(1);               %bit, and sets a new 1st bit
                 %change the self encoded part of the spreading codes

                ds_sig=circshift(ds_sig,[1,1]);
                ds_sig(1)=cd_sig;

                for m=0:ch_bt-1
                    data_reg(ch_bt-(m-1),:) = data_reg(ch_bt-(m),:);
                    spread_reg(ch_bt-(m-1),:) = spread_reg(ch_bt-(m),:);
                    b(ch_bt-(m-1)) = b(ch_bt-m);
                end
            end
        end %----------------- Bits transmitted loop ---------------------

        BER(arraycounter)=(Errors/Bits2Tx); %updates BER for that SNR
        fprintf(')\n');%prints new line in Matlab window

    end %--------------------- SNR increment loop--------------------------

    if (z == 1)
        semilogy(Eb_No,BER,'y'); %sets legend for simulation BER
    end
    if (z == 2)
```

```
        semilogy(Eb_No,BER,'r'); %sets legend for simulation BER
    end
    if (z == 3)
        semilogy(Eb_No,BER,'b'); %sets legend for simulation BER
    end
    if (z == 4)
        semilogy(Eb_No,BER,'g'); %sets legend for simulation BER
    end
    if (z == 5)
        semilogy(Eb_No,BER,'k'); %sets legend for simulation BER
    end
    if (z == 6)
        semilogy(Eb_No,BER,'m'); %sets legend for simulation BER
    end
    hold on;
end

%-------------------- Graph and Graph Titles -----------------------------
for i=1:length(Eb_No)
    EBnotdb = 10^(Eb_No(i)/10);
    theoryBer(i) = .5*(1-sqrt(EBnotdb/(EBnotdb+1)));%Rayleigh fading
    theoryBer2(i) = jdc * Q( sqrt(2*jdc*(10^(Eb_No(i)/10))));%AWGN
end
semilogy(Eb_No,theoryBer2,'m');
hold on;
semilogy(Eb_No,theoryBer,'mx-');
hold on;
axis([0 Max_Eb_No_db 10^-6 0.5]) %sets vaule of axis
grid on %sets dB grid
%these are pretty self explanitory
xlabel('Eb/No (dB)');
ylabel('BER');
title('\bf\it BER vs. Eb/No N=32 Tx=100000');
%legend('theory', 'simulation');
legend('jdc = 1');

function Qfunc = Q(x)
 Qfunc = erfc(x/sqrt(2))*.5;
End

function dotfunc = dot2(y,x)
tempsum = 0;
    for i=1:length(x)
        if x(i) < 0
            bit = -1;
        else
            bit = 1;
        end
        tempsum = tempsum + bit*y(i);

    end

 dotfunc = tempsum;
end
```