University of Alberta

JOINT PACKET DETECTION AND FRAME SYNCHRONIZATION FOR ASYNCHRONOUS WIRELESS NETWORKS

by

Sheehan Khan 

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Electrical and Computer Engineering

Edmonton, Alberta
Spring 2007

Your file Votre référence
ISBN: 978-0-494-29978-4
Our file Notre référence
ISBN: 978-0-494-29978-4

And in my darkest moment, fetal and weeping.
The moon tells me a secret.
"My confidant.
As full and bright as I am, this light is not my own
A million light reflections pass over me
It's source is bright and endless."

~

Reflection
Maynard James Keenan

To the loving folks out on Neptune Orbit

Abstract

When random access techniques are adopted in wireless networks it is necessary to insert a preamble sequence at the start of all packet transmissions to help facilitate detection and synchronization at the receiver. Previously the detection and synchronization tasks were treated as two separate problems. This thesis proposes a new preamble format and detector which can perform both tasks jointly. The detector is derived based on the principals of hypothesis testing and is found to have two mutually exclusive error events; miss and false detection. Expressions for the probabilities of the error events are derived as well as easy to compute approximations which fit closely with simulated data. It is shown that the system performance is fixed by the structure of the preamble sequence and not by the individual bits within it. Furthermore it is shown that with as little as two samples per chip, that the detector performance is close that that of a chip synchronous detector.

Acknowledgements

I would like to thank my family for their endless support. My father Shams, my mother Eila and my sister Tanya. Over the past 25 years, it has been their love that has carried me through life and brought me to the success herein the completion of my Masters degree. To them, not enough thanks can be said.

I would also like to thank my supervisor Dr. Schlegel for his guidance and mentorship over the past two years. Thanks are also due to my friend and colleague Sumeeth Nagaraj who worked with me on this project.

At this point I would like to acknowledge that I received financial support from the following agencies to conduct this research: NSERC, iCore, The University of Alberta, The Alberta Ministry of Advanced Education.

Furthermore on a social level, thanks to all the friends that I have made here in Edmonton. Without them I would have had no one to hang out, watch movies, and drink beer with. The lack of which would have surely driven me insane.

Table of Contents

List of Figures

List of Symbols

Symbol	Definition
H_0	Null hypothesis
H_1	Desired hypothesis
η	Decision Metric for hypothesis testing
G	Threshold for hypothesis testing
$\vec{c}$	Preamble/synch sequence
$\vec{a}$	Differentially encoded preamble sequence
W	Length of preamble sequence
$\mathbf{C}$	Correlation matrix
$U_n(x)$	Chebyshev polynomial of the second kind with order n
Λ	Eigenvalue Matrix
λ	Eigenvalue
$\mathbf{P}$	Matrix of Eigenvectors
$\vec{P}_i$	Eigenvector
P_{ij}	Eigenvector elements
E_p	Total preamble energy transmitted
E_b	Transmitted energy per symbol
E_c	Transmitted energy per chip
L_b	Length of the spreading sequence (chips/symbol)
L_s	Upsampling factor (samples/chip)
T_c	Chip Duration
i_0	Starting moment of the preamble sequence
$s(t)$	Transmitted signal
$p(t)$	Transmit pulse
$n(t)$	Gaussian noise process
Δf	Frequency offset
ϕ	Phase offset
τ	Time offset
$r(t)$	Received signal
$\vec{r}$	Received sequence

List of Abbreviations

Abbreviation	Definition
AWGN	Additive White Gaussian Noise
BPSK	Binary Phase Shift Keying
CDMA	Code Division Multiple Access
DSSS	Direct Sequence Spread Spectrum
FHSS	Frequency Hopping Spread Spectrum
I.I.D.	Independent and Identically Distributed
ISI	Intersymbol Interference
MAC	Medium Access Control
MAI	Multiple Access Interference
MFSK	M-array Frequency Shift Keying
ML	Maximum Likelihood
MPSK	M-array Phase Shift Keying
PDF	Probability Density Function
SNR	Signal to Noise Ratio
Synch	Synchronization
TDMA	Time Division Multiple Access

Chapter 1

Introduction

With recent advances in wireless technologies it has become feasible to place transceiver circuits on almost any device. These devices now have the freedom to communicate with each other. In order for the devices to communicate with each other without a centralized base station to coordinate the network, random access communication techniques must be used.

Additionally, even in many basestation centric networks such as mobile cellular and wireless internet, the basestation is only able to synchronize communication on the down link (basestation to node) communication. It is complicated to have all nodes synchronize their transmissions on the uplink back to the basestation. Thus, the uplink traffic as viewed from the base station may also be a form of random access communication.

In random access systems it is very easy for nodes to transmit data at a random time. The data is encapsulated into a packet and broadcast out into the network. Before any medium access control (MAC) and routing protocols can be run at the receiver to properly relay the packet off to its destination, there are fundamental problems to be solved on the physical level. Primarily, the receiver must first detect the presence of the packet in the received signal. Furthermore, each packet consists

of a series of data frames containing various network parameters as well as the actual data message. Thus, it is not only necessary for the packet to be detected, the receiver must also be synchronized to the frame timing. Meaning that the start and end points of the frame must be located so that the data within the frame can be properly processed. Frame synchronization can only be achieved if the receiver is first synchronized to the lower levels of timing such as symbol, and chip.

The problem of detecting random data at a random time without synchronization is extremely difficult. To facilitate the process of packet detection a preamble is inserted at the start of every packet. Thus, the receiver no longer needs to detect random data at a random time, but needs only to detect the known preamble at a random time. Additionally, the preamble can also used to obtain synchronization at the various levels.

1.1 Thesis Overview

This thesis deals with the problem of preamble detection and frame synchronization for asynchronous wireless networks. Here asynchronous means that nodes in the network are unable to coordinate the timing of their transmissions, as they are all operating on their own clocking circuits which are independent and thus unsynchronized.

We begin in Chapter 2 which provides an over view of classic detection and synchronization techniques.

Chapter 3 presents the channel model under consideration for this thesis and proposes a new packet and preamble structure. A new detector which performs joint packet detection and frame synchronization is derived. The performance of the detector is evaluated based on the error events of missing the packet and falsely

detecting a packet prior to its arrival.

In Chapter 4 the detector is simulated under various design constraints to verify the expressions derived for the system performance in Chapter 3. Additionally, the effect of each of the system design parameters on the performance of the system is discussed.

In Chapter 5 system implementation issues are discussed. The effect of pre-detection and post-detection timing offsets are discussed. Also, sequence selection for implementation is discussed.

Chapter 6 concludes the thesis and provides closing remarks.

Chapter 2

Random Access Communications

In many communication networks it is often infeasible or impractical for the nodes to share timing information. For example, ad-hoc network have no centralized base station to coordinate the communication among the nodes. Thus, there is no sense of a global clock for nodes to synchronize to, rendering it impossible to schedule packet transmissions. To overcome this obstacle medium access control (MAC) techniques have been developed to allow for packet transmissions to occur at random intervals [14].

Even in basestation centric networks random access techniques may be used. This is largely due to fact that in many real life applications the data tends to be "bursty", which means that there are long idle times between packet transmissions. Due to the long idle times between transmissions scheduled communication schemes such as time division multiplexing (TDMA) can become inefficient as there will be many wasted time slots.

Furthermore, in basestation centric communications where all nodes are able to synchronize to the clock at basestation, it is still difficult for nodes to coordinate their transmissions such that they arrive synchronously at the basestation. In order for the transmissions to arrive synchronously each node would then need to advance

or delay their transmissions by an amount of time prescribed by the basestation.

When multiple packets are arriving at the basestation or at a receiving node, the packets interfere with each other creating a superposition of packets and noise. When receiving one of the packets the contributions from all the other packets is then considered to be multiple access interference (MAI) and acts as an additional noise source to the receiver.

In code division multiple access (CDMA) systems the MAI is reduced by assigning different spreading codes to each user. The receiver can then select a specific user by despreading the recieved signal with that users sequence. After despreading the MAI will be suppressed by the processing gain of the system. Thus, it is likely that future ad hoc and random access networks will adopt similar direct sequence spread spectrum (DSSS) techniques. See Appendix A for more details on spread spectrum communications.

This chapter will present classical approaches to packet detection which operate based on estimates of the received signal energy. Along with their extension for the case of DSSS systems. Generic synchronization techniques are also discussed along with their extensions for synchronization to data frames within a packet.

2.1 Packet Detection

To detect known signals in white, spectrally flat, noise it has been shown that the optimal pre-detection filter is the matched filter [10]. The problem of packet detection can then be formulated as a hypothesis test on the filter output, with two possible hypotheses. Hypothesis H_0 is that there is no packet present in the received signal and hypothesis H_1 is that the received signal contains the packet plus noise.

2.1.1 Energy Detection

Early detectors operated on the principal of measuring the received signal energy [10, 11]. A decision metric η is formed to estimate the received signal energy,

$$\eta = \sum_{n=1}^{N} r_n^2. \tag{2.1}$$

Where r_n denotes the n'th filter sample. When the noise is Gaussian distributed η will be chi-square distributed with N degrees of freedom; for complex Gaussian noise there are $2N$ degrees of freedom.

$$f_{\eta|H_0}(x) = \frac{1}{2^{N/2}\sigma^N \Gamma(\frac{1}{2}N)} x^{N/2-1} e^{-x/2\sigma^2} \tag{2.2}$$

$$f_{\eta|H_1}(x) = \frac{1}{2\sigma^2}\left(\frac{x}{s^2}\right)^{(N-2)/4} e^{-(s^2+x)/2\sigma^2} I_{N/2-1}\left(\sqrt{x}\frac{s}{\sigma^2}\right) \tag{2.3}$$

$$s^2 = \sum_{n=1}^{N} m_n^2$$

The distributions of η conditioned on H_0, and H_1 are given in (2.2) and (2.3). Where s^2 is the non-centrality parameter for the H_1 hypothesis, which consist of the sum of the mean sample values. $\Gamma(x)$ is the gamma function, and $I_\alpha(x)$ is the α'th order modified Bessel function of the second kind.

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$$

$$I_\alpha(x) = \sum_{k=0}^{\infty} \frac{(x/2)^{\alpha+2k}}{k!\Gamma(\alpha+k+1)}$$

Setting a threshold G, the detector makes the decision H_0 if $\eta < G$ and H_1 if $\eta > G$. Defining a correct detection to be the event that there is a packet and the detector properly detects it can be denoted as,

$$\begin{aligned} P(correct) &= P(H_1)P(\eta > G|H_1) \\ &= P(H_1)\int_G^\infty \frac{1}{2\sigma^2}\left(\frac{x}{s^2}\right)^{(N-2)/4} e^{-(s^2+x)/2\sigma^2} I_{N/2-1}\left(\sqrt{x}\frac{s}{\sigma^2}\right) dx. \end{aligned} \tag{2.4}$$

Correspondingly a false detection is when there is no signal present and the noise drives the decision metric past the threshold. The probability of false detection is thus given by,

$$\begin{aligned} P(false) &= P(H_0)P(\eta > G|H_0) \\ &= P(H_0)\int_G^{\infty} \frac{1}{2^{N/2}\sigma^N\Gamma(\frac{1}{2}N)} x^{N/2-1}e^{-x/2\sigma^2}dx. \end{aligned} \tag{2.5}$$

2.1.2 Ratio Testing

A drawback to energy-based detection is that it is highly dependent on the received energy and not on the received SNR. For a constant SNR the detection probabilities from (2.4) and (2.5) will be distorted for different values of signal energy.

As an alternative to estimating the signal energy, it is possible to estimate the SNR by increasing the observation interval such that the energy can be estimated twice. The new decision metric is designed by taking the ratio of these powers.

$$\eta = \frac{\sum_{n=1}^{N} r_n^2}{\sum_{n=N+1}^{2N} r_n^2} \tag{2.6}$$

Prior to the arrival of the packet, when the received signal contains only noise it is clear that $\eta = \frac{\sigma^2}{\sigma^2} = 1$. Similarly if all samples contain the packet plus noise the ratio of the powers will still be constant. However, as the packet starts to arrive the estimate in the numerator will start to rise due to the presence of the packet energy E. Thus, η will reach the limit $\frac{E+\sigma^2}{\sigma^2} = SNR + \frac{1}{\sigma^2}$ when the numerator samples all contain the packet, and the denominator remains noise driven. The threshold level G can now be set as an SNR value instead of an absolute value, at the cost of the added samples.

The analysis of the detection events using (2.6) can be done in exactly the same way as before. In order to do so the distribution of η is required. The distribution

of the ratio of two non-central chi-square random variables is given by [20] as the doubly non-central F distribution. The expression for this distribution is very complex and is hence omitted here, as it yields no insight into the problem.

Although, this approach provides protection, against setting the threshold as a specific amount of received energy, it only works well at the moment the signal begins to arrive and requires twice the amount of samples to be processed. Thus, energy-based techniques are typically used because they are easy, and can always detect the presence of a signal.

2.1.3 Spread Spectrum Detection

The detection techniques based on estimating the received signal energy or SNR can be made to work for most communication systems but they break down when used with DSSS. The problem arises because DSSS systems use spreading to operate on low transmission SNR. Often the signal will be comparable to or buried within the noise and thus any energy estimates as in Sections 2.1.1 and 2.1.2 will only produce estimates of the noise energy.

In DSSS systems it is possible to despread the matched filter samples and then run the detector in the high SNR regime. However, as the packet has not yet been detected, the receiver does not have the chip timing and thus there will be a degradation to the despreading as seen in (2.7). The timing error will be some value τ uniformly distributed on the interval $(-\frac{T_c}{2}, \frac{T_c}{2})$. Assuming that the packet has arrived and is in the detector, the n'th symbol after despreading can be expressed as,

$$\hat{b}_n = \sum_{l=1}^{L} c_l R(\tau - lT_c) + \sum_{l=1}^{L} c_l n_l. \tag{2.7}$$

Where $R(\tau)$ is the auto-correlation function of the transmission pulse used.

Clearly the offset τ has a scaling effect on the received energy and also introduces intersymbol interference (ISI) between $\hat{b}_n$ and $\hat{b}_{n+1}$. With typical transmission pulses the ISI at the chip level can be neglected when compared to the scaling effect [26].

In his book Viterbi [26] proposes the following energy-based detector which evaluates the energy of (2.7) for all sample points k within the span of the spreading sequence,

$$\eta_k = |\hat{b}_n|^2 = \left| \sum_{l=1}^{L} c_l r(kT_c) \right|^2 . \tag{2.8}$$

Since η_k is evaluated at all sample points within the span of the spreading sequence, it is guaranteed that at one of the sample points the local spreading sequence will be aligned with the sequence in the received signal.

Operating with the same hypothesis testing principals as before, the H_1 hypothesis is defined to be that the k'th received sample is the starting point of the spreading sequence in the received signal. The H_0 hypothesis is then defined to be that the k'th sample is not the start of the spreading sequence. The detection is performed by selecting k such that η_k is the maximum from all possible hypothesis tests. This result can be refined by performing "post detection integration" which sums the result of (2.8) over successive symbols [26].

Clearly this detection method works well in CDMA systems when there is a carrier present. However, in the random access systems under consideration it is unrealistic to assume the detector can evaluate the decision metric η_k over a large enough region to guarantee that at least one sample corresponds to the start of a spreading sequence. Thus, instead of selecting H_1 as the maximum η_k within a window, the first η_k greater than a preset threshold G can be selected to be H_1.

Aside from allowing the system to operate in lower SNR regions than the previ-

ous energy based techniques allowed, this technique has the advantage of simultaneously synchronizing to the spreading code used whereas the previous techniques would only be able to provide synchronization to chip level if they were employed.

2.2 Synchronization

It is not sufficient to just detect incoming packets. In order to properly demodulate, decode and process the packet, the receiver must also locate the boundaries of the data frames within the packet. The frames can be viewed as either blocks containing useful network parameters or block codewords to decode. For the purpose of synchronization there is no distinction between the two.

In [2] Barker addressed the problem of frame synchronization as that of pattern recognition. If in every N bit data frame the first W bits are set to be a known synchronization (synch) pattern, synchronization can then be achieved if the pattern can be found because the $N-W$ data bits must start thereafter.

To perform the pattern recognition Barker proposed that the data is correlated with the known synch pattern. The pattern is then located by identifying the peak of the correlator output. To select good synch patterns Barker used the combination of the patterns auto-correlation properties along with the probability of the data in the frame to form the same pattern elsewhere in the frame. Based on this criteria a series of patterns were given, now known as the Barker codes.

Later, in [16] Massey derived an optimal ML frame synchronizer for the same W bit synch word in a N bit data frame. It is shown that the optimal rule is to add a correction term to the correlator output to account for the random data within the frame. Additionally, he showed that the problem of locating a multiple of n synch words across n consecutive data frames is equivalent to locating a nW bit synch in

a nN bit frame.

The ML formulation proceeds as follows. Consider a received vector $\vec{r} = (r_1, r_2, \ldots, r_N)$. Since, the vector length is equal to the frame length, the W bit synch word must lie within $\vec{r}$ and may start at any position r_k with equal probability. In the event that the synch word falls in the last $W-1$ positions, then r_1 can then be considered to follow r_N as it would be the appropriate part of the synch word from the previous frame. A second vector $\vec{t}$ can be defined which contains the synch word and $N-L$ zeros, $\vec{t} = (0, \ldots, c_1, c2, \ldots, c_W, \ldots, 0)$. Where $\vec{c} = (c_1, \ldots, c_W)$ is the synch word. The ML synchronizer will then evaluate the distance between both vectors for all possible locations of the synch word and selects the position $\hat{i}_0$ that minimizes it. After averaging out all possible data in $\vec{r}$ the ML decision then becomes the estimate $\hat{i}_0$ which maximizes the statistic,

$$\eta = \sum_{m=1}^{W} c_m r_{m+\hat{i}_0-1} - \sum_{m=1}^{W} |r_{m+\hat{i}_0-1}|. \tag{2.9}$$

A drawback of ML and correlation based synchronization algorithms is that they operate on a received window which is the same length as the data frame. Thus, unless the synchword is at the exact start of the window, the window must span two data frames. The problem is that, this means that a potentially large part of the first data frame will be lost during the synchronization stage. In systems where there is continuous communication this is not a large problem because after the initial synchronization is performed many frames of data will be correctly received. However, in the case of packetized data for the asynchronous networks considered in this thesis, there may only be one frame of data. Synchronization under these conditions can be obtained by modifying the packet structure.

For the case of asynchronous packet transmissions Mehlan and Meyr [17] present

a modified packet structure which works well to facilitate both detection and synchronization. The packet begins with a bit synch. The bit synch is designed to be long enough to allow energy-based detection of the packet. With the detection it is also assumed that the receiver is able to synchronize to the bit timing of the packet and undo any rotation incurred in the channel. Thus, the rest of the receiver can operate coherently on the sampled bits. A second synch word, known as the frame synch, is then inserted into the packet which is then followed by the data frame. In this manner it is now possible to detect the packet and then run the frame synch algorithm to find the start of the data frame.

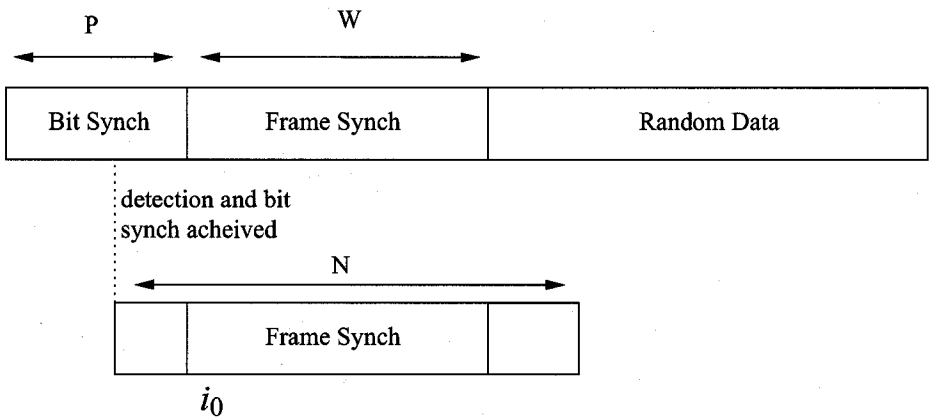

Figure 2.1: Modified packet format used by Mehlan and Meyr

As shown in Figure 2.1 the length of the bit synch is P and the length of the frame synch is still given as W. Given that the packet detection must occur at some time in the first P bits, it is then sufficient to run the synchronization algorithm on a window of length $N = P + W$. This window is guaranteed to capture the entire frame synch along with portions of the bit synch and the random data. It is then possible to follow the same approach as [16] and compute the distance between the received window, and a window containing only the bit and frame synchs. The new

decision metric is given as,

$$\eta = \sum_{m=2-\hat{i}_0}^{0} x_{p+m} r_{m+\hat{i}_0-1} \sum_{m=1}^{W} c_m r_{m+\hat{i}_0-1} - \sum_{m=1}^{W} |r_{m+\hat{i}_0-1}|. \qquad (2.10)$$

Where x_m are the bits of the known bit synch. Note that if the the bit synch is unknown or neglected, (2.10) reverts back to the previous expression derived by Massey. However, this method is much better because the window length used is considerably smaller as it is approximately the length of the frame synch and not the data frame. Typically the data frame must be much larger than the frame synch, otherwise the overhead of the frame synch will become too costly. The main problem with this technique is that nothing is said about how the bit synch should be detected, or how it will be used to remove the incurred rotation.

Chapter 3

Models And Analysis

The problems of packet detection and frame synchronization as presented in Sections 2.1 and 2.2 are essentially identical. Both techniques search the received signal for key sequences which mark the boundaries they are looking for. This raises the question, must these problems be treated separately? The fundamental problem that needs to be solved is locating the start position of the data frame. For this reason the modified packet format shown in Figure 3.1 is adopted for this thesis. It is similar to that of Mehlan and Meyr but instead of having seperate bit and frame synchs a single preamble sequence is used. The length of the preamble is denoted as W. In this chapter a preamble detector will be derived which performs packet detection and frame synchronization operations jointly on the modified packet format. The preamble structure will given in section 3.1.2. Derivation and motivation for a correlation based preamble detector will be presented in Section 3.2, with error analysis in Section 3.3.

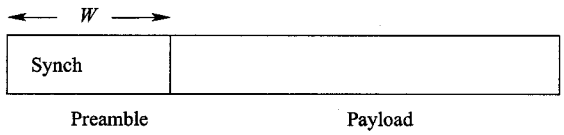

Figure 3.1: Modified packet format for this thesis

3.1 System Model

3.1.1 Channel Conditions

As in Chapter 2 this thesis will consider communication over the additive white Gaussian noise (AWGN) channel. Additionally the channel is considered to add random frequency and phase offsets to the received signal. The phase offset arises from the lack of synchronization between the oscillators at the transmitter and receiver. The frequency offset is introduced by oscillator drift which causes the operating frequency of the oscillator to deviate over time [21]. In mobile systems frequency offset can also be created by Doppler shifts [25]. Thus, denoting the transmitted signal as $s(t)$ the received signal through such a channel is given by,

$$r(t) = e^{j(2\pi\Delta f t+\phi)}s(t) + n(t), \tag{3.1}$$

where $n(t)$ is a complex Gaussian noise process with noise variance $\sigma^2 = N_0/2$ per dimension.

3.1.2 Preamble Structure

Due to the frequency and phase offsets introduced by the channel, a coherent detection scheme cannot be employed for the preamble detector. It is possible to use a pilot sequence which will allow the receiver to estimate the channel and thus compensate for the rotation incurred. However, the use of a plot signal will introduce an additional rate loss due to the transmission time of the pilot [21]. Even if the rate loss is acceptable it introduces a second problem as well. Since the channel estimation must take place prior to detection, the channel estimator will constantly be running. This is highly inefficient because for most of the time there is no signal to process so the estimator will waste energy to estimate an empty channel. This is

unacceptable in many ad hoc network scenarios where power conservation is a key network design parameter [7]. Alternatively, the pilot signal could be considered to be the bit synch from Mehlan and Meyr's packet format and thus the receiver would not be jointly detecting and synchronizing.

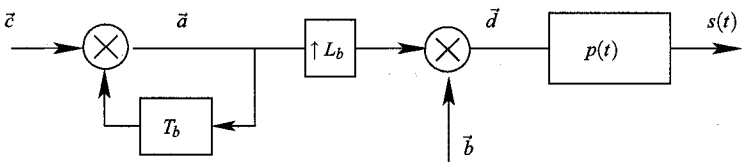

Figure 3.2: Block diagram of the preamble sequence generator

To avoid the use of pilot signals, and the non-coherent schemes of Chapter 2, a differential coherent scheme can be employed to encode the preamble. Figure 3.2 shows the block diagram for the preamble generator. The preamble sequence $\vec{c}$ is a sequence of W symbols. The symbols are i.i.d. random variables with $\mathrm{P}(c_m = 1) = p = 1 - \mathrm{P}(c_m = 0)$. Note that even though any random sequence may be used for the preamble sequence, this sequence must also be known at the receiver in order for the detector to work. In multiuser scenarios all users will use the same preamble sequence such that each node need only have one preamble detector. The differential scheme encodes the sequence $\vec{c}$ into the transmission sequence $\vec{a}$ according to the rule,

$$a_m = c_m a_{m-1}, \qquad (3.2)$$
$$a_1 = c_1.$$

With such a transmission scheme the receiver need not track the absolute phase of the received symbols but can instead track phase transitions of π radians. Which allows the phase rotation in the channel to be transparent to the receiver operation.

Based on the DSSS techniques presented in in Appendix A, the sequence $\vec{a}$ is then upsampled by a factor of L_b and multiplied by sequence $\vec{b}$ to create the modulation sequence $\vec{d}$. The sequence $\vec{b}$ is free to be any binary sequence of length WL_b. This provides a spreading gain of L_b. The main motivation for the use of spreading is to maximally benefit from coherent signal addition at the chip level as the channel allows before relying on the differential addition across the symbol level. The final expression of the baseband preamble is given as,

$$s(t) = \sqrt{E_c} \sum_{k=1}^{WL_b} d_k p(t - kT_c - i_0), \tag{3.3}$$

where E_c is the energy per chip, $p(t)$ is the pulse shaping function, T_c is the chip duration, and i_0 is the starting moment of the preamble. The pulse $p(t)$ is normalized to unit energy such that the total preamble energy is $E_p = WL_bE_c$. The pulse must also meet the Nyquist no ISI criterion,

$$p(kT_c) = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}. \tag{3.4}$$

In order for the receiver to be able to accurately despread the signal and benefit from the coherent signal addition, the rotation across each transmitted symbol a_m must be small. Given the notation above and the channel model (3.1), the rotation across each symbol at the receiver will be $\Delta f L_b T_c$. For the rotation to be negligible L_b should be chosen such that $\Delta f L_b T_c \ll 1$.

3.2 Deriving The Detector

The starting moment of the preamble is a random variable. In most communication networks packet arrival rates are modeled as a Poisson processes with an exponential distribution [14],

$$f_{i_0}(i_0) = \gamma e^{-\gamma i_0} \qquad 0 \leq \gamma, i_0. \tag{3.5}$$

The goal of the detector will be to successfully estimate i_0 from the received signal. By correctly estimating i_0 the detector inherently detects the packet and its starting position. Thus, it jointly solves the detection and frame synchronization problem. Section 3.2.1 derives an optimal ML detector based on the given preamble structure. However, the ML detector is computionally to complex for implementation, so a more practical detector is developed in Section 3.2.2.

3.2.1 The Optimal Detector

Based on the channel model given by (3.1) the received baseband preamble sequence can be denoted as,

$$r(t) = \sqrt{E_c} \sum_{k=1}^{WL_b} d_k p(t - kT_c - i_0) e^{j(2\pi\Delta f t + \phi)} + n(t). \tag{3.6}$$

For a given set of $i_0, \Delta f$, and ϕ it is possible to write an expression for the probability of correct detection as,

$$P(\hat{i}_0 = i_0 | i_0, \Delta f, \phi). \tag{3.7}$$

The optimal detector is the detector which will produce the estimate $\hat{i}_0$ such that (3.7) is maximized. The probability (3.7) is given in [4] for the special case where $i_0 = 0, \Delta f = 0, L_b = 1$. Even for this extremely simplified case the expression of (3.7) is too complex to be included here. Hence, it is clear to see that the generalized expression of (3.7) that would be obtained by integrating out Δf, and ϕ will be far too complicated to express, and computationally too complex for implementation in a practical receiver.

Due to the infeasibility of an optimal detector it is possible to explore the idea of developing a sub-optimal detector. A natural alternative would be to derive an ML detector. Using an ML approach it would be necessary to evaluate $\hat{i}_0$ at several

points in a given time window and select the one that is most likely the start of a preamble. This can be viewed as an extension of Massey's synchonizer as discussed in Section 2.2. However, there are several problems with this methodology. The first is that it demands the storage of the received signal over a large window of time. The second is that a large processing delay is introduced due to the necessity of scanning the entire window to find the most likely estimate. Finally there is the fact that given the distribution (3.5) there is a non zero probability that preamble is not in the interval under investigation.

Alternatively the detection problem can be posed as a hypothesis testing problem, similar to those presented in Section 2.1. In hypothesis testing the H_0 and H_1 hypotheses can be tested in real time as the signal is received. This allows hypothesis testing to avoid the pitfalls of the ML formulation mentioned above. For this reason it is preferable to design the detector based on hypothesis testing principals.

3.2.2 Correlation-Based Detector

Following the same approach as the spread spectrum detector in Section 2.1.3, a chip matched filter can be used as a front-end to the detector. Temporarily assuming chip synchronicity the chip samples are despread according to (3.8). In Section 5.1 it will be shown that results obtained using this assumption do not deviate much from the actual values in a chip asynchronous system.

$$\hat{a}_m(k) = \frac{1}{\sqrt{E_c}L_b} \sum_{l=1+(m-1)L_b}^{mL_b} r(k+l-1)b_l \tag{3.8}$$

The estimated sequence $\vec{\hat{a}}(k)$ can then be differentially decoded and correlated with the known preamble sequence $\vec{c}$. The real component of the correlation can be used to create a decision statistic for the k'th chip interval η_k. The imaginary

component can be ignored because after the differential decoding it is entirely noise driven.

$$\eta_k = \sum_{m=2}^{W} \mathrm{Re}\left\{\hat{a}_m(k)\hat{a}^*_{m-1}(k)c_m\right\} \tag{3.9}$$

In an ML formulation the detector would select η_k such that $\eta_k > \eta_i$ for $i \neq k$. This can then be rephrased as a hypothesis testing problem where the H_1 hypothesis is that the k'th chip is the start of the preamble sequence and the H_0 hypothesis is that it is not. The test can be performed by comparing the decision metric η_k with a threshold G. A natural choice for the threshold is the detector output at the moment that it is fully aligned with the preamble in a noiseless scenario,

$$\begin{aligned} G &= \sum_{m=2}^{W} \mathrm{Re}\left\{a_m(k)a^*_{m-1}(k)c_m\right\} \\ &= (W-1). \end{aligned}$$

However, to allow for flexible system designs we define the threshold to be,

$$G = g(W-1), \qquad g \in [0,1).$$

In Section 4.1 it will be shown that the inclusion of the g term allows a design trade off between the two error events of the system. Thus, allowing the detector to be adopted into different network scenarios.

As the detector runs in real time on the received signal it will declare the detection event on the first $\eta_k > G$. Therefore the detector can be formulated as the system,

$$\hat{i}_0 = k = \mathrm{argmin}_k\{\eta_k \geq G\} \tag{3.10}$$

3.3 Error Events

Based on the definition of the detector in (3.10) there are two possible error events. The event $(\eta_k > G, k < i_0)$ marks the false detection of a packet before it has arrived at the receiver, and the event $(\eta_k < G, k = i_0)$ marks the event that the preamble is missed once it has arrived. Based on the definition that the detector will estimate $\hat{i}_0$ on the first event $(\eta_k > G)$ it is clear that the false detection and miss events are mutually exclusive. Therefore, denoting the probabilities of the error events as P(false) and P(miss) it is possible to formulate the probability of correct detection as,

$$\text{P(correct)} = 1 - \text{P(false)} - \text{P(miss)}. \tag{3.11}$$

In order to evaluate the probabilities of the events in (3.11) the pdf of the decision metric η_k is required. As expressed in (3.9) η_k is a sum of correlated random variables. The distribution of such a summation is difficult to express. Alternatively, it is possible to rewrite (3.9) as a quadratic form.

$$\begin{aligned} \eta_k &= \sum_{m=2}^{W} \text{Re}\left\{\hat{a}_m(k)\hat{a}^*_{m-1}(k)c_m\right\} \\ &= \frac{1}{2}\sum_{m=2}^{W} c_m\left(\hat{a}_m(k)\hat{a}^*_{m-1}(k) + \hat{a}^*_m(k)\hat{a}_{m-1}(k)\right) \\ &= \frac{1}{2}\vec{\hat{a}}^{\dagger}\mathbf{C}\vec{\hat{a}} \end{aligned} \tag{3.12}$$

Where $[.]^{\dagger}$ denotes the conjugate transpose, and $\mathbf{C}$ is the correlation matrix with entries,

$$C_{ij} = \begin{cases} c_i & i = j+1 \\ c_j & j = i+1 \\ 0 & otherwise \end{cases}. \tag{3.13}$$

Now that η_k has a matrix expression it is possible to rotate the matrix using a standard eigenvalue/eigenvector decomposition [1]. The result of which is that η_k can

be expressed as sum of independent random variables,

$$\eta_k = \frac{1}{2}\vec{a}^{\dagger}\mathbf{P}\Lambda\mathbf{P}^{\dagger}\vec{a} = \frac{1}{2}\vec{Z}^{\dagger}\Lambda\vec{Z} = \frac{1}{2}\sum_{i=1}^{W}\lambda_i|Z_i|^2. \tag{3.14}$$

Where λ_i are the eigenvalues of the matrix $\mathbf{C}$ and $|Z_i|^2$ are non-central chi-square random variables with 2 degrees of freedom, non-centrality parameters $\delta_i^2 = |\mathrm{E}[Z_i]|^2$, and $\sigma^2 = \frac{1}{2E_cL_b}$. See Appendix B for a description of chi-square random variables and their parameters.

In Appendix C it is proven that the eigenvalues λ_i depend only on the length W of the preamble sequence. Additionally closed-form expressions of both the eigenvalues and eigenvectors are derived as,

$$\lambda_i = 2\cos\left(\frac{i\pi}{W+1}\right) \tag{3.15}$$

$$\mathbf{P}_{ji} = \sqrt{\frac{2}{W+1}}\left(\prod_{k=2}^{j}c_k\right)\sin\left(\frac{ji\pi}{W+1}\right). \tag{3.16}$$

The distribution of a summation of independent random variables can be obtained by inverting the product of their respective characteristic equations. Thus, the distribution of η_k is given as,

$$f_{\eta_k}(x) = \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{\exp\left\{jt\sum_{i=1}^{W}\lambda_i\delta_i^2/(1-jt\lambda_i/E_cL_b)\right\}}{\prod_{i=1}^{W}(1-jt\lambda_i/E_cL_b)}e^{-j2xt}dt. \tag{3.17}$$

3.3.1 Probability Of Missing The Preamble

The probability of missing the preamble can be calculated by integrating the distribution (3.17) over the region $(-\infty, G)$. Unfortunately, no analytical techniques exist to compute the integral,

$$\mathrm{P(miss)} = \int_{-\infty}^{G}\frac{1}{\pi}\int_{-\infty}^{\infty}\frac{\exp\left\{jt\sum_{i=1}^{W}\lambda_i\delta_i^2/(1-jt\lambda_i/E_cL_b)\right\}}{\prod_{i=1}^{W}(1-jt\lambda_i/E_cL_b)}e^{-j2xt}dtdx. \tag{3.18}$$

The problem arises due to the inability to invert the characteristic function of a linear combination of non-central chi-square random variables. In [22] Raphaeli presents a method of using series expansions to compute the inversion. However, it is found to have convergence issues due to the eigenvalues which lead it to be unstable and unusable for this application. In [5, 3] Gil-Pelaez and Davies develop an efficient technique to numerically integrate expressions such as (3.18) by swapping the order of integrations. Denoting the characteristic function of η as $\Phi(t)$ the reformulated integral is given as,

$$\mathrm{P}(\eta < G) = \frac{1}{2} - \int_{-\infty}^{\infty} \mathrm{Im}\left(\frac{\Phi_\eta(t)e^{-jtG}}{2\pi t}\right) dt. \tag{3.19}$$

The integration can then be done using traditional numerical methods such as the trapezoid rule.

3.3.2 Bounding P(miss)

Consider for a moment the special case of $W = 2$. Evaluating formulas (3.14) and (3.15) for this case results in,

$$\lambda_1 = 2\cos\left(\frac{\pi}{3}\right) = 1 = -\lambda_2 \tag{3.20}$$

$$\eta_{i_0} = \frac{1}{2}\left(|Z_1|^2 - |Z_2|^2\right) \tag{3.21}$$

$$\mathrm{P(miss)} = \mathrm{P}(\eta_{i_0} < G) = \mathrm{P}(|Z_1|^2 - |Z_2|^2 < 2G) \tag{3.22}$$

Observe then that probability (3.22) is in an ideal format to apply the Chernoff upper bound [21].

$$\mathrm{P(miss)} = \mathrm{P}(|Z_1|^2 - |Z_2|^2 < 2G) \leq \min_{t\geq 0} e^{2Gt}\mathrm{E}[e^{-t(|Z_1|^2-|Z_2|^2)}]. \tag{3.23}$$

The expectation in (3.23) can be solved using the characteristic function of the non-central chi-square distribution:

$$\mathrm{E}[e^{-t(|Z_1|^2-|Z_2|^2)}] = \frac{\exp\{t\delta_2^2/(1-t/E_cL_b) - t\delta_1^2/(1+t/E_cL_b)\}}{(1-t/E_cL_b)(1+t/E_cL_b)}. \tag{3.24}$$

The result from (3.23) can be extended to the cases where $W > 2$. Since the eigenvalues are symmetric on the interval $(-2,2)$ the summation (3.14) can be split into two summations. Where one summation is over the positive eigenvalues and the second is over the negative ones. Furthermore, each summation can be approximated by a non-central chi-square random variable with 2 degrees by performing the following moment matching,

$$\begin{aligned}
\mathrm{E}[\zeta_1] &= \sum_{\lambda_i>0} \lambda_i \mathrm{E}[|Z_i|^2] = 2\sigma_{\zeta_1}^2 + \delta_{\zeta_1}^2 && (3.25)\\
\mathrm{var}(\zeta_1) &= \sum_{\lambda_i>0} \lambda_i^2 \mathrm{var}[|Z_i|^2] = 4\sigma_{\zeta_1}^4 + 4\sigma_{\zeta_1}^2\delta_{\zeta_1}^2 && (3.26)\\
\mathrm{E}[\zeta_2] &= \sum_{\lambda_i<0} |\lambda_i| \mathrm{E}[|Z_i|^2] = 2\sigma_{\zeta_2}^2 + \delta_{\zeta_2}^2 && (3.27)\\
\mathrm{var}(\zeta_2) &= \sum_{\lambda_i<0} \lambda_i^2 \mathrm{var}[|Z_i|^2] = 4\sigma_{\zeta_2}^4 + 4\sigma_{\zeta_2}^2\delta_{\zeta_2}^2 && (3.28)
\end{aligned}$$

Thus, $\mathrm{P}(\text{miss}) \approx \mathrm{P}(\zeta_1 - \zeta_2 < 2G)$ and the same bounding technique from (3.23) can be applied.

3.3.3 Probability Of Falsely Detecting The Preamble

The probability of falsely detecting the preamble is dependent on the start time i_0, as any $\hat{i}_0 = k, k < i_0$ results in a false detection.

$$\mathrm{P}(\text{false}|i_0) = \mathrm{P}\left(\bigcup_{k=0}^{i_0-1} \eta_k \geq G|i_0\right) \leq \sum_{k=0}^{i_0-1} \mathrm{P}\left(\eta_k \geq G|i_0\right) \tag{3.29}$$

As shown in (3.29) the union bound can be used to express the probability of false detection conditioned on i_0 as a summation of error events. Additionally, it can be assumed that these events are independent. Then using the same numerical integration technique as for P(miss) the individual probabilities can be evaluated and summed. As will be seen in Chapter 4, when compared with simulation results the union bound is very tight.

The numerical computation of (3.29) can be simplified by noting that the summation can be split into two summations,

$$\mathrm{P}(\text{false}|i_0) \leq \sum_{k=0}^{(i_0-WL_b)_+} \mathrm{P}(\eta_k \geq G|i_0) + \sum_{(i_0-WL_b)_+}^{i_0-1} \mathrm{P}(\eta_k \geq G|i_0), \tag{3.30}$$

where the first summation is across all events where the detector correlates the preamble with noise. The second summation is across all events where there is partial overlap between the received preamble sequence and the correlation sequence in the detector.

Since the events in the first summation are entirely noise driven the $|Z_i|^2$ terms in (3.14) are central chi-square random variables. The distribution of the summation of central chi-square random variables can be expressed by setting $\delta_i^2 = 0$ in (3.17),

$$f_{\eta_k}(x) = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{1}{\prod_{i=1}^{W}(1 - jt\lambda_i/E_cL_b)} e^{-j2xt}dt. \tag{3.31}$$

Using the partial fractions decomposition as in [24] the product in (3.31) can be replaced by a summation. Recognizing the sum terms as the characteristic functions of central chi-square random variables, each term can then be inverted to yield,

$$f_{\eta_k}(x) = \sum_{\lambda_r>0}\left(\prod_{i=1,i\neq r}^{W}\frac{\lambda_r}{\lambda_r-\lambda_i}\right)\frac{2E_cL_b}{\lambda_r}e^{-2xE_cL_b/\lambda_r}. \tag{3.32}$$

Equation (3.32) can then be integrated over the region $[G,\infty)$ to yield the expression,

$$\mathrm{P}(\eta_k > G|k < i_0) = \sum_{\lambda_r>0}\left(\prod_{i=1,i\neq r}^{W}\frac{\lambda_r}{\lambda_r-\lambda_i}\right)e^{-2GE_cL_b/\lambda_r}. \tag{3.33}$$

Thus, the summation over all noise driven false detections can be expressed as,

$$\sum_{k=0}^{(i_0-WL_b)_+} \mathrm{P}(\eta_k \geq G|i_0) = [(i_0 - WL_b)_+ + 1] \sum_{\lambda_r>0} \left(\prod_{i=1,i\neq r}^{W} \frac{\lambda_r}{\lambda_r - \lambda_i} \right) e^{-2GE_cL_b/\lambda_r}. \tag{3.34}$$

Combining equations (3.34) and (3.30) reduces the complexity of computing P(false$|i_0$). However, this is still a conditional probability and requires integration over the distribution of i_0 (3.5) to obtain the actual probability of false detection.

3.3.4 Approximating P(false)

The computation of P(false) can be drastically simplified by noting that due to the spreading gain L_b and the differential encoding, the correlation of the received preamble sequence and the detector preamble sequence will be negligible compared to the correlation with the received noise when the sequences are misaligned. Considering all η_k for $k < i_0$ to be entirely noise driven, equations (3.29) and (3.33) can be combined to approximate the conditional probability as,

$$\mathrm{P}(\text{false}|i_0) \approx (i_0 - 1) \sum_{\lambda_r>0} \left(\prod_{i=1,i\neq r}^{W} \frac{\lambda_r}{\lambda_r - \lambda_i} \right) e^{-2GE_cL_b/\lambda_r}. \tag{3.35}$$

This expression then integrates nicely over the distribution (3.5) to produce the general expression,

$$\mathrm{P}(\text{false}) \approx \left(\frac{1}{\gamma} - 1 \right) \sum_{\lambda_r>0} \left(\prod_{i=1,i\neq r}^{W} \frac{\lambda_r}{\lambda_r - \lambda_i} \right) e^{-2GE_cL_b/\lambda_r}. \tag{3.36}$$

Chapter 4

System Performance

This chapter investigates the effect of the key system parameters on the performance of the system. The probabilities P(miss) and P(false) are plotted based on the theory developed in Section 3.3. Simulation data is provided for verification of the theoretical results.

4.1 Effect Of g

As mentioned earlier in Section 3.2.2, g is a useful system parameter which can be used to select an acceptable level of P(miss) and P(wrong). Figure 4.1 shows the probability of missing the preamble for $g = 0.2, 0.5, 0.8$. The numerical and Chernoff plots correspond to evaluation of formulas (3.18) and (3.23) respectively. It is clear that the bound is not very tight and will be investigated further in Section 4.2. In future plots only (3.18) will be used to plot P(miss).

Note that as g increases, P(miss) increases. This is because the increase in g raises the decision threshold which makes it harder detect the preamble, hence P(miss) increases. Correspondingly Figure 4.2 shows that P(false) decreases with increasing g due to the same threshold raising effect. Thus, g can be used to decrease P(false) at the cost of increasing P(miss). This allows the system to be config-

ured based on which error event is more costly to the system in a given application.

Figure 4.2 also shows that the union bound (3.29) used to calculate P(false) is very tight as it matches the simulation data. The approximation (3.36) obtained by ignoring the self noise is very tight, within fractions of a dB, to the numerical result, even in high SNR. Thus, for the remainder of the chapter (3.36) will be used to plot P(false).

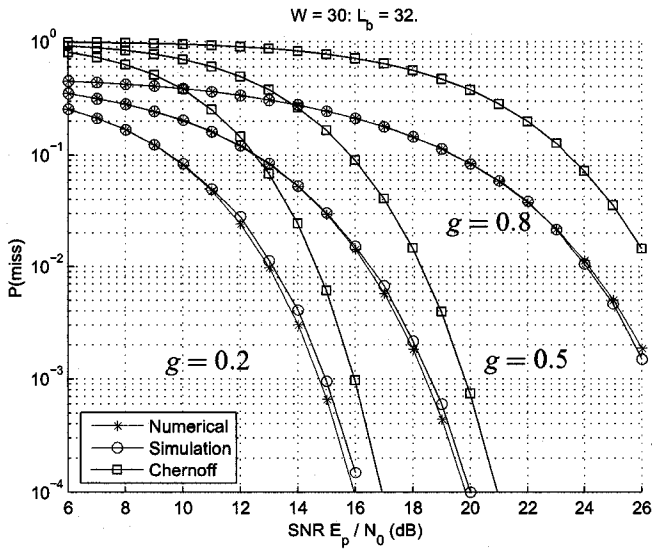

Figure 4.1: P(miss) versus SNR (measured in total preamble energy E_p over noise power N_0) for $g = 0.2, 0.5, 0.8$

When selecting a value of g for a system, the views given by Figures 4.1 and 4.2 are not very helpful because each figure plots either P(miss) or P(false). A much more convenient view is shown in figure 4.3. By plotting P(miss) vs. P(false) evaluated across the SNR region it is clear to see exactly what the trade off is for a given value of g. The SNR points used in Figure 4.3 correspond to $\frac{E_b}{N_0} = 6, 7, \ldots 26$ dB.

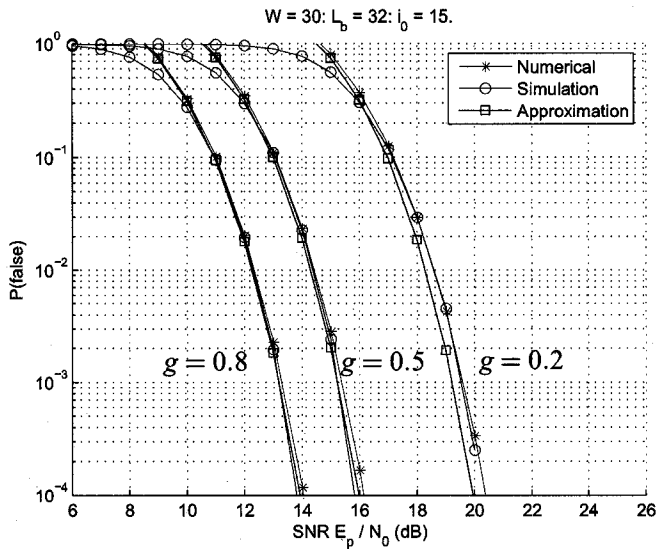

Figure 4.2: P(false) versus SNR for $g = 0.2, 0.5, 0.8$

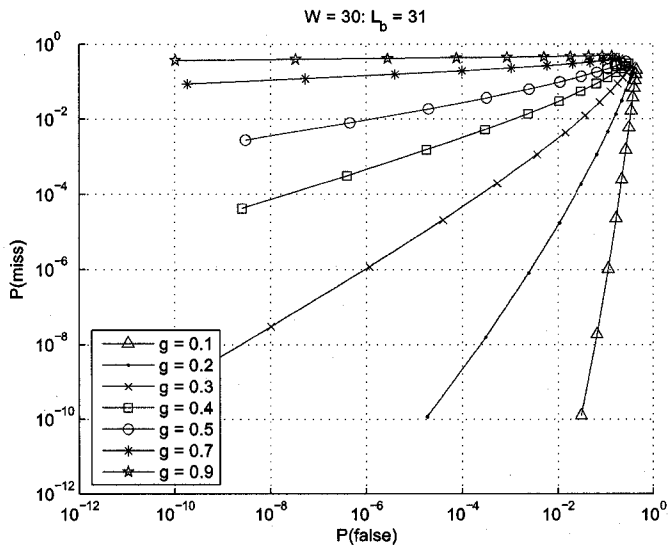

Figure 4.3: P(false) versus P(miss) for various values of g

4.2 Tightness Of The Bound On P(miss)

As seen in Figure 4.1 the Chernoff upper bound on the probability of missing the packet is not very tight. Initial thoughts would be that the error is due to the ap-

proximation obtained by moment matching in (3.25) to (3.28). However, Figure 4.4 shows that for the case when $W = 2$, and there is no approximation, the bound is still loose. Even in very high SNR where P(miss)= 10^{-10} the upper bound is greater by a factor of 10 times.

This problem arises because the Chernoff bound is meant to provide a tight bound on the tails of the density function. However, when evaluating the miss probability the bound is applied over the region $(-\infty, G)$ which includes a considerable portion of the actual density. Thus, the bound becomes very loose in order to fit the pdf. Which leads to the conclusion that it is inappropriate to bound P(miss) in this manner.

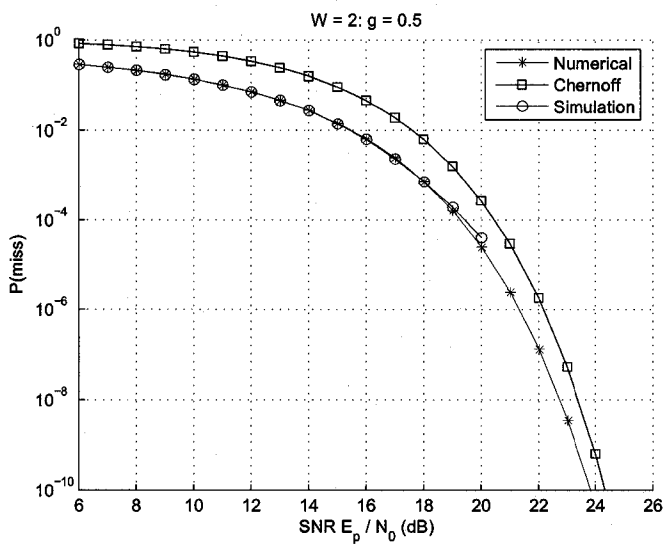

Figure 4.4: P(miss) versus SNR for the special case $W = 2$

4.3 Effect Of Δf

As mentioned earlier in Section 3.1.2, the despreading of the received signal in (3.8) inherently assumes that the chips can be coherently added across the block. For this

reason it was stated that the spreading length L_b be selected such that $\Delta f L_b T_c \ll 1$ for the frequency offset to be negligible. We definine the term $\theta = \Delta f L_b T_c$ to denote the rotation of the signal constellation across each preamble symbol. Figure 4.5 shows the effect of the frequency offset on the probability of missing the preamble for $\theta = 0, \frac{1}{4}\frac{1}{8}\frac{1}{16}$. It is clear that for small offsets there is only a slight shift in performance, but if the offset increases it is almost impossible to detect the preamble.

Practical systems typically operate with carrier frequencies in the GHz range. Therefore, with an oscillator drift in the ballpark of 0.1 ppm the resulting frequency offset is roughly 100 Hz. For the same carrier in a mobile scenario with velocities of up to 300 km/h the maximum resulting Doppler shift is 300 Hz. If the two mechanisms operate together, a fair estimate of the total frequency offset is $\Delta f = 500$ Hz. With a symbol rate of 10^6 symbols/s the corresponding rotation is $\theta = 5\text{x}10^{-4}$. Clearly, in a practical system $\theta \ll 1$ and the effect of the frequency offset will be negligible.

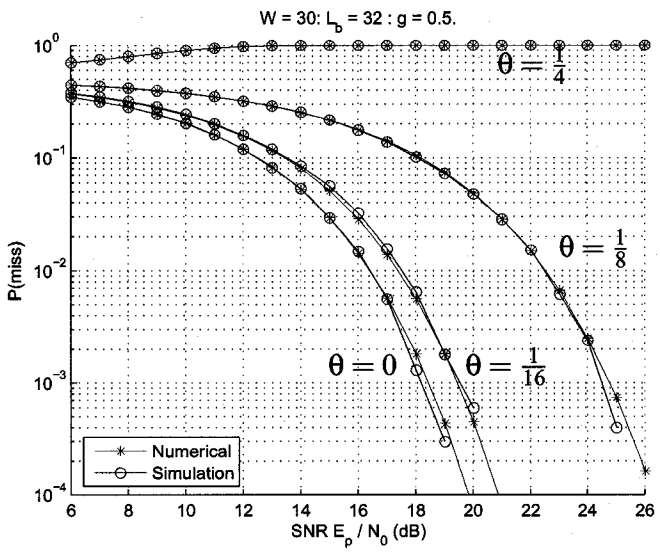

Figure 4.5: P(miss) versus SNR for $\theta = 0, \frac{1}{4}\frac{1}{8}\frac{1}{16}$

The frequency offset has little effect on P(false) because as already determined P(false) is mainly noise driven. The presence of a frequency offset actually helps the approximation of ignoring partial over lap of the received sequence with the correlator since the signal cannot be coherently despread.

4.4 Effect Of L_b

As stated in Section 3.1.2 the total preamble power is denoted $E_p = WL_bE_c$. This can be manipulated into the form $E_p/W = L_bE_c$ and substituted into the expressions of P(miss) and P(false) to yield,

$$\text{P(miss)} = \int_{-\infty}^{G} \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{\exp\left\{jt\sum_{i=1}^{W}\lambda_i\delta_i^2/(1-jt\lambda_iW/E_p)\right\}}{\prod_{i=1}^{W}(1-jt\lambda_iW/E_p)} e^{-j2xt}dtdx \quad (4.1)$$

$$\text{P(false)} = \left(\frac{1}{\gamma}-1\right)\sum_{\lambda_r>0}\left(\prod_{i=1,i\neq r}^{W}\frac{\lambda_r}{\lambda_r-\lambda_i}\right)e^{-2GE_p/W\lambda_r} \quad (4.2)$$

Actually, equation (4.2) can be further reduced by replacing $G = g(W-1)$.

$$\begin{aligned}\text{P(false)} &= \left(\frac{1}{\gamma}-1\right)\sum_{\lambda_r>0}\left(\prod_{i=1,i\neq r}^{W}\frac{\lambda_r}{\lambda_r-\lambda_i}\right)e^{-2g(W-1)E_p/W\lambda_r} \\ &= \left(\frac{1}{\gamma}-1\right)\sum_{\lambda_r>0}\left(\prod_{i=1,i\neq r}^{W}\frac{\lambda_r}{\lambda_r-\lambda_i}\right)e^{-2gE_p/\lambda_r} \quad (4.3)\end{aligned}$$

From (4.1) and (4.3) it is clear to see that the error events depend only on the total preamble energy E_p, the length of the preamble sequence W, and the fraction g chosen for the threshold. Therefore, the only benefit achieved by the spreading is that it can be used to suppress the MAI and channel noise.

4.5 Effect Of W

By examining (4.1) and (4.3) it is not clear what the effect of W is on the error events. By definition $W \geq 2$ due to the differential encoding. To get an idea of the

effect of W, Figures 4.6 and 4.7 show the error events for the cases $W = 2, 15, 30$. In both cases the probability of error is increased with W. Although the effect on P(miss) is rather small, it only deviates at most 1 dB, while the effect on P(false) is much more severe.

However, this comparison has been done for a fixed value of preamble energy, which inherently biases against longer sequences because there is less energy distributed to each preamble symbol $E_b = \frac{E_p}{W}$. A more legitimate comparison is shown in Figures 4.8 and 4.9 where the SNR measure used is in terms E_b/N_0. Here, it is seen that increasing W leads to considerable performance gains. This, is a rather intuitive result given an understanding for the mechanisms that generate the error events.

As earlier mentioned, increasing the number of preamble symbols while keeping the energy per symbol constant implies that there is more energy in the preamble to detect, thus making it harder to miss the packet. It must be noted that, increasing the number of preamble symbols is not as effective as increasing the energy per symbol due to the differential decoding done on the symbol level. For every preamble symbol the detector must decode there is a multiplication $\hat{a}_m \hat{a}^*_{m-1}$ which causes a noise enhancement. Hence, the results from Figure 4.6 which favor shorter sequences for fixed E_p.

False detections occur when the channel noise replicates the preamble sequence. With increased sequence length the preamble becomes more unique and the probability of noise being able to replicate it is reduced. However, in the case where the comparison is made for a constant preamble energy, the reduction in E_b causes the estimated preamble bits $\hat{a}_m$ to be much noisier. Thus, allowing for the detector to pick up considerably higher amounts of energy than expected which increases the

false detection probability.

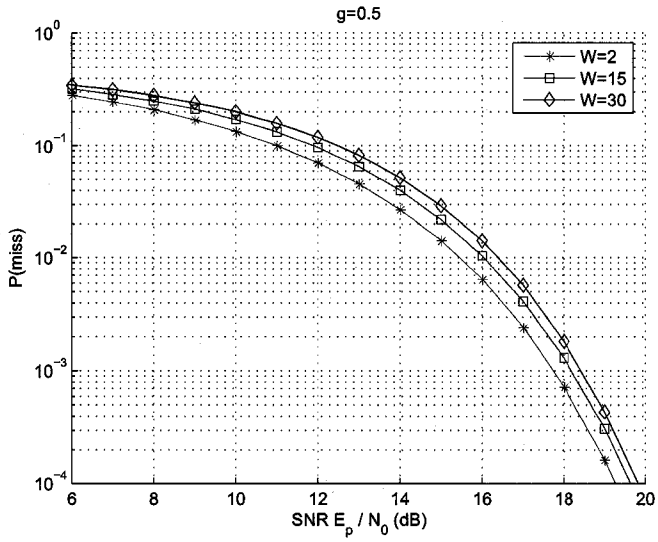

Figure 4.6: P(miss) versus SNR for $W = 2, 15, 30$

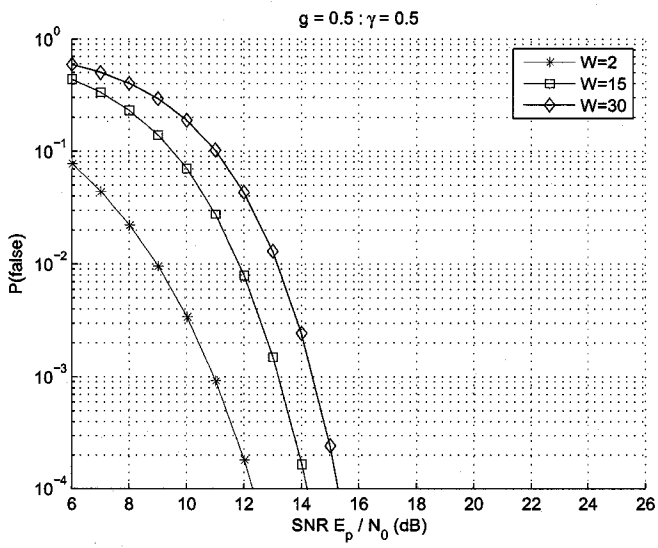

Figure 4.7: P(false) versus SNR for $W = 2, 15, 30$

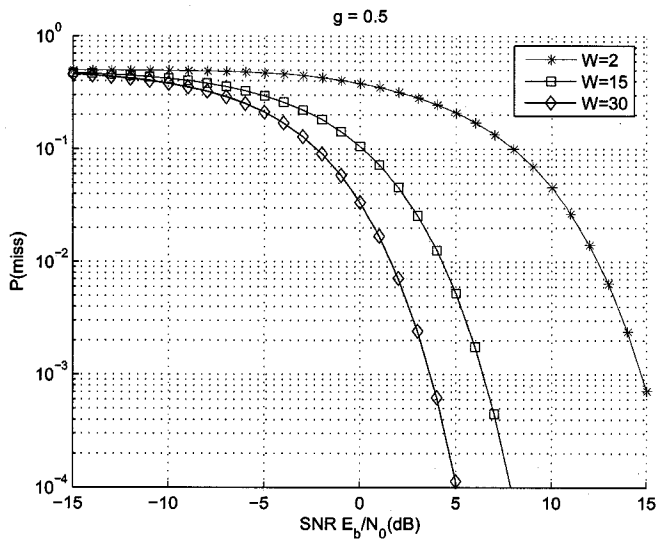

Figure 4.8: P(miss) versus SNR (E_b/N_0) for $W = 2, 15, 30$

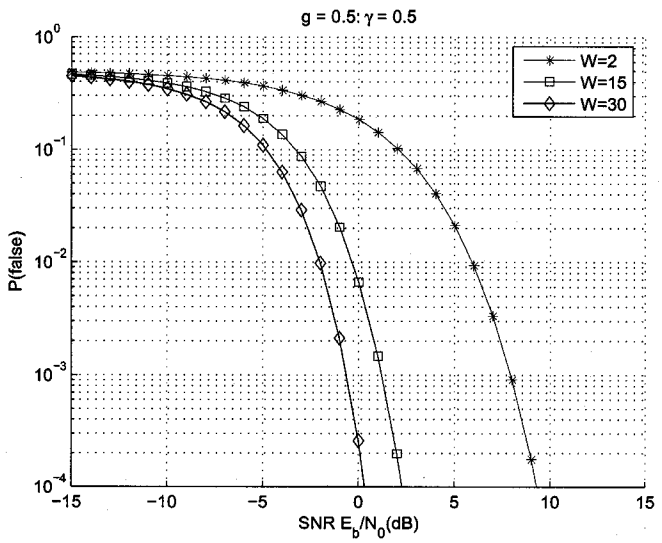

Figure 4.9: P(false) versus SNR (E_b/N_0) for $W = 2, 15, 30$

Chapter 5

Practical Considerations

5.1 Chip Asynchronous Detection

In the development of the detector in Section 3.2.2 it was assumed that by placing a matched filter at the front end the rest of the system can than operate on samples at the chip interval. This assumption implies that the transmitter and receiver are synchronized at the chip level. In an uncoordinated network where the use of a preamble is necessary that kind of synchronization does not exist. Part of the task of the preamble is to adjust the timing offset between the chip clocks on both ends.

The timing offset between the transmitter and receiver can be any where on the interval $[-\frac{T_c}{2}, \frac{T_c}{2})$. Thus, it is modelled as a uniform random variable,

$$f_\tau(\tau) = \begin{cases} \frac{1}{T_c} & |\tau| \leq \frac{T_c}{2} \\ 0 & otherwise \end{cases} . \tag{5.1}$$

The effect of the timing offset on the matched filter output is that chip samples will be given as,

$$\hat{d}_k = d_k \mathrm{R}(\tau) + \sum_{j \neq k} d_j \mathrm{R}(\tau(j-k)T_c) + n_k. \tag{5.2}$$

Where $\mathrm{R}(\tau)$ is the pulse autocorrelation function evaluated at the random offset. Due to the imposed Nyquist criterion on the pulse to be used in Section 3.1.2 the

autocorrelation function must obey the property,

$$\mathrm{R}(kT_c) = \begin{cases} 1 & k = 0 \\ 0 & |k| = 1,2,\ldots,\infty \end{cases} . \tag{5.3}$$

Thus, if τ can be reduced enough such that $\mathrm{R}(\tau) \approx 1$ the effect of the time offset should become negligible enough for the results and analysis of Chapters 3 and 4 to hold.

By over-sampling the received signal by a factor L_s the uncertainty range for the timing offset must shrink by a factor L_s. Thus, the definition of the offset will be,

$$f_\tau(\tau) = \begin{cases} \frac{L_s}{T_c} & |\tau| \le \frac{T_c}{2L_s} \\ 0 & otherwise \end{cases} . \tag{5.4}$$

It is possible to chose L_s such that $\mathrm{R}(\tau) \approx 1$, and $\mathrm{R}(\tau + kTc) \approx 0$ for the given pulse. Thus, one of the L_s samples per chip can be considered chip synchronous.

The detector can now be used to compute the decision metric η_s at the sample level. However, due to the possibility of having more than one $\eta_s > G$ in a chip interval, especially in the chip interval corresponding to i_0, it is no longer sufficient to define the detection event as the first hypothesis $\eta_s > G$. A better definition would be to select the first η_s such that $\eta_s > \{G, \eta_{s+1}\}$. This definition ensures that the detection event occurs at the peak of the correlation output.

With the new definition of a detection event it is important to redefine what a correct detection is, and what the new error events are. A correct detection is an estimate $\hat{i}_0$ that lies within one chip duration $\pm T_c$ of i_0. The miss event then is the absence of a detection event in that same interval, and a false detection is any detection event prior to that interval.

5.2 Performance Results

The asynchronous system is simulated using the new definitions of the detection events. The pulse used is a root raised cosine with roll off factor $\beta = 0.5$, with over-sampling factor $L_s = 2$. Figure 5.1 shows the plots of both P(miss) and P(false) and compares them with their counterparts for a chip synchronous system.

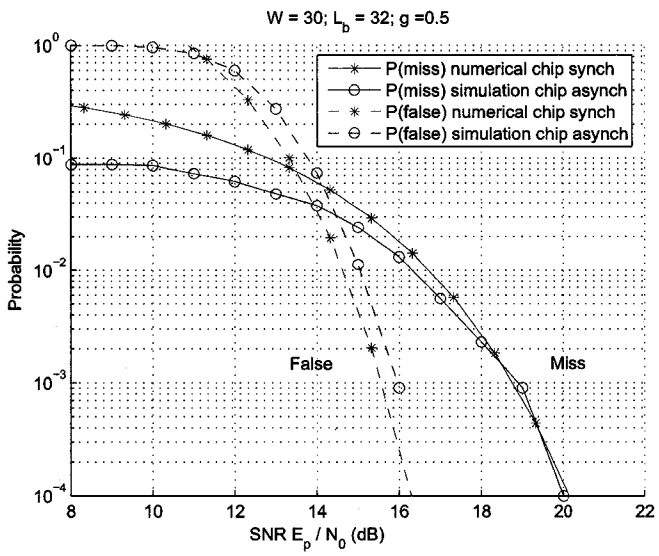

Figure 5.1: Simulation results for an asynchronous system compared to the numerical results of a synchronous system

In the low SNR regime, the chip asynchronous system provides a performance gain in terms of the miss probability. This is due to the fact that when the system is mainly noise driven the distributions of the η_s within the correct detection range are similar. Thereby providing $2L_s$ equal opportunities to correctly detect the packet. However, as the SNR increases the sample η_s that is closest to being chip synchronous starts to dominate and the system approaches the performance of a synchronous system.

The asynchronous systems suffers from a slight performance loss in terms of the

false detection probability. As earlier indicated in Sections 3.3.4 and 4.1 the false detection events are purely noise driven. Thus, the effect of asynchronous sampling does not really effect the false detection events. However, the over sampling by L_s creates a factor L_s more possible detection events that could result in false detections. Thus, the false detection probability scales by L_s due to the additional events.

5.2.1 Post Detection Offset

Now that it is clear that the initial timing offset does not cause the performance of the system to deviate far from chip synchronous case. An interesting question to ask is what is the distribution of the post detection timing offset?

Given (5.2), it is possible to express η_s as a function of the offset,

$$\eta_s = \sum_{m=2}^{W} (\mathrm{R}(\tau)a_m + n'_m)(\mathrm{R}(\tau)a_{m-1} + n'_{m-1})c_m. \tag{5.5}$$

Where the frequency and phase offsets have been removed for ease of notation. Also, the ISI terms arising from the chip asynchronicity have been group with the noise terms to give the n'_m terms. Then applying the same rotation technique as before (5.5) can be expressed as,

$$\eta_s = \sum_{i=1}^{W} \lambda_i |Z_i|^2. \tag{5.6}$$

Clearly the eigenvalues will remain the same, as they were set by the differential structure and have no relation to the received signal. However, due to the scaling by $\mathrm{R}(\tau)$ the non-centrality parameters of the chi-square terms will now be scaled by a factor $\mathrm{R}^2(\tau)$; $s_i^2 = \mathrm{R}^2(\tau)\delta_i^2$. Where δ_i^2 is the corresponding value from the chip synchronous case. Thus, the same inversion technique from Section 3.3.1 can be used to obtain the distribution of η_s for any given offset.

For the case of $L_s = 2$ there will be 4 possible η_s in the range corresponding to a correct detection. Figure 5.2 plots the ranges of these events, denoted as A,B,C,D. Also included on the plot is the value of $R^2(\tau)$ to show how the non-centrality parameters scale with τ.

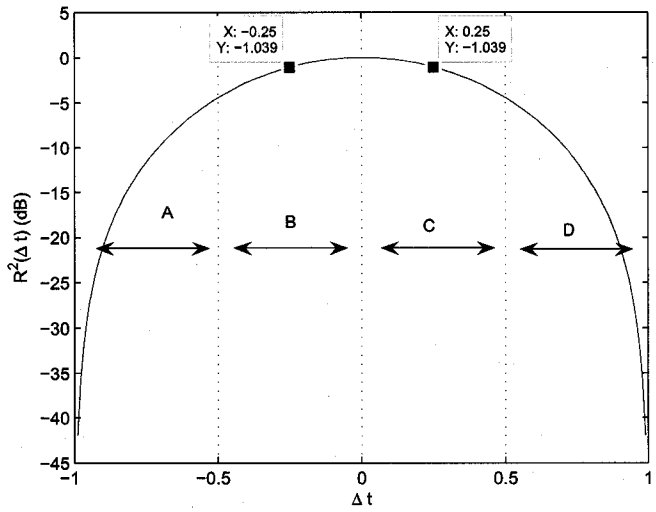

Figure 5.2: Division of the detection range for the case $L_s = 2$

For the post detection offset to be in the range A, it requires that $\eta_A > G, \eta_B, \eta_C, \eta_D$. Likewise for the offset to lie in range B require $\eta_B > G, \eta_A, \eta_C, \eta_D$. This argument can be extended to get the piecewise distribution of the post detection offset,

$$f_\tau(\tau) = \begin{cases} \int_G^\infty f_{\eta|\tau}(x) F_{\eta|\tau+\frac{T_c}{2}}(x) F_{\eta|\tau+T_c}(x) F_{\eta|\tau+\frac{3T_c}{2}}(x) dx & -1 \le -\tau < \frac{T_c}{2} \\ \int_G^\infty f_{\eta|\tau}(x) F_{\eta|\tau-\frac{T_c}{2}}(x) F_{\eta|\tau+\frac{T_c}{2}}(x) F_{\eta|\tau+T_c}(x) dx & -\frac{T_c}{2} \le \tau < 0 \\ \int_G^\infty f_{\eta|\tau}(x) F_{\eta|\tau-T_c}(x) F_{\eta|\tau-\frac{T_c}{2}}(x) F_{\eta|\tau+\frac{T_c}{2}}(x) dx & 0 \le \tau < \frac{T_c}{2} \\ \int_G^\infty f_{\eta|\tau}(x) F_{\eta|\tau-\frac{3T_c}{2}}(x) F_{\eta|\tau-T_c}(x) F_{\eta|\tau-\frac{T_c}{2}}(x) dx & \frac{T_c}{2} \le \tau < 1 \end{cases} \quad (5.7)$$

Where $f_{\eta|\tau}(x)$ and $F_{\eta|\tau}(x)$ are the pdf and cdf of η for the given offset τ. These can be obtained numerically using the Gil-Palez technique [5] as discussed in Section 3.3.1. The density (5.7) can then be numerically evaluated, and is shown in

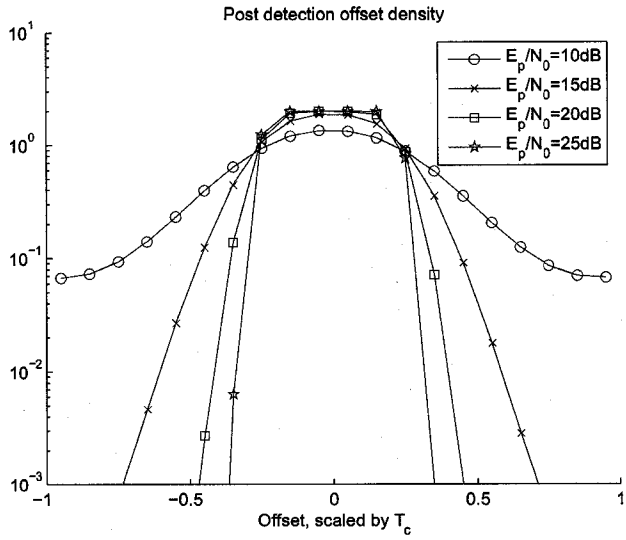

Figure 5.3: Distribution of the post detection timing offset

Figure 5.3. For moderate values of SNR it is seen that the distribution is uniform in the region $(-\frac{T_c}{4}, \frac{T_c}{4})$. As seen from Figure 5.2 the energy loss associated with this kind of offset is approximately 1 dB for the root raised cosine with roll of factor $\beta = 0.5$, which is almost negligible in typical wireless systems. Additionally, there are many post-detection signal processing techniques available to further reduce offsets of this order, such as interpolation filtering [18]. Therefore, in the chip asynchronous scenario, the detector performs joint chip synchronization, frame synchronization, and packet detection.

5.3 Sequence Selection

For the discussion until now it has been assumed that any random sequence $\vec{c}$ can be used for the preamble and any random $\vec{b}$ is used for the spreading sequence, where $\vec{c}$ has W symbols and $\vec{b}$ has WL_b bits. When considering a practical system it is possible to impose additional constraints on the sequences which will allow for

more efficient implementations.

With a random WL_b bit spreading sequence it is necessary to use a filter with WL_b taps to perform the despreading operation. However, if for example the spreading sequence is made to be periodic, a reduced complexity filter can be designed and re-used. A natural choice is to make the sequence L_b periodic, such that the same sequence can be re-used on each preamble symbol. Now the despread filter needs only L_b taps, which drops the computational complexity by a factor of W.

As a result of all symbols using the same spreading sequence, the detector will periodically become aligned to the spreading sequence of the received signal. At these moments, it will pick up the energy of all the preamble symbols present in the received signal. Thus, the assumption of the self noise being negligible may not hold true any longer. In order to keep the self noise down it is very important to select the preamble sequence $\vec{c}$ such that it has very good cross-correlation properties so that it can suppress the energy picked up by the despreading.

The m-sequences would be a likely candidate for good preamble sequences, however, they can only be used if the sequence length $W = 2^m - 1$ for some $m \geq 2$. It is possible to run a search over all possible 2^W sequences and select the one with the best cross-correlation values. However, even for reasonable values W such a search will take considerable time. When discussing false detections in Section 4.5 it was stated that the main source of false detections was attributed to noise being able to replicate the preamble sequence. Thus, if the sequence can be made to look very random, the randomness will help to reduce the probability that neighbouring symbols can be mistaken to be part of the sequence. Hence, the original definition of the preamble symbols to be i.i.d. random variables with probability $\mathrm{P}(c_m = 1) = p$ from Section 3.1.2. Figure 5.4 plots the false detection probability

due to self-noise for random sequences with $p = 0.5, 0.6, 0.7$. Additionally, the approximation of P(false) by ignoring the self noise is plotted for comparison, and so is the performance of a sequence selected based on the criteria of minimizing the maximum cross-correlation value. For all cases, a 15 bit m-sequence is used as the spreading sequence. The sequence obtained via the sequence selection is,

$$\vec{c} = [1,1,1,1,1,-1,-1,1,-1,1]. \tag{5.8}$$

The results from Figure 5.4 are fairly intuitive. In order to maximize the randomness of the sequence it is good to select $p \approx 0.5$. If it is higher or lower it will be dominated by a high amount of $c_m = 1$, or $c_m = -1$ respectively. Additionally, the cross-correlation properties of the m-sequence provide good suppression of the self noise but are only valid if adjacent symbols $a_m a_{m-1}$ have the same sign. Thus, to exploit the benefit it is clear that a good choice for p is in the range $0.5 < p < 0.5 + \varepsilon$. This allows the sequence to appear highly random, while still exploiting the properties of the spreading sequence used. However, if longer sequences are used the effects of sequence selection are diminished and all random sequences will provide similar performance.

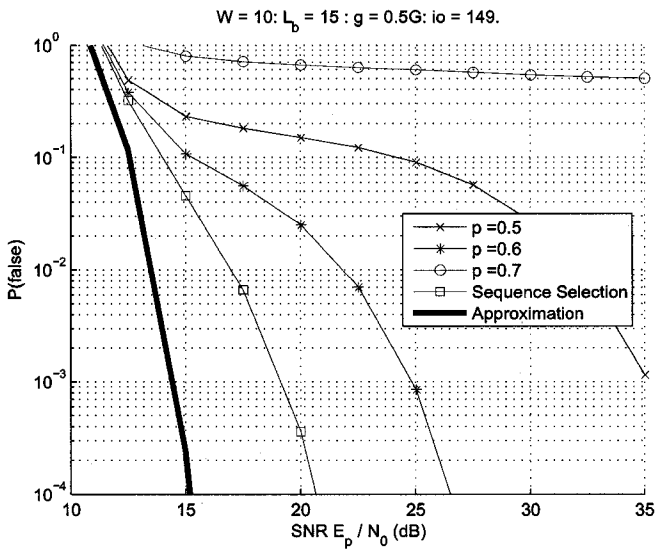

Figure 5.4: P(false) versus SNR for a periodic spreading sequence

Chapter 6

Conclusions And Future Directions

This thesis approached the fundamental physical layer problems of packet detection and frame synchronization for asynchronous wireless networks. By examining classical solutions to these problems it was found that they are quite similar. In essence both are searching the received signal for the presence of a certain sequence. However, these operations are done serially. First the packet is detected, and then a synchronization algorithm is run.

This thesis adopted a modified view of the packet structure such that the detection and synchronization problems could be combined into one detection problem. It was found that the derivation of an optimal ML solution for this problem is both very difficult to express analytically, and would be require the storage and processing of a very large data window; so large that it cannot be implemented in a realistic system. Thus, a correlation based detector is derived using the principals of hypothesis testing.

The developed detector was shown to have two error events. One where the presence of a packet is falsely detected before a packet arrives and the other when a packet has arrived and is not detected. Further, these results are shown to be mutually exclusive which allows the probability of a correct detection to be computed as

$1 - \mathrm{P}(false) - \mathrm{P}(miss)$. Analytical expressions are derived for computing the error probabilities. The derivation of the error probabilities requires that the system be decomposed into its eigenvalues and eigenvectors, and the eigenvalues are shown to be dependent only on the length of the preamble sequence.

It is also shown that the detector can be implemented in practical systems with reasonable computational complexity. With as little as 2 samples per chip the detector performs the joint packet detection and frame synchronization to within a quarter of a chip accuracy. Additionally a methodology is provided to aide in selecting sequences for preamble design.

This thesis has focused on the AWGN channel with additional frequency and phase offsets. However, it was shown in Chapter 4 when examining the approximation for the false detection probability, that the self noise due to partial correlation between the received preamble sequence and the detector is negligible compared to the correlation with the received noise. This shows that the detector will perform well in a multipath channel as it will be able to detect each path separately and thus allow for the use of diversity combining techniques later in the receiver processing. Additionally, the detector can be extended for use in fading channels. Slow fading channels where the rotation introduced by the fading is relatively constant across the preamble sequence, can be viewed as the AWGN channel with a phase offset. Due to the differential encoding of the preamble symbols, this phase offset is easily removed during the detection. Even in fast fading environments the differential encoding of the preamble symbols still provides protection provided that the time between adjacent symbols a_m remains within the channel coherence time.

Bibliography

[1] Howard Anton and Chris Rorres. *Elementary Linear Algebra.* John Wiley and Sons, 2000.

[2] R.H. Barker. *Group Synchronizing of Binary Digital Systems; Communication Theory; Ed. Jackson Willis.* Butterworths Scientific Publications, London, 1953.

[3] R. B. Davies. Numerical inversion of a characteristic function. *Biometrika*, 60(2):415–417, 1973.

[4] Dariush Divsalar and Marvin K. Simon. Multiple-symbol differential detection of mpsk. *IEEE Transactions on Communications*, 38(3):300–308, March 1990.

[5] J. Gil-Pelaez. Note on the inversion theorem. *Biometrika*, 38:481–482, 1951.

[6] Robert Gold. Optimal binary sequences for spread spectrum multiplexing. *IEEE Transactions on Information Theory*, 13(4):619–621, Oct 1967.

[7] Andrea J. Goldsmith and Stephen B. Wicker. Design challenges for energy-constrained ad hoc wireless networks. *Wireless Communications, IEEE*, 9(4):8–27, 2002.

[8] Soloman W. Golomb. *Shift Register Sequences.* Holden-Day, San Francisco, 1967.

[9] Allan Gut. *An Intermediate Course in Probability.* Springer, NY, 1995.

[10] Simon Haykin. *Communication Systems.* John Wiley and Sons, New York, 1978.

[11] Carl W. Helstrom. *Statistical Theory of Signal Detection*. Pergamon Press, 1960.

[12] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System : theory and practice*. Springer-Verlag, New York, 2001.

[13] T. Kasami. Weight distribution formula for some class of cyclic codes. *Coordinated Science Lab., Univ. IL, Urbana. Tech Rep.*, R-285, April 1966.

[14] Alberto Leon-Garcia and Indra Widjaja. *Communication Networks: Fundamental Concepts and key Architectures*. McGraw-Hill, New York, 2004.

[15] J.C. Mason and D.C. Handscomb. *Chebyshev polynomials*. Chapman and Hall, CRC, Boca Raton, Fla, 2003.

[16] James L. Massey. Optimum frame synchronization. *Communications, IEEE Transactions on*, 20(2):115–119, 1972.

[17] Ralf Mehlan and Heinrich Meyr. Optimum frame synchronization for asynchronous packet transmission. *Communications, IEEE International Conference on*, 2:826–830, 1993.

[18] Heinrich Meyr, Marc Moeneclaey, and Stefan A. Fetchel. *Digital Communication Recievers: Synchronization, Channel Estimation, and Signal Processing*. John Wiley and Sons, New York, 1998.

[19] A. Papoulis and S. U. Pillai. *Probability Random Variables and Stochastic Processes*. McGraw-Hill, NY, 2002.

[20] P. B. Patnaik. The non-central χ^2 - and f-distribution and their applications. *Biometrika*, 36(1/2):202–232, 1949.

[21] J.G. Proakis. *Digital Communications*. McGraw-Hill, New York, 2001.

[22] Dan Raphaeli. Distribution of noncentral indefinite quadratic forms in complex normal variables. *IEEE Transactions on Information Theory*, 42(3):1002–1006, May. 1996.

[23] T. J. Rivlin. *Chebyshev Polynomials: From Approximation Theory to Algebra*

and Number Theory. Wiley, New York, 1990.

[24] C. Schlegel. Trellis coded modulation on time-selective fading channels. *Communications, IEEE Transactions on*, 42:1617 – 1627, April. 1994.

[25] Gordon L. Stuber. *Principles of Mobile Communication*. Kulwar Academic Publishers, Massachusetts, 2001.

[26] Andrew J. Viterbi. *CDMA: Principles of Spread Spectrum Communication*. Addison Wesley, Massachusetts, 1995.

[27] Liuqing Yang and Georgios B. Giannakis. Ultra-wideband communications: an idea whose time has come. *Signal Processing Magazine, IEEE*, 21:26–54, 2004.

Appendix A

Spread Spectrum Communication

Spread spectrum systems are communication systems that have a much larger transmission bandwidth than information bandwidth. Originally used by the military during WWII due to its resilience to jamming and its reduced probability of detection by unintend listeners [21], today spread spectrum techniques are being used in many successful commercial applications such as GPS [12], mobile cellular networks [25,26], the 802.11 wifi protocol, and the emerging ultra-wideband systems [27]. As will be shown in sections A.2 and A.3 the ability to provide spreading gain and reduce multiple access interference (MAI) it is likely that future ad hoc networks will adopt spread spectrum techniques.

A.1 Spread Spectrum System Model

The basics of spread spectrum communication can be seen in Figure A.1. The only difference from traditional communication systems is the insertion of the spreading sequence. The spreading sequence consists of a series of short duration symbols called chips. For this reason the spreading sequence is also sometimes referred to as the chipping sequence.

On the transmitter side, each of the symbols coming out of the channel encoder

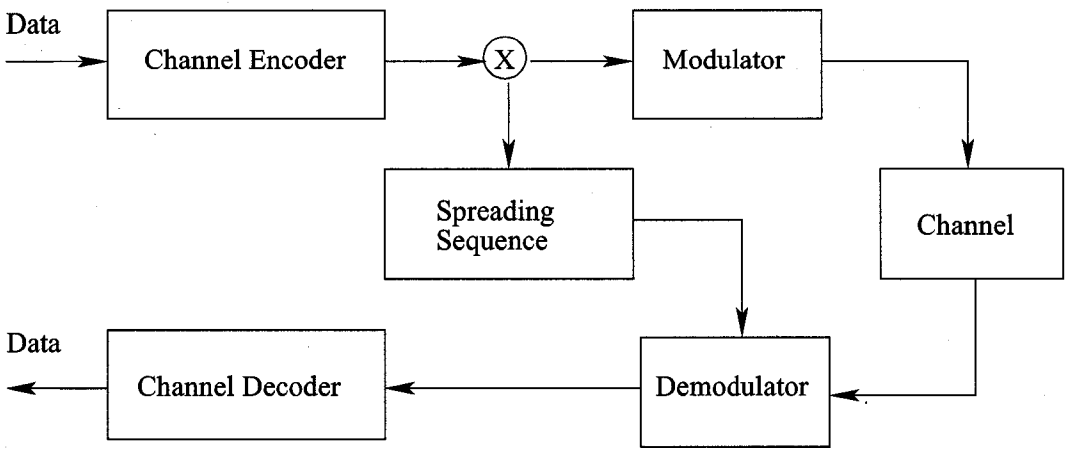

Figure A.1: Block diagram for spread spectrum communication systems

is multiplied with the entire spreading sequence. This has the effect of breaking the symbol into a series of chips which spreads the bandwidth of the system, hence the name spread spectrum. The modulator then operates at the chip level on their product. Typically either PSK or FSK modulators are used. If the modulator is PSK the system is called direct sequence spread spectrum (DSSS), if it is FSK the system is called frequency hopping spread spectrum (FHSS). Of interest in this thesis, is the DSSS case with BPSK modulation.

A.2 Spreading Gain

For the case of DSSS with BPSK modulation the transmitted signal can be expressed as,

$$s(t) = \sqrt{E_c} \sum_{n=1}^{N} b_n \sum_{l=1}^{L} c_l p(t - nT_b - lT_c) \cos(2\pi f t) \qquad \text{(A.1)}$$

Where $b_n, c_l \in \{1, -1\}$ correspond to the output bits of the channel encoder and the chips from the spreading sequence. T_b and T_c denote the bit and chip times respectively. Their ratio $L = T_b/T_c$ is the length of the chipping sequence. $p(t)$ is

the modulation pulse used for transmission, E_c is the transmit energy, and ω is the transmission frequency.

For communication over the additive white Gaussian noise channel (AWGN) the received signal has the form,

$$r(t) = s(t) + n(t). \tag{A.2}$$

Where $n(t)$ is a spectrally flat Gaussian noise process. It can be easily shown that the optimum receiver will down convert the signal and run it through a chip matched filter [21]. To remove the effect of the spreading, ie despread the signal, the chip samples are multiplied with their corresponding values from the chipping sequence and combined.

$$\hat{b_n} = \sum_{l=1}^{L} r_l c_l \tag{A.3}$$

$$= \sqrt{E_c} L b_n + \sum_{l=1}^{L} c_l n_l. \tag{A.4}$$

The despread samples are then passed into the channel decoder. By examining (A.4) it is easy to see that the SNR of the signal going into the decoder is LE_c/N_0 which is a factor L times greater than the transmission SNR of the same system. That is why the length of the spreading sequence L is also referred to as the spreading or processing gain. The ability of spread spectrum system to levarage spreading gain is highly advantagous in many applications where either the decoder cannot provide satisfactory BER on the transmission SNR, or the signal needs to be buried in the noise floor to avoid detection or accommodate multiple user communication.

Note that, although the discussion here has been for the special case of DSSS with BPSK modulation, this result is generic to all spread spectrum systems. It is easy to follow the same steps for systems using MPSK (A.5) and MFSK (A.6) as

the transmitted signals.

$$s(t) = \sqrt{E_c}\sum_{n=1}^{N} b_n \sum_{l=1}^{L} \cos(2\pi f t + c_l \pi) \tag{A.5}$$

$$s(t) = \sqrt{E_c}\sum_{n=1}^{N} b_n \sum_{l=1}^{L} \cos[2\pi(f + c_l \Delta f)t] \tag{A.6}$$

A.3 Spreading Sequences

In order to achieve spreading gain it is sufficient to use any random spreading sequence. However, poor choices of spreading sequences can cause severe degradations to system performance over more compilcated channel models. Consider now a multiple access channel, with M users, each with distinct spreading sequences, the received baseband signal after despreading of the $k'th$ user is,

$$\hat{b_n}(k) = \sqrt{E_c} L b_n(k) + \sum_{m \neq k}^{M} b_n(m) \sum_{l=1}^{L} c_l(k) c_l(m) + \sum_{l=1}^{L} c_l(k) n_l(k). \tag{A.7}$$

Equation (A.7) can be considered to be (A.4) with an additional multiple access inteference (MAI) term. If M is large enough the MAI term can easily kill any effects of spreading gain, when poor sequences are chosen.

An initial approach would be be to select spreading codes that are orthogonal to each other such that the MAI term will become reduce to zero. However, even (A.7) can be further complicated in multipath channels. Each path will then add addtional MAI terms. Thus, codes with low cross and auto-correlation values are desirable as they will reduce the MAI and self noise due to multipath. The m-sequences designed by Golomb [8], along with the Barker [2], Gold [6], Kasami [13], and Walsh-Hadamard [25] sequences are all well known sequences with good correlation values.

Of interest in this thesis are the m-sequences due to their auto-correlation properties. m-sequences are sequences of length $n = 2^m - 1$ with auto-correlation,

$$\mathrm{R}[s] = \begin{cases} n & s = 0 \\ -1 & otherwise \end{cases} . \tag{A.8}$$

The cross correlation properties of DSSS signals are exploited in cellular networks such as IS-95 and CDMA2000 for user selection and MAI suppression. Additionally, they are used in the GPS navigation system to provide distance estimates from the GPS satellites to the receiver [12].

Appendix B

Probability Theory

B.1 Characteristic Function

The characteristic function of a random variable x with distribution $f_x(x)$ is defined as [19],

$$\Phi_x(t) = \int_{-\infty}^{\infty} f_x(x) e^{jtx} dx = \mathrm{E}[e^{jtx}]. \tag{B.1}$$

Given a characteristic function the inversion formula can be used to convert it back to a distribution function,

$$f_x(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \Phi_x(t) e^{-jtx} dt. \tag{B.2}$$

There are many properties and uses for characteristic functions. One of which is for obtaining the distributions of sums of independent random variables.

Consider the two independent random variables x and y with distributions $f_x(x)$ and $f_y(y)$ respectively. Now define a third random variable $z = x + y$. The characteristic function of z is then,

$$\Phi_z(t) = \mathrm{E}[e^{jtz}] = \mathrm{E}[e^{jt(x+y)}] = \mathrm{E}[e^{jtx} e^{jty}] = \mathrm{E}[e^{jtx}]\mathrm{E}[e^{jty}] \tag{B.3}$$

Equation (B.3) shows the property that the characteristic function of the sum of any random variables is equal to the product of the independent characteristic functions.

The distribution can then be obtained using the inversion formula (B.2).

Another use of the characteristic equation is that it can used to calculate the moments of a variable. It is a clear extension from (B.1) that,

$$\mathrm{E}[x^n] = \frac{1}{j^2}\frac{d}{d^n t}\Phi_x(t)|_{t=0} \tag{B.4}$$

B.2 Chi-Square Distribution

The chi-square distribution is a very useful distribution which is derived from the Gaussian distribution. If x is Gaussian with the distribution $\mathrm{N}(m_x, \sigma^2)$, then $y = x^2$ is chi-square distributed. If $m_x = 0$ it is a central chi-square random variable, otherwise it is a non-central chi-square random variable. The distribution and characteristic functions of y are [19],

$$f_y(y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y+m_x^2)/2\sigma^2} \cosh\left(\sqrt{y} m_x/\sigma^2\right) \tag{B.5}$$

$$\Phi_y(t) = \frac{1}{(1-j2t\sigma^2)^{1/2}} e^{jm_x t/(1-j2t\sigma^2)} \tag{B.6}$$

The chi-square distribution can be extended to the case $y = \sum_{i=1}^{N} x_i^2$, where the x_i are independent Gaussian variables with distributions $\mathrm{N}(m_i, \sigma^2)$. The mean's m_i need not be identical, but the variances must. Under this condition y is said to be a chi-square random variable with N degrees of freedom. Using the property (B.3) for sums of independent variables the characteristic function of a non-central chi-square random variable with N degrees of freedom is,

$$\Phi_y(t) = \frac{1}{(1-j2t\sigma^2)^{N/2}} \exp\left(\frac{jt\sum_{i=1}^{N} m_i^2}{(1-j2t\sigma^2)}\right). \tag{B.7}$$

For convenience of notation the non-centrality parameter s^2 is defined to be,

$$s^2 = \sum_{i=1}^{N} m_i^2. \tag{B.8}$$

Therefore chi-square random variables can be added by summing their degrees and centrality parameters.

Equation (B.4) can be used to obtain the following first few moments of the non-central chi-square distribution with N degrees of freedom,

$$\mathrm{E}(Y) = N\sigma^2 + s^2 \tag{B.9}$$

$$\mathrm{E}(Y^2) = 2N\sigma^4 + 4\sigma^2 s^2 + (N\sigma^2 + s^2)^2 \tag{B.10}$$

$$\sigma_Y^2 = 2N\sigma^4 + 4\sigma^2 s^2 \tag{B.11}$$

Appendix C

Properties Of The C Matrix

C.1 Eigenvalues Depend Only On W

The eigenvalues of the **C** matrix are all the solutions to the characteristic equation,

$$\det(\mathbf{C}-\lambda I). \tag{C.1}$$

Using cofactor expansion the determinant in (C.1) can be broken up into smaller determinants.

$$\begin{vmatrix} -\lambda & c_2 & & \\ c_2 & -\lambda & c_3 & \\ & c_3 & -\lambda & \ddots \\ & & \ddots & \ddots \end{vmatrix} = -\lambda \begin{vmatrix} -\lambda & c_3 & & \\ c_3 & -\lambda & c_4 & \\ & c_4 & -\lambda & \ddots \\ & & \ddots & \ddots \end{vmatrix} - c_2 \begin{vmatrix} c_2 & c_3 & & \\ & -\lambda & c_4 & \\ & c_4 & -\lambda & \ddots \\ & & \ddots & \ddots \end{vmatrix}$$

$$= -\lambda \begin{vmatrix} -\lambda & c_3 & & \\ c_3 & -\lambda & c_4 & \\ & c_4 & -\lambda & \ddots \\ & & \ddots & \ddots \end{vmatrix} - c_2^2 \begin{vmatrix} -\lambda & c_4 & & \\ c_4 & -\lambda & c_5 & \\ & c_5 & -\lambda & \ddots \\ & & \ddots & \ddots \end{vmatrix}$$

$$= -\lambda \begin{vmatrix} -\lambda & c_3 & & \\ c_3 & -\lambda & c_4 & \\ & c_4 & -\lambda & \ddots \\ & & \ddots & \ddots \end{vmatrix} - \begin{vmatrix} -\lambda & c_4 & & \\ c_4 & -\lambda & c_5 & \\ & c_5 & -\lambda & \ddots \\ & & \ddots & \ddots \end{vmatrix}$$

Following this technique it is clear that after enough expansions all of the c_m terms will be removed from the expression. Thus, the resulting characteristic polynomial

depends only on the length W of the preamble sequence, as that is what sets the dimensions of the **C** matrix.

C.2 Computing The Eigenvalues

As the characteristic polynomial depends only on W, for a given value of W it will be denoted as $f_W(\lambda)$. With this notation of the characteristic polynomial, it is possible to see a recursive relationship develop while performing the cofactor expansions.

$$\begin{aligned} f_W(\lambda) &= \begin{vmatrix} -\lambda & c_2 & & \\ c_2 & -\lambda & c_3 & \\ & c_3 & -\lambda & \ddots \\ & & \ddots & \ddots \end{vmatrix} \\ &= -\lambda \begin{vmatrix} -\lambda & c_3 & & \\ c_3 & -\lambda & c_4 & \\ & c_4 & -\lambda & \ddots \\ & & \ddots & \ddots \end{vmatrix} - \begin{vmatrix} -\lambda & c_4 & & \\ c_4 & -\lambda & c_5 & \\ & c_5 & -\lambda & \ddots \\ & & \ddots & \ddots \end{vmatrix} \\ &= -\lambda f_{W-1}(\lambda) - f_{W-2}(\lambda) \end{aligned} \tag{C.2}$$

Consider now, the Chebyshev Polynomials of the second kind. They have the following recursive relationship and roots [23],

$$\begin{aligned} U_n(x) &= 2xU_{n-1}(x) - U_{n-2}(x) \\ U_3(x)) &= 8x^3 - 4x \\ U_2(x)) &= 4x^2 - 1 \\ U_1(x)) &= 2x \\ x &= \cos\left(\frac{i\pi}{n+1}\right) \qquad i = 1, \ldots, n \end{aligned}$$

By noticing that $f_W(-2x) = U_W(x)$, it is possible to substitute $2x = -\lambda$ into the

roots of the Chebyshev Polynomials to obtain the roots of the characteristic polynomial (C.2).

$$\begin{aligned} \lambda &= -2\cos\left(\frac{i\pi}{n+1}\right) \\ &= 2\cos\left(\frac{i\pi}{n+1}\right) \end{aligned} \tag{C.3}$$

For ease of notation the minus sign is dropped in (C.3) since the roots are clearly symmetric.

C.3 Computing The Eigenvectors

The eigenvectors of the **C** matrix must satisfy $(\mathbf{C}-\lambda_i\mathbf{I})\vec{P}_i = 0$, this results in the following set of linear equations,

$$\begin{aligned} -P_{1i}\lambda_i + c_2P_{1i} &= 0 \\ c_2P_{1i} - P_{2i}\lambda_i + c_3P_{3i} &= 0 \\ c_3P_{2i} - P_{3i}\lambda_i + c_4P_{4i} &= 0 \\ &\vdots \\ c_{W-1}P_{W-2i} - P_{W-1i}\lambda_i + c_WP_{Wi} &= 0 \\ c_WP_{W-1i} - P_{Wi}\lambda_i &= 0. \end{aligned}$$

Using the initial condition of $P_{1i} = 1$ it is possible to solve for the remaining P_{ji}. While solving these equations the same recursive structure of the Chebyshev polynomials is found again.

$$
\begin{aligned}
P_{2i} &= c_2\lambda_i \\
P_{3i} &= c_3c_2\lambda_i^2 - c_3c_2 = c_3c_2(\lambda_i^2 - 1) \\
P_{4i} &= c_4c_3c_2(\lambda_i^3 - \lambda_i) - c_4c_3c_2\lambda_i = c_4c_3c_2(\lambda_i^3 - 2\lambda_i) \\
&\vdots \\
P_{ji} &= \left(\prod_{j=2}^{j} c_j\right) U_{j-1}(\lambda_i/2)
\end{aligned}
$$

For the special case of $x = \cos\theta$ the Chebyshev polynomials of the second kind can be expressed as [15],

$$U_n(x) = U_n(\cos\theta) = \frac{\sin[(n+1)\theta]}{\sin(\theta)}. \tag{C.4}$$

Since $\lambda_i = 2\cos(i\pi/W+1)$ (C.4) can be used to compute the remaining elements of the eigenvector.

$$
\begin{aligned}
P_{ji} &= \left(\prod_{k=2}^{j} c_k\right) \frac{\sin\left(\frac{ji\pi}{W+1}\right)}{\sin\left(\frac{i\pi}{W+1}\right)} \\
&\propto \sqrt{\frac{2}{W+1}} \left(\prod_{k=2}^{j} c_k\right) \sin\left(\frac{ji\pi}{W+1}\right)
\end{aligned} \tag{C.5}
$$

Equation (C.5) is the preferred expression for the eigenvectors because they must be unit vectors in order to satisfy $\mathbf{C} = \mathbf{P}\Lambda\mathbf{P}^\dagger, \mathbf{P}^\dagger\mathbf{P} = \mathbf{I}$.

Appendix D

Original Contributions

Articles Published/Accepted In Or Submitted To Refereed Journals

• S. Nagaraj, S. Khan, C. Schlegel, and M.V. Burnashev. *On Preamble Detection in Packet-Based Wireless Networks* submitted for publication in IEEE Transactions on Wireless Communication

Other Refereed Contributions

• S.Khan, S. Nagaraj, C. Schlegel, and M.V. Burnashev. *Performance of a Correlation-Based Detector for Packet Wireless Networks* 20th IEEE Canadian Conference on Electrical and Computer Engineering

• S. Nagaraj, S. Khan, C. Schlegel, and M.V. Burnashev. *On Preamble Detection in Packet-Based Wireless Networks* 9th IEEE International Symposium on Spread Spectrum Techniques and Applications, Proceedings of, pp. 476 - 480

Non-Refereed Contributions

• S. Khan, S. Nagaraj, C. Schlegel, and M.V. Burnashev. *Performance of a Correlation Based Preamble Detector for Asynchronous Communication* Canadian Summer School on Communications and Information Theory 2006

• S. Khan. *A Low Complexity Preamble Detector For Wireless Networks* CAIMS - MITACS 2006 Joint Annual Conference