PERFORMANCE OF COHERENT DIRECT SEQUENCE SPREAD SPECTRUM FREQUENCY SHIFT KEYING

A thesis presented to
the faculty of the
Russ College of Engineering and Technology of
Ohio University

In partial fulfillment
of the requirements for the degree
Master of Science

Sudhir Kumar Sunkara
November 2005
This thesis entitled
PERFORMANCE OF COHERENT DIRECT SEQUENCE SPREAD SPECTRUM
FREQUENCY SHIFT KEYING

by
SUDHIR KUMAR SUNKARA

has been approved for
the School of Electrical Engineering and Computer Science
and the Russ College of Engineering and Technology by

David W. Matolak
Associate Professor of Electrical Engineering and Computer Science

Dennis Irwin
Dean, Russ College of Engineering and Technology
We analyze the performance of a multi-user coherently-detected direct sequence spread spectrum frequency shift keying modulated system. Detection is done with respect to an arbitrary user. We derive bit error probability expressions for both synchronous and asynchronous systems. For the synchronous system, we analyze different variations of the system in order to ensure orthogonality. Orthogonality is achieved via both frequency separation and orthogonal codes. In the asynchronous system, orthogonality is achieved only through frequency separation. The channel impairments considered were additive white Gaussian noise and multi-user interference. We then extend this to a Rayleigh fading channel, and provide analytical and simulation results for single user and multi-user cases. We also analyzed the spectral characteristics of these systems.
Acknowledgements

I would like to thank my thesis advisor Dr. David W. Matolak, for his excellent guidance and support during the course of this thesis. This work would not have been possible without his insightful suggestions, and I feel privileged to have worked with him. I would also like to thank Dr. Jeffrey Dill, Dr Mehmet Celenk and Dr. Sergio Lopez for agreeing to be on my committee. None of this work would have been possible without the support of my GA supervisor, Issam Khoury. I am very much grateful to him for all the help he has extended to me during my stay at OU.

I thank my parents and brother for their love and support. Finally, I like to thank all my friends who inspired me during my studies at Ohio University.
Table of Contents

Abstract .................................................................................................................................. 3
Acknowledgements .................................................................................................................. 4
List of Figures .......................................................................................................................... 7

CHAPTER 1: INTRODUCTION .......................................................................................... 9

1.1 Digital Communications ............................................................................................... 9
1.2 Spread Spectrum Communications ............................................................................ 11
1.3 Digital Modulation Techniques .................................................................................. 12
1.4 Thesis Scope .............................................................................................................. 13

CHAPTER 2: SYSTEM MODEL .................................................................................. 15

2.1 Frequency Shift Keying .............................................................................................. 15
2.2 Literature Review ....................................................................................................... 17
2.3 Transmitter Structure ................................................................................................. 17
2.4 Receiver Structure ...................................................................................................... 19
2.4 System Variations ....................................................................................................... 21
2.4 Power Spectrum .......................................................................................................... 22

CHAPTER 3: ANALYSIS ............................................................................................ 30

3.1 Single User Bit Error Probability ............................................................................... 30
3.2 Decision Statistics ...................................................................................................... 34
3.3 System Variations ....................................................................................................... 36
3.4 Asynchronous System ................................................................................................. 42
CHAPTER 4: SIMULATIONS ............................................................. 52
  4.1 Simulation Model ................................................................. 52
  4.2 Simulation Results ............................................................... 54

CHAPTER 5: SUMMARY, CONCLUSIONS AND FUTURE WORK .......... 62
  5.1 Summary and Conclusions ..................................................... 62
  5.2 Suggestions for Future Work ............................................... 63

REFERENCES ............................................................... 64

APPENDIX ............................................................... 65
List of Figures

Figure 1.1. Block diagram of a typical digital communication system ............................10
Figure 2.1. BFSK modulated waveform for transmitted sequence 10010 (CPFSK) .........16
Figure 2.2. BFSK modulated waveform for transmitted sequence 10010 (non-CPFSK) .16
Figure 2.3. Transmitter block diagram ..............................................................................18
Figure 2.4. Receiver block diagram for first user .............................................................20
Figure 2.5. Power spectrum for a 3 user BFSK system ....................................................23
Figure 2.6. Power spectrum of single user DS-SS-BFSK ................................................24
Figure 2.7. Power spectrum for two user case. .................................................................26
Figure 2.8. Power spectrum of two user case: different PG such that BW is equal ..........28
Figure 2.9. $P_b$ vs. $E_b/N_0$, two users, different PG, same main-lobe bandwidth ..........29
Figure 3.1. BER vs. $E_b/N_0$ performance comparison (analytical) of coherent and non-
coherent BFSK...................................................................................................................31
Figure 3.2. Simulated and analytical performance (BER vs. $E_b/N_0$) of a single user coherente BFSK system on the AWGN channel .........................................................32
Figure 3.3. Single user BER performance for DS-SS-BFSK modulated system ..........33
Figure 3.4. Chip timing relationship between users $p$ and $k$, $L_k=4$, $\varepsilon_k \approx 0.5$ ..........44
Figure 3.5. Single user Tx and channel structure for a DS-SS-BFSK system over Rayleigh faded channel .................................................................49
Figure 4.1. Schematic description of MATLAB simulation model .................................53
**Figure 4.2.** $P_b$ vs. $E_b/N_0$, synchronous, different spacing, different codes, PG =15, 3 users (case 1)....................................................................................................................55

**Figure 4.3.** $P_b$ vs. $E_b/N_0$, synchronous, different spacing, same codes PG=15, 3 users (case 2)........................................................................................................................................................................56

**Figure 4.4.** $P_b$ vs. $E_b/N_0$, synchronous same spacing, different codes, PG=32, 2 users (case 3)........................................................................................................................................................................................................57

**Figure 4.5.** $P_b$ vs. $E_b/N_0$, synchronous, same spacing, same codes PG=31, 2 users (case 4)........................................................................................................................................................................................................58

**Figure 4.6.** $P_b$ vs. $E_b/N_0$, asynchronous, PG =31, 2 users .................................................59

**Figure 4.7.** $P_b$ vs. $E_b/N_0$, Rayleigh fading channel, PG=31.................................................60

**Figure 4.8.** $P_b$ vs. $E_b/N_0$, multi-user Rayleigh fading channel, PG =15, 2 users. ............61
Chapter 1

INTRODUCTION

In this chapter, we review basic concepts of digital communication systems used in this research. We present a brief overview of the spread spectrum varieties of digital communication systems. Then, we present a brief description of various digital modulation schemes. Finally, we present the scope of this thesis.

1.1 Digital Communications

Digital communications is a branch of communications that utilizes discontinuous signals, i.e., signals which appear in discrete steps, for example 0 and 1 for binary. It is different from analog communications, which uses continuous waveforms for transmitting data. The advantages of digital communications techniques include the following: greater data processing options and flexibilities; robustness to transmission impairments such as noise; and the ability to use error-detection and error-correction codes which further improve the performance. The disadvantages are that it requires somewhat complex equipment, may in some instances have limited transmission speed, and for some cases may require more bandwidth.

The principal feature of a digital communication system is that during a finite interval of time, the transmitter sends a waveform from a finite set of possible waveforms, in contrast to an analog communication system, which sends a waveform from an infinite variety of waveform shapes with theoretically infinite resolution [1]. In a digital communication system, the objective at the receiver is not to reproduce a
transmitted waveform with precision; instead, the objective is to determine from a noise-perturbed signal which waveform from the finite set of waveforms was sent by the transmitter. A block diagram of a digital communication system is shown in Figure 1.1 [2].

Figure 1.1. Block diagram of a typical digital communication system.

The upper row in Figure 1.1 depicts the various signal transformations from the information source to the transmitter output. Source encoding is performed primarily to remove redundant information, whereas the channel coding is done in order to decrease the probability of bit error in the presence of random noise. Modulation is necessary to convert the symbols to waveforms that are compatible with the transmission channel. We provide more details about modulation schemes in Section 1.3. The lower row of blocks denotes the signal transformations from the receiver input to the information sink. It can
be observed that the processes undergone from the receiver input to information sink are basically opposite to the transformations undergone by the signal from the source to the transmitter output. A detailed description and functionality of each block in Figure 1.1 can be found in [2].

### 1.2 Spread Spectrum Communications

Over the last 50 years, a class of digital modulation techniques called "Spread Spectrum" (SS) has been developed. The SS investigation was motivated primarily by the desire to achieve highly secure digital communication. These techniques are called spread spectrum because the transmission bandwidth employed is much greater than the minimum bandwidth required to transmit the information. Spreading is accomplished by means of a spreading signal, often called a code signal or pseudo-random (PR) or pseudo-noise (PN) signal. One important parameter of a SS system is the processing gain, defined as the ratio of transmission bandwidth to the information bandwidth. The advantages of SS systems include the following: suppression of interference such as that which comes from multipath propagation; resistance to jamming; reduction of energy density; and use in multiple access techniques such as code-division multiple access (CDMA).

An SS system can primarily be implemented in one of two varieties: direct sequence (DS) and frequency hopping (FH). In a DS-SS system, the information sequence is multiplied by a high-speed PN signal for spreading. The PN signal is independent of the data and from a spectral perspective, it has noise-like properties. At the receiver, a replica of the PN signal from the transmitter is used for de-spreading and
subsequent data recovery. The wideband signal required for a SS system is generated in a different manner in a FH system. The FH system takes the data signal and modulates it with a carrier signal whose center frequency hops from frequency to frequency over a wide band. The specific order in which frequencies are occupied is a function of the code signal, and the rate of hopping from one frequency to another is a function of the information rate. In this work, we focus on DS-SS communication systems. We do this for two reasons: (1) DS-SS technology is in popular use today in many applications, and (2) the use of FH-SS for FSK modulation has been thoroughly studied.

1.3 Digital Modulation Techniques

Digital modulation is the process by which digital symbols are transformed into waveforms that are compatible with the characteristics of the channel. In the case of baseband modulation, these waveforms take the form of shaped pulses. Bandpass modulation is the process by which an information signal is impressed upon a sinusoidal waveform; for digital modulation, such a sinusoid of duration \( T \) is referred to as a digital symbol. The sinusoid has just three features that can be used to distinguish it from other sinusoids: amplitude, frequency, and phase. Thus, bandpass modulation can be defined as the process whereby the amplitude, frequency, or phase of a radio frequency carrier, or a combination of them, is varied in accordance with the information to be transmitted. The general form of the carrier wave is given as follows:

\[
s(t) = A(t)\cos(2\pi f_c t + \theta(t)) \tag{1.1}
\]

where, \( A(t) \) is the time varying amplitude,

\( f_c \) is the carrier frequency, and \( \theta(t) \) is the time varying angle.
Some of the types of bandpass modulation techniques commonly employed are Phase Shift Keying (PSK), Frequency Shift Keying (FSK), Amplitude Shift Keying (ASK), Continuous Phase Modulation (CPM), etc.

When the receiver exploits knowledge of the carrier’s phase to detect the signals, the process is called coherent detection; when it does not utilize a phase reference, such a process is called noncoherent detection. Noncoherent systems are typically less complex than coherent ones, but coherent systems have better bit error rate performance.

1.4 Thesis Scope

Non-coherent detection of FSK, and of SS-FSK modulated systems has been investigated [3], [4]. Coherent FSK is has some applications in optical communication systems [6], [7], [9]. The main objective of this thesis is to develop analytical results for coherent detection of SS-FSK modulated systems, and corroborate these results with computer simulations.

The spectral characteristics of FSK waveforms are first described. We investigated the power spectra of both unspread and DS-spread FSK waveforms. We have also derived a set of relationships that allow different system users to have the same signal bandwidth, but different processing gains, a variation heretofore not explored.

To derive the probability of bit error expressions, we begin by considering a synchronous system. Synchronism refers to synchronism at the chip level. An example is a base to cellular transmitter. We considered different variations of this system, in terms of frequency separation and codes, to check for orthogonality. Probability of bit error expressions were developed for all orthogonal cases. We then considered an
asynchronous system, with all user signals being asynchronous with the user signal with respect to which we are detecting.

The channel impairments we considered were additive white Gaussian noise (AWGN), and multi-user interference. We also extended the analysis to a flat Rayleigh faded channel, and derived probability of bit error expression for a multi-user system. We provide simulation results to corroborate our analysis.
Chapter 2

SYSTEM MODEL

In this chapter, we discuss the basics of frequency shift keying modulation and then briefly discuss previous and related work. We then describe the system model employed in this thesis. After this, we present a description of the different cases we have considered to ensure orthogonality of the signals. Finally, we describe the power spectra of spread and unspread FSK waveforms.

2.1 Frequency Shift Keying

Frequency Shift Keying (FSK) is a digital modulation technique in which the symbols are represented by unique frequency signals that are transmitted through the channel. The general analytical expression for an $M$-FSK modulated signal is given by

$$ s_i(t) = \sqrt{\frac{2E}{T}} \cos(\omega_i t + \phi) \quad 0 \leq t \leq T, i = 1, \ldots, M $$

(2.1)

where $\omega_i = 2\pi f_i$, the term $f_i$ is the tone frequency, in Hz, corresponding to the transmitted symbol, and is a function of carrier frequency $f_c$ and the transmitted bits. The term $\phi$ is the phase associated with the frequency $f_i$; often this is modeled as random. For $M$-FSK, $f_i$ takes $M$ discrete values. The parameter $E$ is the symbol energy of the signal and $T$ is the symbol time. The symbol time is the time required to transmit one symbol. The carrier frequency is related to the bit/symbol time, and is typically much greater than the reciprocal of the bit/symbol time. Figure 2.1 shows a continuous phase binary FSK (CPFSK) modulated waveform for an example transmitted sequence 10010.
The bit time is $T_b=0.1\,\text{s}$, and the frequency separation is $1/T_b$. The choice of frequencies in Figure 2.1 yields a continuous phase waveform, which is not typically the case for FSK. For the same transmitted sequence, and different spacing between the frequency tones we obtain a non-continuous phase waveform as shown in Figure 2.2. The bit time is again 0.1s, and the frequency separation is $0.5/T_b$. 

Figure 2.1: BFSK modulated waveform for transmitted sequence 10010 (CPFSK).

Figure 2.2: BFSK modulated waveform for transmitted sequence 10010 (non-CPFSK).
2.2 Literature Review

As mentioned in Chapter 1, non-coherent detection of SS-FSK modulated signals has been studied recently, by Yang and Hanzo [3]. In this paper, they consider a spread spectrum multiple access system where the DS spread signals are only orthogonal over the symbol duration and not over the chip duration. Therefore, the frequency band of a spread FSK tone may be fully or partially overlapping with other spread signals. These authors develop an estimate of the variance of the multiple-access interference, with the aim of analyzing performance in this multi-user environment. They consider non-coherent demodulation at the receiver. Their paper concludes that for a given system bandwidth, the system’s BER (bit error rate) can be optimized by controlling the amount of spectral overlapping.

In our research, we are also assuming that the DS spread signals are orthogonal over the symbol duration only. Ryu et.al., in [4], extended this analysis by considering multi-tone jamming as an additional impairment. Geraniotis [5] analyzed the performance of non-coherent DS spread multiple access systems. Coherent demodulation was not considered because of its more complex implementation. Yet as noted in the first chapter, coherent FSK is now finding some application, in particular in optical communication systems [6], [7], thus we have analyzed the performance of a coherently detected SS-BFSK modulated system.

2.3 Transmitter Structure

We are considering a multi-user system with additive white Gaussian noise (AWGN), and multi-user interference (MUI) as the impairments. Later, we also address
channel fading. Each user generates its bit sequence, and the bit sequence is then BFSK modulated, i.e., at symbol (bit) transitions, there is a shift in the frequency of the modulated signal. The modulated waveforms are then spread using direct sequence spreading codes. We consider different cases when all the users have the same spreading codes and all of them have different spreading codes. The spreading codes employed are pseudo-random codes, for the most part—this is discussed subsequently. All the transmitted signals are then transmitted over the AWGN channel. Figure 2.3 shows the transmitter diagram. A possible application for this kind of transmitter structure is a cellular base station.

![Figure 2.3: Transmitter block diagram.](image)

The transmitted signal for the $k^{th}$ user is given as follows:

$$s_k(t) = \sqrt{2P^{(k)}} c_k(t) \cos\left\{2\pi f_c + b_k(t) \Delta_k t + \theta_k\right\}$$

(2.2)
where $P^{(k)}$ is the power of the $k^{th}$ user transmitted signal, and $c_k(t)$ is the signature (spreading) waveform consisting of a sequence of rectangular pulses $c^{(k)}_g$, where each “chip” pulse is of duration $T_c$. The spreading waveform is defined as

$$c_k(t) = \sum_g c_g P_{T_c}(t - gT_c), \quad (2.3)$$

where $c_g \in \{\pm 1\}$, and the waveform $P_T(t-nZ)$ is a rectangular pulse equal to unity from $t=nZ$ to $t=nZ+T$, and zero elsewhere. The $k^{th}$ user data waveform $b_k(t)$ consists of a sequence of mutually independent binary random variables with unit amplitude, +1 or -1, corresponding to rectangular pulses of duration $T_b$:

$$b_k(t) = \sum_n b_n P_{T_b}(t - nT_b), \quad (2.4)$$

with $b_n \in \{\pm 1\}$. The frequencies of the tones are at $f_c + b_n \Delta$, with $\Delta = i_k/(4T_b)$, where $T_b$ is the symbol (bit) time, and $i_k$ is a fixed integer, which specifies the spacing between the frequency tones for the $k^{th}$ user. The variable $\theta_k$ is the arbitrary phase introduced by the $k^{th}$ FSK modulator. $n(t)$ is the AWGN.

### 2.4 Receiver Structure

The receiver we employ is a conventional correlator receiver, and we detect with respect to an arbitrary user denoted the first user. The remaining users are considered interfering users. The block diagram for the receiver is shown in Figure 2.4.
The received signal, which is obtained after adding all the user transmitted signals and the AWGN, is multiplied with the spreading code of the first user, and then demodulated with respect to the frequency tones of the first user. Detection is coherent. We assume perfect knowledge of the phase throughout this work. The outputs of the multipliers are integrated over one bit time and the outputs of the correlators are then compared. The larger of the two is assigned to a binary 1. The output of this comparator is then compared to the known transmitted bits of the first user, to determine the bit error ratio (BER) for this system. We have considered different variations at the transmitter end to ensure orthogonality among signals, and we initially consider a synchronous system. This synchronism refers to synchronism at the chip level. Such a setting is a model for a point-to-multipoint link, where all user signals are generated at a common base station, e.g., a terrestrial cellular “forward” link. These cases are now described.
2.5 System Variations

Case 1: Users are separated in frequency and have different spreading codes

The user signals are separated in frequency by virtue of the spacing between the unspread signals ($\Delta$), and signals have different random spreading codes. Orthogonality is achieved via the frequency separation and not by the codes. The codes are used here for spectral spreading and signal “confidentiality” only. To ensure orthogonality for a coherent FSK system, the minimum separation between the frequency tones of a given user should be equal to $1/(2T_b)$, where $T_b$ is the bit time. In general, orthogonality can be maintained by employing any integer multiple of this minimal separation. The spreading codes, as mentioned, are randomly generated. The frequency spacing between the users’ signals is better illustrated in Fig. 2.5, which shows the unspread power spectra for a 3-user case. The “innermost” user’s frequency tones are separated by $1/T_b$. The second user’s frequency tones are separated by $2/T_b$, and the “outermost” user’s frequency tones are spaced by $3/T_b$. With this choice of frequency separation, all the user signals are orthogonal with respect to each other.

Case 2: Users are separated in frequency with the same spreading code

The user signals are separated in frequency in the same manner as in case 1. The only difference here is that all user waveforms are spread using the exact same spreading code.

Case 3: Users have the same frequency separation and different spreading codes.

In this case, orthogonality of the system is ensured by choosing orthogonal spreading codes for the users. Two spreading sequences are said to be orthogonal if their
correlation is zero. We employ the popular Walsh-Hadamard codes for this orthogonal code set, which requires that all signals be synchronized, at the chip level. All the users have the exact same frequency separation between their frequency tones, and the signals are orthogonal by virtue of the orthogonal spreading codes employed to differentiate between the users.

**Case 4:** Users have the same frequency separation and the same spreading code.

Even though the frequency tones of individual user signal are orthogonal, the user signals are not mutually orthogonal in this case. Since all users have the same spreading code, the signals are not orthogonal, and in fact, the signals are fully correlated. This case is mentioned only for completeness, as fully correlated signals can not be employed in a multi-user application.

We then extend the analysis to an asynchronous system. In an asynchronous system, code orthogonality is not employed, as the signals are asynchronous at the chip level, and no codes are known that maintain orthogonality over an arbitrary range of relative delays. Orthogonality of such a system is achieved via frequency separation of the tones.

### 2.6 Power Spectrum

The Power Spectral Density (PSD) of a signal represents the distribution of power in the frequency domain. FSK is a form of frequency modulation and the spectra of FM signals are very complex due to the fact that the modulating wave doesn’t just cause a change in the frequency of the carrier. The FM process creates many sidebands. FSK
spectra are thus much more complicated to predict than those of “linear” modulations such as PSK. We have plotted power spectra for both un-spread and spread BFSK cases, obtained via computer simulation using the periodogram method [8]. The power spectra for multi-user BFSK signals are as expected. Figure 2.5 shows a set of example power spectra for a 3 user BFSK modulated system (unspread).

![Power spectrum for a 3 user BFSK system.](image)

In this figure, the frequency tones of the “innermost” user are separated by $2/(2T_b)$, the second user tones are separated by $4/(2T_b)$, and the $3^{rd}$ user tones are separated by $6/(2T_b)$. For plotting, the bit time $T_b$ is normalized to unity. The carrier frequency is set to $9R_b/2$. This value was chosen so that the sidelobes are clear and evident in the plot. The sampling frequency $f_s = 4f_c$. The power spectra were generated using 100
realizations, each of 128 bits. The signal parameter values were maintained for all users, and the frequency tones of the users have different separations, as noted. The power spectrum of a spread BFSK signal has also been plotted. The input binary bits are first BFSK modulated, and this waveform is then multiplied with the spreading waveform with chip rate $R_c$. The carrier frequency is a function of $R_c$. The spacing between the frequency tones is still a function of $R_b$. Figure 2.6 shows the power spectrum of a single user’s spread BFSK waveform. For this case, the processing gain is large enough so that the spectrum appears to be essentially that of the random binary spreading waveform.

![Power Spectrum, DS-SS-BFSK, PG=32, $R_b=1$, $f_c=24R_c$.](image)

**Figure 2.6: Power spectrum of single user DS-SS-BFSK, $R_b=1$, $PG=32$, $f_c=24R_c$.**

This figure was plotted for a processing gain of 32. The carrier frequency was $f_c = 4R_c$, where $R_c = PG \times R_b$. The sampling frequency $f_s = 6f_c = 24R_c$. The spacing between the two frequency tones is $1/2T_b$. 
Figure 2.7 shows the DS-SS-FSK power spectra for two users. We have used different spreading codes on the two users. The dotted waveform represents the power spectra of user 1 and the solid line waveform represents user 2. The specifications for user 1’s signal are as follows: bit rate $R_b=1$, processing gain $PG_1=32$, chip rate $R_{c_1} = PG_1 \times R_b$, carrier frequency $f_{c_1} = 4R_{c_1}$. The frequency tones are at $f_{c_1} \pm 1R_b$. The first nulls on either side of $f_c$ occur at $f_{c_1} + \left( R_{c_1} + \frac{1}{2}R_b \right)$ and $f_{c_1} - \left( R_{c_1} + \frac{1}{2}R_b \right)$. The specifications for user 2’s signal are as follows: bit rate $R_b=1$, processing gain $PG_2=32$, chip rate $R_{c_2} = PG_2 \times R_b$, carrier frequency $f_{c_2} = 4R_{c_2}$. The frequency tones are at $f_{c_2} \pm 8R_b$. The first nulls on either side of $f_c$ occur at $f_{c_2} + \left( R_{c_2} + 4R_b \right)$ and $f_{c_2} - \left( R_{c_2} + 4R_b \right)$. Not apparent in this plot is the slightly different bandwidths of these two signals. This is due to the different values of frequency separations for the two tones of each user signal. For processing gains very large with respect to these frequency separations, this bandwidth difference will be minor, but for smaller processing gains, the signal bandwidths may differ substantially.

In order to have all signals occupy the same bandwidth (~ main lobe), we also investigated a setting wherein by changing the processing gain of the various user signals, we could have all the signal spectra occupy the same spectral band. Specifically, we analyzed the case where the spectral nulls of the main lobes of all the user spectra occurred at the same frequencies.
For a generic case with two users with the same bit rate $R_b$, the nulls on the positive side of carrier frequency occur at the following frequencies:

User 1:  
\[
  f_c + \left( R_{c1} + \frac{i_1}{2} R_b \right)
\]

User 2:  
\[
  f_c + \left( R_{c2} + \frac{i_2}{2} R_b \right)
\]  \hspace{1cm} (2.5)

where here, $i_1, i_2$ are integers that specify the spacing between the frequency tones of user 1 and user 2, respectively. Similarly, the nulls on the negative side of the carrier frequency occur at the following frequencies:

User 1:  
\[
  f_c - \left( R_{c1} + \frac{i_1}{2} R_b \right)
\]

User 2:  
\[
  f_c - \left( R_{c2} + \frac{i_2}{2} R_b \right)
\]  \hspace{1cm} (2.6)

For the first nulls to be equal, we have from Eq. (2.5) and Eq.(2.6)
\[
\left\{ f_c + \left( R_{c1} + \frac{i_2}{2} R_b \right) \right\} - \left\{ f_c + \left( R_{c2} + \frac{i_1}{2} R_b \right) \right\} = 0
\]

Using the knowledge that \( R_c = PGR_b \), we get

\[
P_{G2} R_b - P_{G1} R_b + R_b \left( \frac{i_2 - i_1}{2} \right) = 0
\]

Canceling out \( R_b \), we get

\[
P_{G2} = \left\{ P_{G1} + \frac{i_1 - i_2}{2} \right\}
\]

(2.7)

Similarly, for the first nulls on the negative side to coincide we have

\[
P_{G2} = \left\{ P_{G1} + \frac{i_1 - i_2}{2} \right\}
\]

(2.8)

Therefore the condition, from either (2.7) or (2.8), that will ensure that two users will have the same main-lobe bandwidth is

\[
P_{G2} = P_{G1} + \frac{i_1 - i_2}{2}
\]

(2.9)

We have to ensure that \( i_1 - i_2 \) is an integer multiple of 2 for the processing gains to be integers. Using this condition, we show in Figure 2.8 that the main-lobe bandwidths of both the users are indeed the same when this condition is met. In this figure, we used \( i_2 = 10 \), and \( i_1 = 18 \). The processing gain of user 1 is 20, and therefore from Eq. (2.9), we have the processing gain of the second user is \( 20 + (18-10)/2 = 24 \).
Figure 2.8: Power spectrum of two user case: different PG such that BW is equal.

Figure 2.9 plots the simulated probability of bit error performance for this case of unequal processing gains, on the AWGN channel. From the figure, we see that orthogonal signaling performance is obtained for this case. Since the processing gains are different, we employ different spreading codes. The codes are synchronous at the chip level. Figure shows the performance of user 1 in the presence of AWGN and the second user. The BER performance of the second user is similar to that of the first user in identical conditions.
Figure 2.9: $P_b$ vs. $E_b/N_0$, two users, different PG, same main-lobe bandwidth.
Chapter 3

ANALYSIS

In this chapter, we analyze the performance of the DS-SS-BFSK system for both the synchronous and asynchronous cases. We analyze the different variations of the system, discussed in Chapter 2.

3.1 Single User Bit Error Probability

When the receiver exploits knowledge of the carrier’s phase to detect the signals, the process is called coherent detection; when it does not utilize a phase reference, such a process is called non-coherent detection. For two FSK tones to be orthogonal, the minimum separation between them has to be $1/(2T_s)$ if the detection is coherent, and the minimum separation has to be $1/T_s$ if the detection is non-coherent, where $T_s$ is the symbol/bit time.

The general expression for the probability of bit error for binary coherent signals is given as [1]:

$$P_b = \frac{1}{\sqrt{2\pi}} \int_{(1-\rho)E_s/N_0}^{\infty} \exp\left(-\frac{\mu^2}{2}\right) d\mu$$

(3.1)

where $\rho = \cos \theta$ is the cross correlation coefficient between signals $s_1(t)$ and $s_2(t)$ that are defined by (2.1) and $\theta$ is the angle between signal vectors $s_1$ and $s_2$. For binary antipodal signals such as BPSK, $\theta = \pi$, so $\rho = -1$. For orthogonal signals such as BFSK, $\theta = \pi/2$, since the vectors $s_1$ and $s_2$ are perpendicular to each other and so $\rho = 0$. Thus, (3.1) can be written as
\[ P_b = \frac{1}{\sqrt{2\pi}} \int_{E_b/N_0}^\infty \exp\left(-\frac{u^2}{2}\right) du = Q\left(\sqrt{\frac{E_b}{N_0}}\right) \]  

(3.2)

where \(Q(x)\), the complementary error function, is defined as \(Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2} dt, x \geq 0\),

gives the area under the tail of the Gaussian probability density function (PDF).

The probability of bit error for binary non-coherent signals is given as follows [1]:

\[ P_b = \frac{1}{2} \exp\left(-\frac{E_b}{2N_0}\right) \]  

(3.3)

To illustrate these systems performance, Figure 3.1 shows the BER versus energy per bit to noise density ratio \(E_b/N_0\) of both coherent and non-coherent BFSK in AWGN.

\[ \text{Figure 3.1: BER vs. } E_b/N_0 \text{ performance comparison (analytical) of coherent and non-coherent BFSK} \]
The simulated performance of a single user coherently detected BFSK modulated system is compared with the analytical result in Figure 3.2. The simulation is carried out with “conventional” BFSK (for a waveform that is not spread), transmitted over an AWGN channel. The bit frequency, $R_b$, is normalized to one. The carrier frequency $f_c = 5R_b$. The separation between the frequency tones is $0.5R_b$ thus ensuring orthogonality. Analytical and simulation results are in very good agreement, validating the simulation accuracy.

![Figure 3.2: Simulated and analytical performance (BER vs. $E_b/N_0$) of a single user coherent BFSK system on the AWGN channel.](image)

Now, if we spread the BFSK modulated waveform using a spreading code with chip rate $R_c$, we still get orthogonality if the FSK tones are separated by $1/(2T_s)$. The
spread signals do not need to be separated by $1/(2T_c)$, where $T_c$ is the chip time, for the system to be orthogonal. If they are separated by $1/(2T_c)$, the FSK tone spectra do not overlap. Figure 3.3 shows the analytical and simulated BER results for a coherently detected DS-SS-BFSK system with a single user. The processing gain is $PG = 31$, and the bit rate $R_b$ is normalized to one. The chip rate, $R_c = PG R_b$, and the carrier frequency $f_c = 4R_c$. The frequency tones are spaced by $0.5f_c$. Once again, for this spread case, simulation and analytical results agree very well.

Figure 3.3: Single user BER performance for DS-SS-BFSK modulated system.
3.2 Decision Statistics

The transmitted signal for the $k^{th}$ user in a synchronous system as described in Figure 2.3 is given as

$$s_k(t) = \sqrt{2P^{(k)}} c_k(t) \cos \{2\pi [f_c + b_k(t)\Delta_k]t + \theta_k \}$$  \hspace{1cm} (3.4)

The parameters in (3.4) are described in (2.3) and (2.4). From Figure 2.4, we have the received signal as

$$r(t) = \sum_{k=1}^{K} s_k(t) + n(t)$$ \hspace{1cm} (3.5)

The decision statistic for the zeroth bit (without loss of generality) is then obtained as

$$Z_m = \int_{0}^{T_c} r(t)c_{i}(t)\cos[2\pi(f_c + m\Delta_1)t]dt$$ \hspace{1cm} (3.6)

where $m = +1$ or -1. Using Eq. (3.5) then yields

$$Z_m = \int_{0}^{T_c} \left\{ \sum_{k=1}^{K} s_k(t) + n(t) \right\}c_{i}(t)\cos[2\pi(f_c + m\Delta_1)t],$$

which can be expanded as

$$Z_m = \int_{0}^{T_c} \sum_{k=1}^{K} \sqrt{2P^{(k)}} c_k(t) \cos \{2\pi [f_c + b_k(t)\Delta_k]t + \theta_k \} + n(t) \right\}c_{i}(t)\cos[2\pi(f_c + m\Delta_1)t]dt$$  \hspace{1cm} (3.7)

After applying trigonometric identities to (3.7) and performing some algebra, the double frequency term can be ignored because $f_c \gg T_b^{-1}$. Then (3.7) can be broken up into the desired signal part, an interfering signal part, and the noise component. Specifically we have

$$Z_m = \sqrt{\frac{P^{(i)}}{2}} T_b \left[ D_m + \sum_{k=2}^{K} I_m^{(k)} + N_m \right]$$ \hspace{1cm} (3.8)
where

\[ D_m = \delta(b, m) \cos \theta_{k,m} \]  

(3.9)

is the desired signal component, \( \delta(b, m) = 1 \) for \( b = m \), otherwise \( \delta(b, m) = 0 \), and \( \theta_{k,m} \) is the phase angle of the reference (receiver local oscillator) signal when \( b=m \).

The noise component of the decision statistic is

\[
N_m = \left( \frac{P^{(1)}}{2} T_b \right)^{-1} \int_0^{T_b} n(t)c_1(t) \cos[2\pi(f_c + m\Delta_1)t]dt
\]

(3.10)

The Gaussian noise has a mean of 0, so \( E[N_m] = 0 \). We next determine the noise variance. This is computed as follows:

\[
Var[N_m] = \frac{2}{P^{(1)}T^2} E \left[ \int_0^{T_b} n(t) \cos(2\pi(f_c + m\Delta_1)t)dt \int_0^{T_b} n(y) \cos(2\pi(f_c + m\Delta_1)y)dy \right]
\]

Rearranging the terms on the right side, we have

\[
Var[N_m] = \frac{2}{P^{(1)}T^2} \int_0^{T_b} \int_0^{T_b} E[n(t)n(y)] \cos(2\pi(f_c + m\Delta_1)t) \cos(2\pi(f_c + m\Delta_1)y)dt\,dy
\]

\[
= \frac{2}{P^{(1)}T^2} \int_0^{T_b} R_n(t-y) \cos^2(2\pi(f_c + m\Delta_1)t)dt
\]

Since noise is white, its autocorrelation function \( R_n(t-y) = \sigma_n^2 \delta(t-y) \), then

\[
Var[N_m] = \frac{2}{P^{(1)}T^2} \sigma_n^2 \int_0^{T_b} \cos^2(2\pi(f_c + m\Delta_1)t)dt
\]

\[
= \frac{2}{P^{(1)}T^2} \sigma_n^2 \frac{T_b}{2}
\]

\[
= \frac{1}{P^{(1)}T_b} N_0
\]
Thus we have
\[
Var[N_m] = \left( \frac{2E_b}{N_0} \right)^{-1}
\] (3.11)
since the bit energy \( E_b = P^{(1)}T_b \).

For our synchronous case, the interference term is given by
\[
I_m^{(k)} = \sqrt{\frac{P_T^{(k)}}{P^{(1)}}} \frac{1}{T_b} \int_0^{T_b} c_i(t) c_k(t) \cos\left\{2\pi \left( b^{(k)} \Delta^{(k)} - m \Delta^{(1)} \right) + \theta \right\} dt
\] (3.12)
where \( P_T^{(k)} \) is the power of the interfering \( k^{th} \) user, and \( P^{(1)} \) is the power of user 1 (the user whose bits we are detecting). Let \( x^{(k)} = b^{(k)} \Delta^{(k)} - m \Delta^{(1)} \), where \( b^{(k)} \) is the bit transmitted by the \( k^{th} \) user, and \( m = +1 \) or \( -1 \). We next analyze the interference term for each of our cases of interest. This term will be zero mean due to the zero mean value of the random spreading codes.

### 3.3 System Variations

**Case 1:** When \( \Delta f \) is different for all users and different codes are used for each user, user separation is both by code and \( \Delta f \). The integral over \( T_b \) in (3.12) can be replaced by a sum of \( N \) integrals over \( T_c \) corresponding to the DS chip duration.

\[
I_m^{(k)} = \sqrt{\frac{P_T^{(k)}}{P^{(1)}}} \frac{1}{T_b} \sum_{i=0}^{N-1} c_i^{(k)} c_i^{(1)} \int_{iT_c}^{(i+1)T_c} \cos(2\pi \chi^{(k)} t + \theta) dt
\] (3.13)

\[
= \sqrt{\frac{P_T^{(k)}}{P^{(1)}}} \frac{1}{T_b} \sum_{i=0}^{N-1} c_i^{(k)} c_i^{(1)} \left\{ \sin\left(2\pi \chi^{(k)} T_c \right) \cos\left[2\pi \chi^{(k)} (2i+1) T_c + \theta \right] \right\}
\]
where $N$ is the processing gain of the system defined as $N = R_c/R_b$. Since we will be modeling the multiuser interference as a Gaussian random variable, we need to calculate the variance of the interference term, which is

$$E\left[I_m^{(k)}\right] = \frac{P_f^{(k)}}{P^{(l)}} \frac{T_c}{T_b} \sin c^2(2\pi x^{(k)}T_c) E\left[\sum_{l=0}^{N-1} c_l^{(k)} c_l^{(l)} \cos\left[2\pi x^{(k)}(2l+1)\right] + \sum_{p=0}^{N-1} c_p^{(d)} c_p^{(l)} \cos\left[2\pi x^{(k)}(2p+1)\right] T_c + \theta]\right]$$

$$= \frac{P_f^{(k)}}{P^{(l)}} \frac{T_c}{T_b} \sin c^2(2\pi x^{(k)}T_c) \left[\sum_{l=0}^{N-1} \sum_{p=0}^{N-1} E\left[\frac{c_l^{(k)} c_p^{(d)}}{E}\left[\cos\left[2\pi x^{(k)}(2l+2p+2)T_c + \theta\right]\right]\right] + \sum_{p=0}^{N-1} c_p^{(d)} c_p^{(l)} \cos\left[2\pi x^{(k)}(2l-2p)\right] T_c + \theta\right]\right]$$

where $\text{sinc}(x) = \sin(x)/x$. When taking the ensemble average, the expectation of the cosine term with $\theta$ is zero, because this phase term is a random variable uniformly distributed in $[0, 2\pi)$. For random codes, we have equal probabilities that $c_l^{(k)} = \pm 1$, and $c_p^{(d)} = \pm 1$, hence $E[c_l c_p]$ equals one when $l=p$ and zero otherwise. Thus we obtain

$$\text{Var}[I_m^{(k)}] = \frac{1}{2} \frac{P_f^{(k)}}{P^{(l)}} \left(\frac{T_c}{T_b}\right)^2 \sin c^2(2\pi x^{(k)}T_c) (N)$$

$$= \frac{P_f^{(k)}}{2P^{(l)}} \frac{1}{N} \sin c^2(2\pi x^{(k)}T_c),$$

$$\text{(3.14)}$$

where $N$ is the processing gain, and $x^{(k)} = b^{(k)} \Delta^{(k)} - m\Delta^{(l)}$.

The transmitted bit sequence consists of a series of $\pm 1$ terms each occurring with equal probability. Therefore $x^{(k)}$ takes values either in the set $\{-\Delta^{(k)} + \Delta^{(l)}, \Delta^{(k)} + \Delta^{(l)}\}$ or $\{-\Delta^{(k)} - \Delta^{(l)}, \Delta^{(k)} - \Delta^{(l)}\}$, when the term $m=-1$, and $+1$, respectively, with equal probability. So, the variance of the interfering term can be calculated by averaging (3.14) over all the possible four values of $x^{(k)}$. We therefore have
\( \text{Var}\left[I_m^{(k)}\right] = \frac{1}{4} \left\{ \sum_{l=1}^{P_t^{(k)}} \frac{P_l^{(k)}}{2P_t^{(l)}} \sin c^2(2\pi \Delta^{(k)} \Delta T_c) + \sum_{l=1}^{P_t^{(k)}} \frac{P_l^{(k)}}{2P_t^{(l)}} \sin c^2(2\pi \Delta^{(k)} \Delta T_c) \right\} \)

\[= \frac{1}{4} \frac{P_t^{(k)}}{2P_t^{(l)}} \left\{ \sin c^2(2\pi (\Delta^{(k)} - \Delta^{(l)}) T_c) + \sin c^2(2\pi (\Delta^{(k)} + \Delta^{(l)}) T_c) \right\} \]

\[= \frac{1}{4} \frac{P_t^{(k)}}{2P_t^{(l)}} \left\{ \sin c^2(2\pi (i^{(k)} - i^{(l)})/N) + \sin c^2(2\pi (i^{(k)} + i^{(l)})/N) \right\} \quad (3.15) \]

where \(i^{(k)}\) and \(i^{(l)}\) are integers that specify frequency spacing for the \(k^{th}\) and the 1\(^{st}\) users.

The mean value of the square of the desired signal term is \(\frac{1}{2}\), and thus using (3.15) and (3.11), the SNIR is given as

\[\text{SNIR} = \frac{1/2}{\left( \frac{2E_b}{N_0} \right)^{-1} + \frac{1}{4} \frac{P_t^{(k)}}{2P_t^{(l)}} \left\{ \sin c^2(2\pi (i^{(k)} - i^{(l)})/N) + \sin c^2(2\pi (i^{(k)} + i^{(l)})/N) \right\}} \quad (3.16) \]

Approximating the MUI as Gaussian, the probability of bit error is now given as:

\[P_b = \mathcal{Q}(\sqrt{\text{SNIR}}) \quad (3.17)\]

To corroborate our analysis we provide simulation results in Chapter 4 for each of the cases.

**Case 2:** Here, \(\Delta f\) is different for all users, with the same spreading code for all the users.

When we are considering the same spreading code for all the users, the MUI term in (3.12) can be written as

\[I_m^{(k)} = \sqrt{\frac{P_t^{(k)}}{P_t^{(l)}}} \frac{1}{T_b} \int_0^{T_b} c_i(t)c_k(t) \cos(2\pi \Delta^{(k)} t + \theta)dt \]

\[= \sqrt{\frac{P_t^{(k)}}{P_t^{(l)}}} \frac{1}{T_b} \sum_{l=0}^{N-1} c_i^{(k)} c_j^{(l)} \int_0^{T_b} \cos(2\pi \Delta^{(k)} t + \theta)dt \]
The product \( c_i^{(k)} c_i^{(l)} \) is equal to one since the spreading code is identical for all users.

Using this, and evaluating the integral, we have

\[
I_m^{(k)} = \sqrt{\frac{P_l^{(k)}}{P_1} \frac{1}{T_b} \frac{\sin(2\pi x^{(k)} T_c)}{2\pi x^{(k)}} \sum_{l=0}^{N-1} \cos[2\pi x^{(k)} (2l+1) T_c + \theta]} \tag{3.18}
\]

The closed form solution for the summation term in general is given by the following simpler form [2]

\[
\sum_{l=0}^{G-1} \cos(\alpha l + \beta) = \frac{\cos\{\beta + \lfloor (G - 1)/2 \rfloor \sin(G \alpha / 2)\}}{\sin(\alpha / 2)} \tag{3.19}
\]

From (3.18), we set \( \alpha = 4 \pi x^{(k)} T_c \), and \( \beta = 2 \pi x^{(k)} T_c + \theta \), therefore the summation term in (3.18) can now be written as

\[
\sum_{l=0}^{N-1} \cos[2\pi x^{(k)} (2l+1) T_c + \theta] = \frac{\cos\{2\pi x^{(k)} T_c + \theta + 0.5(N - 1)4 \pi x^{(k)} T_c\} \sin\{N2\pi x^{(k)} T_c\}}{\sin[2\pi x^{(k)} T_c]} \]

Then if we let \( A = 2 \pi x^{(k)} T_c + \theta + 0.5(N - 1)4 \pi x^{(k)} T_c \), we obtain

\[
I_m^{(k)} = \sqrt{\frac{P_l^{(k)}}{P_1} \frac{1}{T_b} \frac{\sin(2\pi x^{(k)} T_c)}{2\pi x^{(k)}} \frac{\cos(A) \sin\{N2\pi x^{(k)} T_c\}}{\sin\{2\pi x^{(k)} T_c\}}} \]

\[
= \sqrt{\frac{P_l^{(k)}}{P_1} \frac{1}{T_b} \frac{\sin c\{N2\pi x^{(k)} T_c\} \cos(A)}} \]

\[
= \sqrt{\frac{P_l^{(k)}}{P_1} \sin c\{N2\pi x^{(k)} T_c\} \cos(A)} \tag{3.20}
\]

Next, we need to calculate the variance of this interference term.

\[
E\left[I_m^{(k)}\right]^2 = E\left[\frac{P_l^{(k)}}{P_1} \sin c^2\{N2\pi x^{(k)} T_c\} \cos^2(A)\right]
\]

\[
= \frac{P_l^{(k)}}{P_1} \sin c^2\{N2\pi x^{(k)} T_c\} \frac{1}{2} E[1 + \cos(2A)]
\]
The term $A$ is a function of the uniformly distributed variable $\theta$, and therefore the mean of $\cos(2A)$ is zero. Thus,

$$Var[I^{(k)}_{\infty}] = \frac{P^{(k)}_l}{2P_i} \sin c^2 \left( N2\pi x^{(k)}\Delta T_c \right)$$  \hspace{1cm} (3.21)

As mentioned, the variance of the interference is found by averaging the right side of (3.21) over all four possible values of $x^{(k)}$. This yields

$$Var[I^{(k)}_{\infty}] = \frac{1}{4} \left\{ \sum_{i_{\infty} = 1}^{P^{(k)}_l} \frac{P^{(k)}_l}{2P^{(l)}} \sin c^2 \left( 2N\pi x^{(k)}\Delta T_c \right) + \sum_{i_{\infty} = 1}^{P^{(l)}} \frac{P^{(l)}_l}{2P^{(l)}} \sin c^2 \left( 2N\pi x^{(k)}\Delta T_c \right) \right\}$$

$$= \frac{P^{(k)}_l}{4P^{(l)}} \left\{ \sin c^2 \left( 2N\pi (\Delta^{(k)} + \Delta^{(l)}) \right) + \sin c^2 \left( 2N\pi (\Delta^{(k)} - \Delta^{(l)}) \right) \right\}$$

$$= \frac{P^{(k)}_l}{4P^{(l)}} \left\{ \sin c^2 \left( \pi (i^{(k)} + i^{(l)}) / 2 \right) + \sin c^2 \left( \pi (i^{(k)} - i^{(l)}) / 2 \right) \right\}$$ \hspace{1cm} (3.22)

Then for this case, our SNIR is given as

$$SNIR = \frac{1}{\left( \frac{2E_b}{N_0} \right)^{-1} + (K - 1) \frac{P^{(k)}_l}{4P^{(l)}} \left\{ \sin c^2 \left( \pi (i^{(k)} + i^{(l)}) / 2 \right) + \sin c^2 \left( \pi (i^{(k)} - i^{(l)}) / 2 \right) \right\}^{1/2}}$$ \hspace{1cm} (3.23)

and the bit error probability expression is

$$P_b = Q \left( \sqrt{\frac{2E_b}{N_0}}^{-1} + (K - 1) \frac{P^{(k)}_l}{4P^{(l)}} \left\{ \sin c^2 \left( \pi (i^{(k)} + i^{(l)}) / 2 \right) + \sin c^2 \left( \pi (i^{(k)} - i^{(l)}) / 2 \right) \right\}^{1/2} \right)$$ \hspace{1cm} (3.24)

**Case 3:** In this case we have $\Delta f$ the same for all users, with different spreading code on all user signals. The multiuser interference term is
\[ I_m^{(k)} = \sqrt{\frac{P_f^{(k)}}{P_f^{(l)}}} \frac{1}{T_b} \int_0^{T_b} c_i(t)c_k(t) \cos(2\pi\alpha^{(k)} t + \theta) dt \]

\[ = \sqrt{\frac{P_f^{(k)}}{P_f^{(l)}}} \frac{1}{T_b} \sum_{l=0}^{N-1} c_i^{(k)}c_l^{(l)} \int_0^{T_b} \cos(2\pi\alpha^{(k)} t + \theta) dt \]  

(3.25)

We employ orthogonal spreading codes to ensure orthogonality amongst user signals. Two spreading sequences are said to be orthogonal if their correlation is zero. We employ the popular Walsh-Hadamard codes for this orthogonal code set. Since the spreading codes are orthogonal, their product equals zero. Mathematically, we have

\[ \sum_{l=0}^{N-1} c_i^{(k)}c_l^{(l)} = 0 \]  

(3.26)

Therefore the interference term in (3.25) would also equal zero, leaving us with only noise. The probability of bit error is given by the expression for unspread coherent FSK, i.e., by

\[ P_b = Q \left( \sqrt{\frac{1/2}{\frac{2E_b}{N_0}}} \right) = Q \left( \sqrt{\frac{E_b}{N_0}} \right) \]  

(3.27)

**Case 4:** For this case, we have \( \Delta f \) the same for all users and the same spreading code on all users. As mentioned previously, this is the worst case scenario, and is generally not applicable for a multiuser system. The signals are not orthogonal, and therefore the BER performance will be very poor compared to the other systems.
3.4 Asynchronous System

All the analysis so far has been done assuming the system to be synchronous at the chip level. We now extend this analysis to the asynchronous case. In an asynchronous system, with arbitrary delays among user signals, we can’t achieve code orthogonality, since no codes are known that maintain orthogonality over an arbitrary range of relative delays. Orthogonality of such a system is achieved via frequency separation of the tones.

The received signal $r(t)$ is given by

$$r(t) = \sum_{k=1}^{K} s_k(t - \tau_k) + n(t) \quad (3.28)$$

where $n(t)$ is the channel noise and is again assumed to be a zero-mean stationary Gaussian process with double-sided spectral density of $N_0/2$. The term $\tau_k$ is the time delay relative to the reference signal. The $k^{th}$ transmitted signal is expressed as

$$s_k(t - \tau_k) = \sqrt{2P} c_k(t - \tau_k) \cos\{2\pi[f_c + b_k(t - \tau_k)\Delta] + \psi_k\}, 1 \leq k \leq K \quad (3.29)$$

where $\psi_k = \theta_k - 2\pi[f_c + b_k(t - \tau_k)\Delta]\tau_k$. The decision variable is given by

$$Z_m = \int_{0}^{T} r(t)c_1(t)\cos[2\pi(f_c + m\Delta)\Delta]\cos[2\pi(f_c + m\Delta)\Delta]dt \quad m = +1 \text{ or } -1;$$

From (3.28), we have

$$Z_m = \int_{0}^{T} \left[ \sum_{k=1}^{K} s_k(t - \tau_k) + n(t) \right]c_1(t)\cos[2\pi(f_c + m\Delta)\Delta]dt \quad (3.30)$$

Upon simplification, we get

$$Z_m = \sqrt{\frac{P}{2}} T \left[ D_m + \sum_{k=2}^{K} f_m^{(k)} + N_m \right] \quad (3.31)$$
where \( D_m = \delta(b_0, m) \cos \theta_m \) is the desired signal component, \( \delta(b_0, m) = 1 \) for \( b_0 = m \), otherwise \( \delta(b_0, m) = 0 \), and \( \theta_m \) is the phase angle of the reference signal when \( b_0 = m \).

Also, the noise term is

\[
N_m = \left( \frac{P}{2 T_b} \right)^{-1} \int_0^\tau n(t)c_1(t)\cos[2\pi(f_c + m\Delta, t)]dt .
\]

The interference term in (3.31) is given by the following expression

\[
I_m^{(k)} = \frac{1}{T_b} \left[ R_d[\tau_k, b_{k-1}^k, \psi_{-1}^k] + \hat{R}_d[\tau_k, b_1^k, \psi_1^k] \right]
\] (3.32)

with

\[
R_d[\tau_k, b_{k-1}^k, \psi_{-1}^k] = \int_0^{\tau_k} c_1(t)c_k(t-\tau_k)\cos\left\{2\pi\left[ b_{k-1}^{(k)} \Delta^{(k)} - m\Delta^{(l)} \right] + \psi_{-1}^k \right\}dt
\] (3.33)

\[
\hat{R}_d[\tau_k, b_1^k, \psi_1^k] = \int_{\tau_k}^{\tau_k} c_1(t)c_k(t-\tau_k)\cos\left\{2\pi\left[ b_1^{(k)} \Delta^{(k)} - m\Delta^{(l)} \right] + \psi_1^k \right\}dt
\] (3.34)

where \( R_d[\tau_k, b_{k-1}^k, \psi_{-1}^k] \) and \( \hat{R}_d[\tau_k, b_1^k, \psi_1^k] \) are the partial cross correlation terms corresponding to the transmitted bit sequences. We now proceed to find the variance of the interference term to calculate the bit error probability.

We first calculate the variance of the cross-correlation term \( R_d \) given by (3.33). We quantize the delay \( \tau_k \) into an integer number of chips plus a fractional-chip part as follows:

\[
\tau_k = (L_k + \varepsilon_k)T_c
\] (3.35)

where \( L_k \in \{1, 2, \ldots, N-1\} \) and \( 0 \leq \varepsilon_k \leq 1 \). Figure 3.4 illustrates the chip timing relationship for two users.
By breaking down (3.33) into a sum of integrals over each fractional chip period, we have

\[
R_d[\tau_k, x] = \sum_{l=0}^{L_k-1} c_1(l) \left[ c_k(-L_k + l - 1) \int_{T_c}^{(l+\varepsilon_k)T_c} \cos(2\pi\chi^{(k)}_l t + \psi) dt + c_k(-L_p + m) \int_{(l+\varepsilon_k)T_c}^{(l+1)T_c} \cos(2\pi\chi^{(k)}_l t + \psi) dt \right]
\]

where \( \chi^{(k)}_l = b^{(k)}_l \Delta^{(k)} \) \(- m \Delta^{(l)} \). Upon integration of the cosine terms, we have

\[
R_d[\tau_k, x] = \sum_{l=0}^{L_k-1} c_1(l) c_k(-L_k + l - 1) \frac{\sin(2\pi\chi^{(k)}_l \varepsilon_k T_c)}{2\pi\chi^{(k)}_l} \cos[2\pi\chi^{(k)}_l (2l + \varepsilon_k) T_c + \psi] + \sum_{l=0}^{L_p-1} c_1(l) c_k(-L_k + l) \frac{\sin[2\pi\chi^{(k)}_l (1 - \varepsilon_k) T_c]}{2\pi\chi^{(k)}_l} \cos[2\pi\chi^{(k)}_l (2l + 1 + \varepsilon_k) T_c + \psi] \tag{3.36}
\]

We now take the expectation of the square of (3.36). The terms obtained by multiplying two different-chip-indexed terms will be equal to zero since the chips are independent of each other. So we have to take the expectation of the sum of the squares of the two terms on the right side of (3.36).
\[
E[R_d(\tau_k, x)^2] = E \left\{ A \left( \sum_{l=0}^{L-1} c_l(l) c_k(-L_k + l - 1) \cos \left[ 2\pi \chi^{(k)}(2l + \varepsilon_k) \right] + B \left( \sum_{p=0}^{L-1} c_l(p) c_k(-L_k + p + 1) \cos \left[ 2\pi \chi^{(k)}(2p + 1 + \varepsilon_k) \right] \right) \right\}
\]

(3.37)

where \( A = (\varepsilon_k T_c)^2 \sin c^2 \left( 2\pi \chi^{(k)} \varepsilon_k T_c \right) \), and \( B = \left( (1 - \varepsilon_k) T_c \right)^2 \sin c^2 \left( 2\pi \chi^{(k)} (1 - \varepsilon_k) T_c \right) \).

Rearranging the terms in (3.37), we have

\[
E[R_d(\tau_k, x)^2] = \begin{bmatrix}
A \left( \sum_{l=0}^{L-1} c_l(l) c_k(-L_k + l - 1) \cos \left( (2l + \varepsilon_k) T_c \right) \right) \\
+ B \left( \sum_{p=0}^{L-1} c_l(p) c_k(-L_k + p + 1) \cos \left( (2p + 1 + \varepsilon_k) T_c \right) \right)
\end{bmatrix}
\]

(3.38)

When taking the ensemble average in (3.38), the expectation of the cosine term containing \( \psi \) is zero, because it is a random variable uniformly distributed from \([0, 2\pi)\).

For equal probability that \( c_l(k) = \pm 1, \) and \( c_p(k) = \pm 1, \) \( E[c_l c_p] \) equals one when \( l=p \) and zero otherwise. Thus we obtain the much simpler expression

\[
E[R_d(\tau_k, x)^2] = L_k \left\{ \left( \varepsilon_k T_c \right)^2 \sin c^2 \left( 2\pi \chi^{(k)} \varepsilon_k T_c \right) + \left( (1 - \varepsilon_k) T_c \right)^2 \sin c^2 \left( 2\pi \chi^{(k)} (1 - \varepsilon_k) T_c \right) \right\}
\]

(3.39)
In (3.39), \( x_{i-1}^{(k)} \) takes the values in \( \{-\Delta^{(k)} + \Delta^{(i)}, -\Delta^{(k)} - \Delta^{(i)}\} \) when \( m=-1 \), and +1 respectively, with equal probability. So, the variance of the interfering term can be calculated by averaging (3.39) over the two possible values of \( x_{i-1}^{(k)} \). We therefore have

\[
E[R_d(\tau_c)|^2] = \frac{L_k}{2} \left\{ \left( \varepsilon_k T_c \right)^2 \left\{ \sin c^2 \left( 2\pi \left( \Delta^{(k)} - \Delta^{(i)} \right) \varepsilon_k T_c \right) + \sin c^2 \left( 2\pi \left( \Delta^{(k)} + \Delta^{(i)} \right) \varepsilon_k T_c \right) \right\} + \left( (1-\varepsilon_k) T_c \right)^2 \left\{ \sin c^2 \left( 2\pi \left( \Delta^{(k)} - \Delta^{(i)} \right) (1-\varepsilon_k) T_c \right) + \sin c^2 \left( 2\pi \left( \Delta^{(k)} + \Delta^{(i)} \right) (1-\varepsilon_k) T_c \right) \right\} \right\} \tag{3.40}
\]

We next evaluate the expectation of the square of the second partial cross correlation term given by (3.34). By breaking down (3.34) into a sum of integrals over each fractional chip period, we have

\[
\hat{R}_d[\tau_k, x] = \sum_{l=-L_k+1}^{N_k-1} c_l(l) \left[ c_k(-L_k + l - 1) \int_{\tau_{l+1}}^{\tau_l} \cos(2\pi \tau_{x(l)} t + \psi) dt \right. \left. + c_k(-L_k + m) \int_{\tau_{l+1}}^{\tau_l} \cos(2\pi \tau_{x(l)} t + \psi) dt \right] \tag{3.41}
\]

Simplifying the above equation similar to the procedure described for the first partial correlation function, we obtain

\[
E[\hat{R}_d(\tau_c, x)^2] = (N_B - L_k)^2 \left\{ \left( \varepsilon_k T_c \right)^2 \sin c^2 \left( 2\pi \tau_{x(l)} \varepsilon_k T_c \right) + \left( (1-\varepsilon_k) T_c \right)^2 \sin c^2 \left( 2\pi \tau_{x(l)} (1-\varepsilon_k) T_c \right) \right\} \tag{3.42}
\]

In (3.42), \( x_i^{(k)} \) takes the values in \( \{\Delta^{(k)} + \Delta^{(i)}, \Delta^{(k)} - \Delta^{(i)}\} \), when \( m=-1 \), and +1 respectively, with equal probability. So, the variance of the interfering term can be calculated by averaging (3.42) over the two possible values of \( x_i^{(k)} \). We therefore have
\[
E\left[ \hat{R}_d(t) \right]^2 = \left( \frac{N \cdot L_k}{2} \right) \left\{ \left( \epsilon_k T_c \right)^2 \left( \sin c^2 \left( 2\pi \left( \Delta^{(k)} - \Delta^{(1)} \right) \epsilon_k T_c \right) + \sin c^2 \left( 2\pi \left( \Delta^{(k)} + \Delta^{(1)} \right) \epsilon_k T_c \right) \right) \right\} + \left( (1 - \epsilon_k) T_c \right)^2 \left( \sin c^2 \left( 2\pi \left( \Delta^{(k)} - \Delta^{(1)} \right) (1 - \epsilon_k) T_c \right) + \sin c^2 \left( 2\pi \left( \Delta^{(k)} + \Delta^{(1)} \right) (1 - \epsilon_k) T_c \right) \right) \right\}
\]  
(3.43)

From (3.32), we now have,

\[
E[I_m^2] = \frac{P_i}{2 P_b} \frac{N}{T_b^2} \left\{ \left( \epsilon_k T_c \right)^2 \left( \sin c^2 \left( 2\pi \left( \Delta^{(k)} - \Delta^{(1)} \right) \epsilon_k T_c \right) + \sin c^2 \left( 2\pi \left( \Delta^{(k)} + \Delta^{(1)} \right) \epsilon_k T_c \right) \right) \right\} + \left( (1 - \epsilon_k) T_c \right)^2 \left( \sin c^2 \left( 2\pi \left( \Delta^{(k)} - \Delta^{(1)} \right) (1 - \epsilon_k) T_c \right) + \sin c^2 \left( 2\pi \left( \Delta^{(k)} + \Delta^{(1)} \right) (1 - \epsilon_k) T_c \right) \right) \right\}
\]  
(3.44)

Simplifying (3.44) using the knowledge that \( N = T_c / T_b \) and \( \Delta^{(k)} = i^{(k)} / T_b \), we get

\[
E[I_m^2] = \frac{P_i}{4 P_b} \frac{1}{N} \left\{ \left( \epsilon_k \right)^2 \left( \sin c^2 \left( 2\pi \left( i^{(k)} - i^{(1)} \right) \epsilon_k / N \right) + \sin c^2 \left( 2\pi \left( i^{(k)} + i^{(1)} \right) \epsilon_k / N \right) \right) \right\} + \left( (1 - \epsilon_k) \right)^2 \left( \sin c^2 \left( 2\pi \left( i^{(k)} - i^{(1)} \right) (1 - \epsilon_k) / N \right) + \sin c^2 \left( 2\pi \left( i^{(k)} + i^{(1)} \right) (1 - \epsilon_k) / N \right) \right) \right\}
\]  
(3.45)

When the delay \( \tau_k \) is zero, we see that the above result is identical to that of the synchronous case. We now average the interference term over the random variables \( \epsilon_k \) and \( L_k \). We model \( L_k \) as a uniformly distributed integer over \([0, N-1]\), and \( \epsilon \) as a real uniform variate over \([0,1)\).

Let \( y_1 = \frac{i^{(k)} - i^{(1)}}{N}, y_2 = \frac{i^{(k)} + i^{(1)}}{N} \). Averaging (3.45) over \( \epsilon_k \), we now have

\[
E \left[ (I_m^2) \right] = \frac{P_i}{4 P_b} \frac{1}{N} \left\{ \frac{\sin^2 \left( 2\pi y_1 \epsilon_k \right)}{(2\pi y_1)^2} + \frac{\sin^2 \left( 2\pi y_2 \epsilon_k \right)}{(2\pi y_2)^2} + \frac{\sin^2 \left( 2\pi y_1 (1 - \epsilon_k) \right)}{(2\pi y_1)^2} + \frac{\sin^2 \left( 2\pi y_2 (1 - \epsilon_k) \right)}{(2\pi y_2)^2} \right\} dx
\]  
(3.46)

Solving the integral in (3.46) we get
\[
E\left[\left(\frac{I_m^{(k)}}{N}\right)^2\right] = \frac{P_l^{(k)}}{4P_t} \frac{1}{N} \left[ \frac{1}{(2\pi y_1)^2} \left(1 - \sin c(4\pi y_1)\right) + \frac{1}{(2\pi y_2)^2} \left(1 - \sin c(4\pi y_2)\right) \right] \tag{3.47}
\]

Simplifying (3.47) by substituting for \(y_1\) and \(y_2\), we finally obtain

\[
E\left[\left(\frac{I_m^{(k)}}{N}\right)^2\right] = \frac{P_l^{(k)}}{4P_t} \frac{1}{N} \left[ \frac{1}{\left(2\pi\left(i^{(k)} - i^{(1)}\right)\right)} \left(1 - \sin c\left(4\pi\left(i^{(k)} - i^{(1)}\right)/N\right)\right) \right. \\
+ \left. \frac{1}{\left(2\pi\left(i^{(k)} + i^{(1)}\right)\right)} \left(1 - \sin c\left(4\pi\left(i^{(k)} + i^{(1)}\right)/N\right)\right) \right]\tag{3.48}
\]

This expression is used in the SNIR expression analogous to (3.16), and then into the BER expression of (3.17) to estimate performance for this asynchronous case.

### 3.5 Rayleigh Fading Channel

Mobile communication channels can be characterized by two types of fading effects: large-scale fading and small-scale fading. Large-scale fading represents the average signal power attenuation or path loss due to motion over large areas [2]. The statistics of large-scale fading provide a way of computing an estimate of path loss as a function of distance. Small-scale fading is a “local” phenomenon, in that it is nearly independent of distance between the transmitter and receiver. Small-scale fading refers to the dramatic changes in signal amplitude and phase that can be experienced as a result of small changes in the spatial positioning between a receiver and transmitter. Small-scale fading is often modeled as *Rayleigh fading* if there are multiple reflective paths that are large in number and of approximately equal amplitude in the absence of a line-of-sight signal component. The fading we consider is “flat,” or frequency non-selective.
This type of fading arises when all the multipath delays are much smaller than the shortest baseband signal duration—here, the chip time $T_c$.

In the system we are considering, fading is encountered after all the transmitted signals are summed, when this aggregate signal is sent through the channel. The AWGN is then added to this signal at the receiver. We now proceed to describe the bit error rate computations for single user and multi-user Rayleigh faded channels. Figure 3.5 shows a block diagram of the transmitter and channel structure for this Rayleigh fading case illustrating a single user DS-SS-BFSK transmission.

![Figure 3.5: Single-user Tx and channel structure for a DS-SS-BFSK system over Rayleigh faded channel](image)

In a synchronous multi-user environment, all the spread waveforms are spread and then transmitted over the channel. The resultant signal is Rayleigh faded and AWGN is then added. The received signal is now given as

$$r(t) = \alpha \sum_{k=1}^{K} s_k(t) + n(t)$$  \hspace{1cm} (3.49)

where $\alpha$ is the fading amplitude of the channel and $n(t)$ is the AWGN with mean zero and variance $N_0/2$. We have without loss of generality set the channel propagation delay to zero.
Using the binary minimum error probability detector formulation developed in [2], the probability of bit error for a BFSK modulated system can be written as

\[ P_2(\gamma_b) = \mathcal{Q}(\sqrt{\gamma_b}) \quad (3.50) \]

where the effective signal to noise ratio at the receiver is \( \gamma_b = \alpha^2 E_b / N_0 \).

In order to obtain an expression for the bit error ratio (BER) on a Rayleigh fading channel, \( P_2(\gamma_b) \) must be averaged over the PDF of \( \gamma_b \). When \( \alpha \) is Rayleigh distributed, \( \alpha^2 \) has a chi-square probability distribution with two degrees of freedom. Consequently, \( \gamma_b \) is also chi-square distributed with two degrees of freedom, and its PDF is given by [2]

\[ p(\gamma_b) = \frac{1}{\bar{\gamma}_b} e^{-\gamma_b / \bar{\gamma}_b}, \gamma_b \geq 0 \quad (3.51) \]

where \( \bar{\gamma}_b \) is the average signal-to-noise ratio, defined as \( \bar{\gamma}_b = E(\alpha^2) E_b / N_0 \).

The expression for BER is then obtained by averaging \( P_2(\gamma_b) \) over \( p(\gamma_b) \)

\[ P_b = \int_0^\infty P_2(\gamma_b) p(\gamma_b) d\gamma_b \quad (3.52) \]

\[ = \int_0^\infty \mathcal{Q}(\sqrt{\gamma_b}) \frac{1}{\bar{\gamma}_b} e^{-\gamma_b / \bar{\gamma}_b} d\gamma_b \]

Using integration by parts, and solving (3.51), we get

\[ P_b = \frac{1}{2} \left[ 1 - \sqrt{\frac{\bar{\gamma}_b}{2 + \bar{\gamma}_b}} \right] \quad (3.53) \]

We assume that the mean square value of the Rayleigh variable is one for simulations.

In the multi-user setting, we model the multi-user interference as Gaussian. The phase of each user at the transmitter is different and random, \( \alpha \) is Rayleigh distributed,
so we can model the multi-user interference (conditioned on the fading variable) as Gaussian. The probability of bit error is then given as

$$P_b = \frac{1}{2} \left[ 1 - \sqrt{\frac{\bar{\gamma}_{bu}}{2 + \bar{\gamma}_{bu}}} \right]$$

(3.54)

where $\bar{\gamma}_{bu}$ is the average signal-to-noise plus interference ratio, defined as

$$\bar{\gamma}_{bu} = E(\alpha^2) \frac{E_b}{N_0 + I_0}.$$ 

(3.55)

where $I_0$ is the multi-user interference term derived for different cases in the previous sections.
Chapter 4

SIMULATIONS

In this chapter, we present the simulation model used in the thesis and also the results obtained for different variations of the system.

4.1 Simulation Model

A DS-SS-BFSK simulation model was developed for this research. One of the methods used for the performance evaluation of digital systems is estimation of bit error probability. We employ the *Monte Carlo* method for this estimation. The *Monte Carlo* method is a numerical method for statistical simulation which utilizes sequences of random numbers to perform the simulation. The simulation computes an estimate for the bit error probability. The simulation model was developed in MATLAB®.

At the transmitter end, random binary data is generated using the `rand` function in MATLAB. The `rand` function generates random numbers chosen from a uniform distribution on the interval (0,1). The generated bits are then BFSK modulated. The frequency tones are at $f_c \pm R_b i$, where $f_c$ is the carrier frequency. Spreading codes are then generated similar to the way data bits were generated. The length of the spreading code is $PG$, the processing gain. The chip rate is $R_c = PG \times R_b$. The BFSK modulated waveform is then multiplied with the spreading code to obtain the spread BFSK waveform. The bit rate $R_b$ is normalized to one for convenience.
The impairments present in the channel are AWGN, MUI, and Rayleigh fading. The AWGN is generated by using the \textit{randn} function in MATLAB. The function \textit{randn} generates random numbers chosen from a normal (Gaussian) distribution with mean zero and variance one. By keeping energy per bit $E_b$ constant and changing the variance of AWGN, desired values of $E_b/N_0$ are obtained.
The Rayleigh fading is generated by using the fact that the square root of the summation of two squared Gaussian random variables is a Rayleigh random variable, i.e., if \( X \) and \( Y \) are two Gaussian random variables, then \( z = \sqrt{X^2 + Y^2} \), is a Rayleigh random variable, and it is characterized by the single parameter \( E(z^2) \), the mean square value of \( z \) where \( E(.) \) is the expectation operator. As noted, we have used a mean-square value of 1 for the Rayleigh distribution in the simulations. The unit mean-square value of Rayleigh fading energy keeps the average \( E_b/N_0 \) ratio constant for a constant value of transmitted bit energy.

The signal that enters the receiver is despread and downconverted by the composite code signal. The signal is then integrated over the symbol time, and hard bit decisions are made on the decision variables. The bit error rate estimation is performed by comparing these received bit estimates with transmitted bits. The input parameters to these simulations are as follows

- \( R_b \): bit rate,
- \( f_c \): Carrier frequency
- \( f_s \): Sampling frequency
- \( PG \): Processing gain,
- \( E_b/N_0 \) range
- Spacing between the frequency tones \( \Delta f \)

### 4.2 Simulation Results

In this section, we present the results of the simulations we obtained for different variations of the system we studied. Figure 4.2 plots the simulated bit error rate vs.
analytical bit error rate for a 3-user synchronous DS-SS-BFSK system in an AWGN channel. This plot is for the case where we have different spacings between the frequency tones, and also different spreading codes for each user. Orthogonality amongst the users is achieved via frequency separation. The first user’s frequency tones are at \( f_c \pm 0.5f_b \), the second user’s tones are at \( f_c \pm f_b \), and the third user’s are at \( f_c \pm 3f_b \). The common input parameters considered were: \( R_b = 1 \), processing gain, \( PG = 15 \), carrier frequency \( f_c = 4R_c \), and sampling frequency \( f_s = 4f_c \).

![Figure 4.2: \( P_b \) vs. \( E_b/N_0 \), synchronous, different spacing, different codes, PG =15, 3 users (case 1).](image)

It is evident from the figure that simulations yield orthogonal performance. We get similar BER performance for the other two users also.

For the same specifications used in the plots of Figure 4.2, we plot the simulated performance of the synchronous system when all the user signals are spread using the
same spreading code (Case 2). Here again, orthogonality is achieved via frequency separation. Figure 4.3 shows the simulated performance is in very close agreement with the analysis and orthogonal performance is achieved. Again, we get similar performance for the other two users.

![Figure 4.3: $P_b$ vs. $E_b/N_0$, synchronous, different spacing, same codes PG=15, 3 users (case 2).](image)

Figure 4.4 plots the simulated performance of the synchronous system when orthogonality is achieved through spreading codes. For this simulation, we employ Walsh-Hadamard codes of length 32, i.e., the processing gain used is 32. We ran this simulation for a two user system. Both the users have their frequency tones at $f_c \pm 0.5 f_b$. The other parameters remain the same as in the previous two figures. As discussed in Chapter 3, we again have orthogonal signaling performance, and simulation
and analytical results are in good agreement. The BER performance of the second user is identical to that of the first user.

Figure 4.4: $P_b$ vs. $E_b/N_0$, synchronous same spacing, different codes, PG=32, 2 users (case 3).

Figure 4.5 shows the plot for the case where all the users have the exact same frequency separation and same spreading codes, for a two-user example. Even though the frequency tones of an individual user signal are orthogonal, the user signals are not mutually orthogonal. As previously noted, the signals are fully correlated. Figure 4.4 confirms our analysis that this system will perform poorly in a multi-user application. Both the users have their frequency tones at $f_c \pm 0.25 f_b$. 
Figure 4.5: $P_b$ vs. $E_b/N_o$, synchronous, same spacing, same codes $PG=31$, 2 users (case 4).

For an asynchronous system, Figure 4.6 plots the simulated and analytical bit error probabilities. We employ a two user system with the relative delay of the second user equal to $0.7T_b$. The first user’s frequency tones are at $f_c \pm 0.5f_b$, and the second user’s tones are at $f_c \pm f_b$. The common input parameters were $R_b=1$, processing gain, $PG=31$, carrier frequency $f_c=4R_c$, and sampling frequency $f_s=4f_c$. As can be seen for this case, orthogonality is still maintained by virtue of the frequency separation, even when the signals are asynchronous.
Figure 4.6: $P_b$ vs. $E_b/N_0$, asynchronous, PG =31, 2 users

We also ran simulations for a Rayleigh fading channel. Figure 4.7 shows the simulated and analytical performance of single user DS-SS-BFSK signal on the Rayleigh channel. There is excellent agreement between the simulated and analytical values for this channel as well.
Finally, as explained in section 3.5, we can model the multi-user interference as Gaussian, and obtain multiuser performance results on the fading channel. Figure 4.8 shows that there is very good agreement between the simulations and analysis for this case.

Figure 4.7: $P_b$ vs. $E_b/N_0$, Rayleigh fading channel, PG=31
Figure 4.8: $P_b$ vs. $E_b/N_0$, multi-user Rayleigh fading channel, PG =15, 2 users.
Chapter 5

SUMMARY, CONCLUSIONS AND FUTURE WORK

In this chapter, we summarize the work done in this thesis, and present conclusions drawn from this research. We conclude the chapter by listing some suggestions for future work.

5.1 Summary and Conclusions

Non-coherent detection for DS-SS-FSK has already been investigated [3], thus in this work we studied the performance of several coherently detected DS-SS-FSK systems. In the first part of the thesis, we derived probability of bit error expressions for a synchronous coherently detected DS-SS-BFSK system. Synchronism refers to synchronism at the chip level. Coherent detection is more complex to implement than non-coherent, but it is now finding applications in optical communication systems [6], [7]. We have analyzed both single user and multi-user systems. We have analyzed and simulated these systems with several variations to ascertain performance. These variations were made in the frequency separations of the users and in the type of spreading codes used. We also investigated the spectral characteristics of both spread and non-spread BFSK waveforms. Spectra with different frequency spacings were plotted, and we derived conditions on frequency separations $\Delta f$ and processing gains for which mainlobe bandwidths are equal in a multiuser DS-SS-FSK system.

In the next part of the thesis, we extended our analysis to study an asynchronous system. In an asynchronous system, we will not have as many variations as we have in a
synchronous system because orthogonality of the system is obtained via only the
frequency separation and not by orthogonal codes. Given appropriate frequency
separations, our analyses and simulations yielded orthogonal performance.

In all this work, the impairments considered were AWGN and MUI. We next
considered Rayleigh fading as an additional channel impairment. We derived probability
of bit error expressions for single-user and multi-user environments. Simulations were
again in very good agreement with analysis. We concluded that orthogonality of the
signals in a given system can be achieved either via frequency separation or via
orthogonal spreading codes. And since coherent FSK is now finding applications in
optical communications, we hope these expressions are useful for further research.

5.2 Suggestions for Future Work

We have not considered the effect of varying the spacing between the user’s
frequency tones on the performance of the system. For example, it is well known that
coherent BFSK can obtain better than orthogonal signaling performance with a frequency
separation greater than that for orthogonal signaling [2]. Hence, we could investigate this
modification for improved performance. An additional idea for possible future work is
that the system performance can be analyzed for other impairments not considered here.
This includes channel dispersion, and in-band interference, or jamming. Another area of
possible research is to extend the analysis for an $M$-ary system.
REFERENCES:


[8]. MATLAB, Mathworks

clear all;
close all;
N=5; %Number of bits
%xb=[1 0 0 1 0];
xb=BIN01(N,0.5); %Generate random binary data
fb=10; %Bit frequency
fc=fb; %Carrier frequency
fs=100*fc; %Sampling frequency
tt=[];
for ii=1:N
    t(ii,:)=(ii-1)/fb:1/fs:ii/fb-1/fs; % Time vector for ii-th bit
    st(ii,:)=cos(2*pi*(fc+xb(ii)*fb*1).*t(ii,:)); % FSK Transmitted Waveform, ii-th bit
    tt=[tt t(ii,:)];
end
stp=st'; % Transpose signal matrix for subsequent "vectorizing"
s=stp(:);
%t1=linspace(0,0.5,500);
t2=0:1/fs:(N/fb)-1/fs; % time axis
plot(t2,s,'r','linewidth',2);
xlabel('time');
ylabel('Amplitude');
%title('BFSK modulated waveform for the transmitted sequence (1 0 0 1 0)');
grid on;
% ************************************************************************************ End of Program  ************************************************************************************
clear all; 
close all;
clc;
Eb=1; % Set Bit Energy to 1
Smin=0; % Min Eb/N0, dB
Sinc=0.5; % Eb/N0 increment
Smax=6; % Max Eb/No, dB
Sm=Smin:Sinc:Smax; % Eb/No Range
SdB=10.^(Sm/10); % Eb/No, numerical
fb=1; % Bit frequency
fc=4*fb; % Carrier frequency
fs=4*fc; % Sampling frequency
i=0.5; % Spacing between the tones
fr=fs/fb; % Number of samples per bit
% N=12; % Number of bits for testing
cnt=0;
phi=2*pi*randn(1,1); % Random Phase for the user
for x=Smin:Sinc:Smax
    cnt=cnt+1;
    S=(10.^(x/10)); % Eb/N0 numerical
    Pba(cnt)=0.5*erfc(sqrt(S/2)); % Analytical Pb for coherent BFSK
    Ner=150; % Number of errors
    N(cnt)=ceil(Ner./Pba(cnt)) ; % Number of bits transmitted for each SNR
    Pbs(cnt)=0; % Initialize simulated probability vector
    xb=2*BIN01(N(cnt),0.5)-1; % Generating data bits for user
    tt=[];
    for ii=1:N(cnt)
        t(ii,:)=(ii-1)/fb:1/fs:ii/fb-1/fs; % Time vector for ii-th bit
        st(ii,:)=cos(2*pi*(fc+xb(ii)*fb*i/2).*t(ii,:)+phi); % FSK Transmitted Waveform, ii-th bit
        tt=[tt t(ii,:)];
    end
    stp=st'; % Transpose signal matrix for subsequent "vectorizing"
s=stp(:); \hspace{1cm} \% Re-shape signal matrix into vector

s_n=sqrt(Eb/sum(s(1:fr).^2))*s'; \hspace{1cm} \%Normalize the energy of the signal

s_len=length(s); \hspace{1cm} \%Length of the transmitted BFSK modulated waveform

y=sqrt(Eb/(2*S)); \hspace{1cm} \% Standard Deviation of Noise

noise=y*randn(1,s_len); \hspace{1cm} \% Define Noise

\%CHANNEL

r=s_n+noise; \hspace{1cm} \% Add AWGN to the transmitted waveform

\%RECIIVER

\%We first define the two correlator waveforms. At the correlator we
\%multiply with respect to the frequency tones of the transmitted signal
\%and then integrate it over one time period. So we define the
\%correlator signal with respect to each bit time

ttj=[];

for jj=1:N(cnt)
    tj(jj,:)=((jj-1)/fb:1/fs:jj/fb-1/fs;
    sn10(jj,:)=cos(2*pi*(fc+(fb*i/2)).*tj(jj,:)+phi);
    sn20(jj,:)=cos(2*pi*(fc-(fb*i/2)).*tj(jj,:)+phi);
    ttj=[ttj,tj(jj,:)];
end

\%The next 4 lines transpose the generated matrix and vectorize just
\%like we did for the transmitted signal

stp1=sn10';

stp10=stp1(:);

stp2=sn20';

stp20=stp2(:);

stp10_n=sqrt(1/sum(stp10(1:fr).^2))*stp10;

stp20_n=sqrt(1/sum(stp20(1:fr).^2))*stp20;

count=1;

\%This "for" loop multiplies the received signal with the correlator
\%signal and then integrates the product. The largest value is then
\%detected as a binary 0

for k=1:fr:s_len
    r1=((sum(r(k:fr+k-1).*stp10_n(k:fr+k-1)))); \hspace{1cm} \% Correlator 1
    r2=((sum(r(k:fr+k-1).*stp20_n(k:fr+k-1)))); \hspace{1cm} \% Correlator 2
    d(count)=r2<r1;\hspace{1cm} \% Largest value is detected as binary 0
    count=count+1;
end

e=2*d-1; \hspace{1cm} \% Converting the 0/1 to -1/+1

er=sum(abs(e-xb)/2); \hspace{1cm} \% Counting the number of errors

Pbs(cnt)=er./N(cnt); \hspace{1cm} \% Simulated Pb
end

semilogy(Sm,Pba,Sm,Pbs,'*-');
xlabel('E_b/N_0');
ylabel('P_b');
legend('Analytical P_b','Simulated P_b');
grid;

% ******************** End of Program ****************************

%******************************************************************
% Program simulates the performance of a single user DS-SS-BFSK modulated system
% The input parameters are
% E_b/N_0 range
% Bit rate, Rb
% Processing Gain, PG
% Carrier Frequency, f_c
% Sampling Frequency, f_s
% Spacing between frequency tones, i.
%******************************************************************

clear all;
close all;
clc;
Eb=1; % Set Bit Energy to 1
Smin=0; % Min E_b/N_0, dB
Sinc=0.5; % Eb/N0 increment
Smax=6; % Max Eb/No, dB
Sm=Smin:Sinc:Smax; % Eb/No Range
SdB=10.^(Sm/10); % Eb/No, numerical
fb=1; % Bit frequency
PG=10; % Processing Gain
Rc=PG*fb; % Chip Rate
fc=2*Rc; % Carrier frequency
fs=4*fc; % Sampling frequency
i=0.5; % Spacing between the tones
fr=fs/fb; % Number of samples per bit
% N=12; % Number of bits for testing
cnt=0;
phi=2*pi*randn(1,1); % Random Phase for the user
for x=Smin:Sinc:Smax
  cnt=cnt+1;
  S=(10^(x/10)); % Eb/N0 numerical
  Pba(cnt)=0.5*erfc(sqrt(S/2)) % Analytical Pb for coherent BFSK
  Ner=150; % Number of errors
  N(cnt)=ceil(Ner./Pba(cnt)); % Number of bits transmitted for each SNR
  Pbs(cnt)=0; % Initialize Simulated Probability vector
xb=2*BIN01(N(cnt),0.5)-1; %Generating data bits for user
code=2*BIN01(PG,0.5)-1; % Spreading Codes

tt=[];
for ii=1:N(cnt)
    t(ii,:)=(ii-1)/fb:1/fs:ii/fb-1/fs;
    st(ii,:)=cos(2*pi*(fc+xb(ii)*fb*i/2).*t(ii,:)+phi); % FSK Transmitted Waveform, ii-th bit
    tt=[tt t(ii,:)];
end
stp=st'; % Transpose signal matrix for subsequent "vectorizing"
s=stp(:); % Re-shape signal matrix into vector

s_n=sqrt(Eb/sum(s(1:fr).^2))*s'; %Normalize the energy of the signal
os=OverN(s_n,PG); %Oversample Transmitted Sequence

cs=repmat(code,1,length(s)); %Replicate spreading code for the length of the transmitted sequence
spr=os.*cs; %DS-Spread the waveform
spr=spr/sqrt(PG);
spr=spr(:); %Vectorize the spread waveform

s_len=length(s); %Length of the transmitted BFSK modulated waveform

%CHANNEL
y=sqrt(Eb/(2*S)); % Standard Deviation of Noise
noise=y*randn(1,length(spr)); % Define Noise
r=spr+noise; % Add AWGN to the transmitted waveform

%RECIEVER
rd=reshape(r,PG,length(s)); % Reshape received vector into an PG by s_len matrix
rde=rde';
bhat=rde*code'; % Recieved waveform
% We define the two correlator waveforms. At the correlator we multiply with respect to the frequency tones of the transmitted signal and then integrate it over one time period. So we define the correlator signal with respect to each bit time
ttj=[];
for jj=1:N(cnt)
    tj(jj,:)=(jj-1)/fb:1/fs:jj/fb-1/fs;
    sn10(jj,:)=cos(2*pi*(fc+fb*i/2).*tj(jj,:)+phi);
    sn20(jj,:)=cos(2*pi*(fc-fb*i/2).*tj(jj,:)+phi);
    ttj=[ttj,tj(jj,:)];
end
% The next 4 lines transpose the generated matrix and vectorize just like we did for the transmitted signal
stp1=sn10';
stp10=stp1(:);
stp2=sn20';
stp20=stp2(:);
stp10_n=sqrt(1/sum(stp10(1:fr).^2))*stp10;
stp20_n=sqrt(1/sum(stp20(1:fr).^2))*stp20;
count=1;

% This "for" loop multiplies the received signal with the correlator
% signal and then integrates the product. The largest value is then
% detected as a binary 0
for k=1:fr:s_len
    r1=(((sum(bhat(k:fr+k-1).*stp10_n(k:fr+k-1)))); % Correlator 1
    r2=(((sum(bhat(k:fr+k-1).*stp20_n(k:fr+k-1)))); % Correlator 2
    d(count)=r2<r1; % Largest value is detected as binary 0
    count=count+1;
end

e=2*d-1; % Converting the 0/1 to -1/+1
er=sum(abs(e-xb)/2); % Counting the number of errors
Pbs(cnt)=er./N(cnt) % Simulated Pb
end

semilogy(Sm,Pba,Sm,Pbs,'*-');
xlabel('E_b/N_0');
ylabel('P_b');
legend('Analytical P_b','Simulated P_b');
grid;

% ******************* End of Program ******************
Sinc=0.5; %Symbol Frequency
Smax=4; %Sampling frequency
Sm=Smin:Sinc:Smax; %Bit frequency
SdB=10.^(Sm/10); %Processing Gain
PG=10; %Carrier frequency
fb=1; fs=100; %Sampling frequency
PG=10; %Processing Gain
Rc=PG*fb;
fc=2*Rc; %Symbol Frequency
fs=4*fc; %Sampling frequency
fr=fs/fb;

%N=12; %Number of bits
cnt=0;
%xb=BIN01(N,0.5)%Generating Binary Sequence for FSK modulation
for x=Smin:Sinc:Smax
    cnt=cnt+1;
    S=(10^((x/10));
    Pba(cnt)=0.5*erfc(sqrt(S/2)) % Analytical Pb for coherent BFSK
    Ner=150; % Number of errors
    %N(cnt)=5;
    N(cnt)=ceil(Ner./Pba(cnt)) ; % Number of bits transmitted for each SNR
    Pbs(cnt)=0;
    %N(cnt)=50;
    xb=2*BIN01(N(cnt),0.5)-1; % Data bits
    xb2=2*BIN01(N(cnt),0.5)-1;
    code=2*BIN01(PG,0.5)-1; % Spreading Codes
    code2= 2*BIN01(PG,0.5)-1;
    tt=[];
    for ii=1:N(cnt)
        t(ii,:)=((ii-1)/fb:1/fs:ii/fb-1/fs; % Time vector for ii-th bit
        st(ii,:)=cos(2*pi*(fc+xb(ii)*fb*1).*t(ii,:)); % FSK Transmitted Waveform, ii-th bit
        %sn10(ii,:)=cos(2*pi*(fc-(fb*0.5)).*t(ii,:)); %Correlator Waveform when -1 is sent
        %sn20(ii,:)=cos(2*pi*(fc+(fb*0.5)).*t(ii,:)); %Correlator Waveform when +1 is sent
        tt=[tt t(ii,:)];
    end
stp=st'; % Transpose signal matrix for subsequent "vectorizing"
s=stp(:); % Re-shape signal matrix into vector
s_n=sqrt(Eb/sum(s(1:fr).^2))*s; %Normalize
os=OverN(s_n,PG);
cs=repmat(code,1,length(s));
spr=os.*cs;
spr=spr/sqrt(PG);
spr=spr(:);

2

tt2=[];
for ii2=1:N(cnt)
    t2(ii2,:)=(ii2-1)/fb:1/fs:ii2/fb-1/fs;         % Time vector for ii-th bit
    st2(ii2,:)=cos(2*pi*(fc+xb2(ii2)*fb*2).*t2(ii2,:)); % FSK Transmitted Waveform, ii-th bit
    %sn10(ii,:)=cos(2*pi*(fc-(fb*0.5)).*t(ii,:)); %Correlator Waveform when -1 is sent
    %sn20(ii,:)=cos(2*pi*(fc+(fb*0.5)).*t(ii,:)); %Correlator Waveform when +1 is sent
    tt2=[tt2 t2(ii2,:)];
end
stp2=st2';           % Transpose signal matrix for subsequent "vectorizing"
s2=stp2(:);           % Re-shape signal matrix into vector
s_n2=sqrt(Eb/sum(s2(1:fr).^2))*s2'; %Normalize
os2=OverN(s_n2,PG);
cs2=repmat(code2,1,length(s2));
spr2=os2.*cs2;
spr2=spr2/sqrt(PG);
spr2=spr2(:);

ttj=[];
for jj=1:N(cnt)
    tj(jj,:)=(jj-1)/fb:1/fs:jj/fb-1/fs;
    sn10(jj,:)=cos(2*pi*(fc+(fb*1)).*tj(jj,:));
    sn20(jj,:)=cos(2*pi*(fc-(fb*1)).*tj(jj,:));
    ttj=[ttj,tj(jj,:)];
end
stp1=sn10';
stp10=stp1(:);
stp2=sn20';
stp20=stp2(:);

%s_n=sqrt(Eb/sum(s(1:fr).^2))*s'; %Normalize
stp10_n=sqrt(1/sum(s(1:fr).^2))*stp10;
stp20_n=sqrt(1/sum(s(1:fr).^2))*stp20;
s_len=length(s);
y=sqrt(Eb/(2*S));                        % Standard Deviation of Noise
noise=y*randn(1,length(spr));          % Define Noise
r=spr'*spr2'+noise;
rd=reshape(r,PG,length(s));                   % Reshape received vector into an PG by s_len matrix
rde=rd';
bhat=rde*code';                      % Perform despreading
%RECIEVER
% Number of samples in each bit time
count=1;
ttc=[];
for i=1:fr:s_len
    r1=((sum(bhat(i:fr+i-1)'.*stp10_n(i:fr+i-1)''))); % Correlator 1
    r2=((sum(bhat(i:fr+i-1)'.*stp20_n(i:fr+i-1)''))); % Correlator 2
    d(count)=r2<r1;                        % Largest value is detected as binary 0
    count=count+1;
end
e=2*d-1;
er=sum(abs(e-xb)/2);                        % Counting the number of errors
Pbs(cnt)=er./N(cnt)                      % Simulated Pb
end
q=sum(s);
j=j-1;
semilogy(Sm,Pba,Sm,Pbs,'*-');
xlabel('E_b/N_0');
ylabel('P_b');
legend('Analytical P_b','Simulated P_b');
grid;

%END OF PROGRAM

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Program simulates the performance of a two user Asynchronous DS-SS-BFSK modulated system
% The input parameters are
% E_b/N_0 range
% Bit rate, Rb
% Processing Gain, PG
% Carrier Frequency, fc
% Sampling Frequency, fs
% Delay in the second waveform
%END OF PROGRAM

clear all;
clc;
close all;
% fb=10; %Bit frequency
% fs=100; %Sampling Frequency
fb=1; %Bit Frequency
fc=2*9*fb/4; % Carrier frequency
fs=4*(fc+2*9*fb/4); % Sampling frequency
fr=fs/fb; %number of samples for a bit time
fb1=40;
fs1=400;
fr1=fs1/fb1;

%Eb=1; % Bit energy normalized to one
ENmin=0; % Min Eb/N0 (dB)
ENmax=7; % Max Eb/N0 (dB)

t1=0:1/fs:(1/fb)-(1/fs); %To be used at the receiver as correlators
sn10=cos(2*pi*(fc-(fb*0.5))*t1);
sn20=cos(2*pi*(fc+(fb*0.5))*t1);
sn40=cos(2*pi*40*t1);
sn60=cos(2*pi*60*t1);
sn30=(sn10+sn20)/sqrt(2);

cnt=0;
PG=10; %Processing Gain
code=2*BIN01(PG,0.5)-1; %generate Spreading Code
code1=2*BIN01(PG,0.5)-1;

for x=ENmin:0.5:ENmax
    x
    cnt=cnt+1;
    S=(10^(x/10));
    Pba(cnt)=0.5*erfc(sqrt(S/2)) % Analytical Pb for coherent BFSK
    Ner=150; % Number of errors
    N(cnt)=ceil(Ner./Pba(cnt)) ; % Number of bits transmitted for each SNR
    Pbs(cnt)=0;
    %TRANSMITTER
    tt=[];
    %USER 1
    b=2*(randn(1,N(cnt)) < 0.5)-1; %Generating bits (0/1) for FSK modulation
    for ii=1:N(cnt)
        t(ii,:)=(ii-1)/fb:1/fs:ii/fb-1/fs; % Time vector for ii-th bit
        st(ii,:)=cos(2*pi*(fc+b(ii)*fb*0.5).*t(ii,:)); % FSK Transmitted Waveform, ii-th bit
        tt=[tt t(ii,:)];
    end
    stp=stp'; % Transpose signal matrix for subsequent "vectorizing"
    s=stp(:); % Re-shape signal matrix into vector
    s=sqrt(Eb/sum(sn10.^2))*s;
s_len = length(s);
ss = overN(s, PG);  % Oversample the transmitted waveform.

ss = repmat(code, 1, s_len);  % Replicate spreading code for the length of s(t)
ss = ss .* ss;  % DS-Spread the data
ss = ss / sqrt(sum(code.^2));

% USER 2
b1 = 2 * (randn(1, N(cnt)) < 0.5) - 1;  % Generating bits (0/1) for FSK modulation
%fb = 10;
%fs = 120;
tt1 = [];
shift = 0.7;
for i1 = 1:N(cnt)
    t1(i1,:) = (i1-1)/fb:1/fs:i1/fb-1/fs;  % Time vector for ii-th bit
    st1(i1,:) = cos(2*pi*(fc+b(i1)*fb*1).*(t1(i1,:)-shift));  % FSK Transmitted Waveform, ii-th bit
    tt1 = [tt1 t1(i1,:)];
end
stp1 = st1';  % Transpose signal matrix for subsequent "vectorizing"
sl = stp1(:,);  % Re-shape signal matrix into vector

s1 = sqrt(Eb / sum(sn40.^2)) * s1;
s_len_1 = length(s1);  % Length of s
ss1 = overN(s1, PG);  % Oversample the transmitted waveform.

% Shifting the code.
for i = 1:PG
    k = i + ceil(shift);
    if k > PG, k = k - PG; end;
code1(k) = code1(i);
end
cs1 = repmat(code1, 1, s_len_1);  % Replicate spreading code for the length of s(t)
ss1 = cs1 .* ss1;  % DS-Spread the data
ss1 = ss1 / sqrt(sum(code1.^2));

% CHANNEL
sum1 = (ss + ss1);
sum2 = sqrt(sum(ss.^2) / sum(sum1.^2)) * sum1;
y = sqrt(Eb / (2*S));  % Standard Deviation of Noise
noise = y * randn(1, length(ss));  % Define Noise
r = sum1 + noise;  % Received Waveform

% RECIEVER
rd = reshape(r, PG, s_len);  % Reshape received vector into an PG by s_len matrix
rde = rd';
bhat=rde*code';                          % Perform despreading
len=length(bhat);
inp=bhat';

% Demodulation
count=1;
for i=1:fr:len
   r1=(sum(inp(i:fr+i-1).*sn10)); % Correlator 1
   r2=(sum(inp(i:fr+i-1).*sn20)); % Correlator 2
   d(count)=r2>r1;               % Largest value is detected as binary 0
   count=count+1;
end
e=2*d-1;
% Received Binary Vector
er=sum(abs(e-b))/2;                    % Counting the number of errors
Pbs(cnt)=er./N(cnt)                    % Simulated Pb
end

EN=[ENmin:0.5:ENmax];                   % Eb/N0 Vector
figure;
semilogy(EN,Pba,'r',EN,Pbs,'b*-');grid;
xlabel('E_b/N_0, dB'); ylabel('P_b');
%title('P_b vs. E_b/N_0 for BFSK AWGN Channel');
%title([' P_b vs. E_b/N_0 for BFSK AWGN Channel PG=',num2str(PG),'
',Rb=',num2str(fb),',f_c=',num2str(fc),',f_s=',num2str(fs)]);
legend('P_b--Analytical','P_b--Simulated');

%**********************************************************************************
% Program simulates the performance of a single user Rayleigh faded DS-SS-BFSK modulated system
% The input parameters are
% E_b/N_0 range
% Bit rate, Rb
% Processing Gain, PG
% Carrier Frequency, fc
% Sampling Frequency, fs
% Spacing between frequency tones, i.
%**********************************************************************************

clear all;
clc;
close all;
fb=10;                                  % Bit frequency
fs=50;                                  % Sampling Frequency
fr=fs/fb;                        %number of samples for a bit time
Eb=1;                        % Bit energy normalized to one
ENmin=0;                     % Min Eb/N0 (dB)
ENmax=8;                     % Max Eb/N0 (dB)
t1=0:1/fs:(1/fb)-(1/fs);            %To be used at the reciever as correlators
sn10=cos(2*pi*(10)*t1);
sn20=cos(2*pi*(20)*t1);
cnt=0;
PG=10;
code=BIN01(PG,0.5);                  %DS-SS Spreading code
for x=ENmin:ENmax
  cnt=cnt+1;
  S=(10^((x/10)));
  Pba(cnt)=0.5*erfc(sqrt(S/2))            % Analytical Pb for coherent BFSK
  Pbray(cnt)=0.5*(1-sqrt(S./(2+S)));
  Ner=200;
  N(cnt)=ceil(Ner./Pba(cnt)) ;      % Number of errors
  Ns(cnt)=ceil(Ner./Pba(cnt));   % Number of bits transmitted for each SNR
  Pbs(cnt)=0;
  %TRANSMITTER
  [s,b]= call_fsk(N(cnt),fs,fb);       % FSK Transmitted Waveform
  s=sqrt(Eb/sum(sn20.^2))*s;            % Normalize the signal Energy
  s_len=length(s);
  s1=OverN(s,PG);                     %Oversample the transmitted waveform
  cs=repmat(code,1,length(s));                  %Replicate spreading code for the length of
  b1=cs.*s1;                              %DS-Spread the data
  b1=sqrt(Eb/sum(cs.^2))*b1;               %Normalize the energy of the signal
  %CHANNEL
  [r_f,phi]=rayleigh1(N(cnt),0);
  r_f_o=OverN(r_f,PG*fr);
  b_r=b1.*r_f_o;
  b_r_n=sqrt(sum(b1.^2)/sum(b_r.^2))*b_r;
  y=sqrt(Eb/(2*S))                           % Standard Deviation of Noise
  noise=y*randn(1,length(b1));                % Define Noise
  r=b_r_n+noise;
  rd=reshape(r,PG,length(s));            % Reshape received vector into an PG
  by s_len matrix
  rde=rd';                                  % Perform despreading
  bhat=rde*code';                            % Recieved Binary Vector
  er=sum(abs(e-b));                             % Counting the number of errors
  Pbs(cnt)=er/N(cnt)                            % Simulated Pb
end
%Plotting the curves
EN=[ENmin:ENmax]; %Eb/N0 Vector
figure;
semilogy(EN,Pbray,'r',EN,Pbs,'b*-',EN,Pba,'d-'); grid;
xlabel('E_b/N_0, dB'); ylabel('P_b');
%title('P_b vs. E_b/N_0 for DS-SS-BFSK AWGN and Rayleigh fading Channel');
legend('P_b--Analytical Rayleigh','P_b--Simulated Rayleigh','P_b--AWGN');
axis([0 ENmax 1e-4 1e-0]);

%****************************** End of Program ******************************

%This program plots the power spectrum of unspread BFSK case. T
clear all;
clc;
close all;
%%%%%%%USER
Rb=1; % Bit frequency
f_c=2*9*Rb/4;
f_s=2*(f_c+2*9*Rb/4); % Sampling Frequency
Eb=1; % Bit energy normalized to one
fsTc=5; % normalized sampling frequency
Nbp=128; % number of bits/trial
Nr=100; % number of trials (realizations)
ss1=2^(ceil(log2(fsTc*Nbp))); % nearest (larger) power of 2 for fft
spect=zeros(1,ss1); % initialize spectrum vector for v
f=0:f_s/ss1:f_s-1/ss1; % frequency vector for plotting
t=0:t_s/1/Rb-1/t_s;

sn10=cos(2*pi*(f_c+Rb*.25)*t+2*rand(1,1)*pi);
tt=[]; % Initialize time vector, for output & plotting
for k=1:Nr
    b=2*(rand(1,Nbp) < 0.5)-1; % Generating bits (1/-1) for FSK modulation
    for ii=1:Nbp
        t(ii,:)=(ii-1)/Rb:1/f_s:ii/Rb-1/f_s; % Time vector for ii-th bit
        st(ii,:)=cos(2*pi*(f_c+b(ii)*Rb*1).*t(ii,:)+2*rand(1,1)*pi); % FSK Transmitted Waveform, ii-th bit
    end
    stp=st'; % Transpose signal matrix for subsequent "vectorizing"
s=stp(:); % Re-shape signal matrix into vector
    spect=spect+((abs(fft(s,ss1))).^2)/Nr; % Power Spectrum of the Spread BFSK Signal
end
m=max(spect);
lspect=spect/m;
lspect=10*log10((spect/m)); % spectrum in dB, normalized
figure;
plot(f,lspect(1:ss1/1+1*0));
title('BFSK Spectrum, dB, f_c=4.5Hz');xlabel('frequency/fs');
grid;

%%%%%%%%%%%%%%%%%%%%%%%% USER
2%%%%%%%%%%%%%%%%%%%%%%%%

% Bit frequency
f_c_1=(2*9*Rb_1)/4;
f_s_1=2*(f_c_1+(2*9*Rb_1)/4); % Sampling Frequency
Eb=1; % Bit energy normalized to one
fsTc=5; % normalized sampling frequency
Nbp=128; % number of bits/trial
Nr=100; % number of trials (realizations)
ss2=2^ceil(log2(fsTc*Nbp)) % nearest (larger) power of 2 for fft
spect_1=zeros(1,ss1); % initialize spectrum vector for v
f_1=10*(0:f_s_1/ss2:f_s_1-1/ss2); % frequency vector for plotting
t_1=0:1/f_s_1:1/Rb_1-1/f_s_1;
sn10=cos(2*pi*(f_c_1+b(ii)*Rb_1.5).*t_1+2*rand(1,1)*pi); % FSK Transmitted Waveform, ii-th bit
for k1=1:Nr
  b_1=2*(rand(1,Nbp) < 0.5)-1; % Generating bits (-1,1) for FSK modulation
  for ii_1=1:Nbp
    t_1(ii_1,:)=(ii_1-1)/Rb_1:1/f_s_1:ii_1/Rb_1-1/f_s_1; % Time vector for ii-th bit
    st1(ii_1,:)=cos(2*pi*(f_c_1+b(ii_1)*Rb_1)*t_1+2*rand(1,1)*pi);
    tt_1=[tt_1 t_1(ii_1,:)]; % Initialize time vector, for output & plotting
  end
  stp1=st1'; % Transpose signal matrix for subsequent "vectorizing"
s1=stp1(:); % Re-shape signal matrix into vector
  s1=sqrt(Eb/sum(sn10.^2))*s1;
spect_1=spect_1+((abs((fft(s1',ss1))).^2))/Nr; % Power Spectrum of the Spread BFSK Signal
end
m_1=max(spect_1);
lspect=spect/m;
lspect_1=10*log10((spect_1/m_1)); % spectrum in dB, normalized

%%%%%%%%%%%%%%%%%%%%%%%% USER
3%%%%%%%%%%%%%%%%%%%%%%%%
Rb=1; % Bit frequency
f_c=2*9*Rb/4;
fs=2*(f_c+2*9*Rb/4); % Sampling Frequency
Eb=1; % Bit energy normalized to one
fsTc=5; % normalized sampling frequency
Nbp=128; % number of bits/trial
Nr=100; % number of trials (realizations)
ss1=2^(ceil(log2(fsTc*Nbp))) % nearest (larger) power of 2 for fft
spect1=zeros(1,ss1); % initialize spectrum vector for v
f=0:fs/ss1:fs-1/ss1; % frequency vector for plotting
t=0:1/fs:Rb-1/fs;

sn10=cos(2*pi*(f_c+Rb*.25)*t+2*rand(1,1)*pi);
tt=[]; % Initialize time vector, for output & plotting
for k=1:Nr
    b=2*(rand(1,Nbp) < 0.5)-1; % Generating bits (1/-1) for FSK modulation
    for ii=1:Nbp
        t(ii,:)=(ii-1)/Rb:1/fs:ii/Rb-1/fs; % Time vector for ii-th bit
        st(ii,:)=cos(2*pi*(f_c+b(ii)*Rb).*t(ii,:)+2*rand(1,1)*pi); % FSK Transmitted Waveform, ii-th bit
    end
    stp=st'; % Transpose signal matrix for subsequent "vectorizing"
    s=stp(:); % Re-shape signal matrix into vector

    s=sqrt(Eb/sum(sn10.^2))*s;
    spect1=spect1+((abs(fft(s',ss1)).^2)/Nr); % Power Spectrum of the Spread BFSK Signal
end
m1=max(spect1);
lspect1=10*log10((spect1/m1)); % spectrum in dB, normalized
figure;
plot(f,lspect1(1:ss1/1+1*0),'g',f,lspect1(1:ss1/1+1*0),'b',f,lspect1(1:ss1/1+1*0),'r'); %title('BFSK Spectrum, dB');
xlabel('frequency');
ylabel('PSD,dB');
legend('User 1','User 2','User 3');
axis([0 fs/2 -60 0]);grid;

% *********************** End of Program %%%%

% This program plots the Power Spectrum of DS-SS-BFSK.
clear all;
clc;
close all;
Nb=128; % Number of bits
Nr=50;  % Number of realizations
PG=32    % Processing gain
Rb=1;    % Bit Rate
Re=PG*Rb;  % Chip rate
fc=4*Rc;  % Carrier Frequency

fs=4*(fc+2*Rc);  % Sampling Frequency for the FSK transmitted waveform
fr=fs/Rc;  % Normalized Sampling Frequency

nofft=2^(ceil(log(PG*Nb)/log(2)))  % nearest (larger) power of 2 for fft

f=0:fs/nofft:fs-1/nofft;  % Frequency axis
pspect=zeros(1,nofft);  % Initialize PS vector for Spread case

for i=1:Nr
    %b=ones(1,Nb);  % All ones for testing
    b=2*(rand(1,Nb)<0.5)-1;  % Generating bits (1/-1) for FSK modulation
    tt=[];  % Initialize time vector, for output
    % Generating the FSK waveform
    for ii=1:Nb
        t(ii,:)=(ii-1)/Rb:1/fs:ii/Rb-1/fs;  % Time vector for ii-th bit
        st(ii,:)=cos(2*pi*(fc+b(ii)*Rb)*10.*t(ii,:)+2*randn(1,1)*pi);  % FSK Transmitted Waveform, ii-th bit
        tt=[tt t(ii,:)];
    end
    stp=st';  % Transpose signal matrix for subsequent "vectorizing"
    s=stp(:);  % Re-shape signal matrix into vector

    % We now spread the waveform and find the power spectrum
    code=2*BIN01(PG,0.5)-1;  % Generating Spreading code
    codeS=OverN(code,fr);  % Oversample the spreading code
    % Next line oversamples signal s? s is already sampled at fs, so what is this?
    %os=OverN(s',Re*fr);  % Oversample the transmitted waveform
    %repc=repmat(codeS,1,length(s));  % Replicate Spreading code for the length of waveform
    repc=repmat(codeS,1,Nb);
    %spr=repc.*os;  % DS Spread the code
    spr=repc.*s';
    spr=spr/sqrt(sum(spr.^2));  % Normalize the energy of the spread waveform
    p=(abs(fft(spr,nofft))).^2;  % FFT of the transmitted waveform
    pspect=pspect+(p/Nr);  % Power Spectrum
    %clear b st spect s;
end
smax=10*log10((pspect)/max(pspect));  % Normalized Power Spectrum, in dB
figure; plot(f,smax,'r'); grid;
title(['Power Spectrum, DS-SS-BFSK, PG=' num2str(PG) 'Rb=',num2str(Rb),'fs=8*Rc']);
xlabel('Frequency/f_s');
axis([0 fs/2 -30 0]);
% ****************************************End of Program*************

% Function BIN01.m generates a random binary vector x, with elements in set {0,1}
% Probability of a 1 is an input parameter p0, and length of x is N.
% Syntax y=BIN01(N,p0), where 0<= p0 <=1

function xb = BIN01(N,p0)
xb=(rand(1,N) < p0);

% End of Function %

function [r, phi] = rayleigh1(Ns,u)
% [r, phi] = rayleigh1(Ns,u)
% % A Rayleigh fading simulator based on a "conventional lowpass"
% response
% for the power spectrum (using a 2 pole elliptic filter), a reasonable
% approximation to the "Clarke" spectrum, which assumes isotropic
% scattering
% about the mobile, which has an isotropic antenna. This model more
% accurate
% in suburban/rural areas.
% A FILTER MUST BE SELECTED TO SET THE RELATIVE DOPPLER
% FREQUENCY
% %
% % INPUTS:
% % Ns = # samples of the Rayleigh fading process to produce
% % u = parameter to select filter; 0:no filtering, 1:fDT=0.01,
% % 2:fDT=0.1
% %
% % OUTPUTS:
% % r = row vector containing Ns samples of the Rayleigh fading
% % process
% % phi= row vector containing Ns samples of the phase
% %
% j=sqrt(-1);
% % SELECT ONE OF THE FILTERS BELOW
% if u == 0; % no filter, for memoryless fading
% aa=[1]; bb=[1];
end
if u == 1
% Filter #1, fdT=0.01 (or, fd/fs=0.01)
    aa=[1 -1.9346 0.9376];  bb=[3.772 -4.7 3.772]*1e-3;
end
if u == 2
% Filter #2, fdT=0.1 (or, fd/fs=0.1)
    aa=[1 -1.295 0.5296];  bb=[5.765 0.1064 5.765]*1e-2;
end

% generate the Gaussian i.i.d. r.v.'s, w/100 "extra" samples for
% initialization
Iin = randn(1,Ns+100);
Qin = randn(1,Ns+100);

% pass the input components through the filters
Iout = filter(bb,aa,Iin);
Iout = Iout(101:Ns+100);
Qout = filter(bb,aa,Qin);
Qout = Qout(101:Ns+100);

phi = angle(Iout + j*Qout);  % Compute phase vector
r = sqrt(Iout.^2 + Qout.^2);  % Compute Rayleigh magnitude vector

% normalize for E[r^2]=1
rms = sqrt( mean( r.*r ) );
r = r/rms;

% ********** End of Function***********

% Function OverN oversamples input vector x by N
% thus, for example, if N=3,
% x=[x(1) x(2) ... x(M)] becomes y=[x(1) x(1) x(1) x(2) x(2) x(2) ... x(M) x(M) x(M)]
% not yet generalized for matrices

function y=OverN(x,N)
Lx=length(x);
y=zeros(1,N*Lx);  % initialize oversampled vector y
for kk=1:Lx      % loop to create y
    for jj=1:N
        y((kk-1)*N+jj)=x(kk);
    end
end

% ********** End of Function *************