Order Number 1348473

Quick simulation of multi-channel direct sequence spread spectrum communication systems

Al-Turki, Faisal Turki Ali, M.S.

The University of Arizona, 1992

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

QUICK SIMULATION OF MULTI-CHANNEL DIRECT SEQUENCE SPREAD SPECTRUM COMMUNICATION SYSTEMS

by
Faisal Turki Ali Al-Turki

A Thesis Submitted to the Faculty of the
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
In Partial Fulfillment of the Requirements
For the Degree of
MASTER OF SCIENCE
WITH A MAJOR IN ELECTRICAL ENGINEERING
In the Graduate College
THE UNIVERSITY OF ARIZONA

1 9 9 2

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: FAISAL ALTURKI Saig

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

Pamela Nielsen — May 5 1992

Pamela A. Nielsen
Assistant Professor of
Electrical and Computer Engineering — Date

Randall K. Bahr — 5/5/92

Randall K. Bahr, Co-Director — Date

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Randall K. Bahr for his never ending patience and help, and for his guidance in the work leading to the completion of this thesis. I would also like to thank Bruce R. Ward for his help in innumerable ways, and all my friends for their support and their help in finishing this work.

TO MY FAMILY

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

With the increasing popularity of spread spectrum techniques, one would like to evaluate the merit of these methods by studying system bit error rates (BER) in the presence of noise and other interfering signals. Unfortunately in the case of spread spectrum systems, easily evaluated expressions for the BER do not exist, therefore one must resort to estimation via simulation techniques. Since the BER of a typical digital communication system is very low, extensive computations are required to adequately estimate this probability. One way to overcome this problem is to apply the method of importance sampling, which attempts to "speed up" the estimation of small quantities by altering the underlying probability distribution. In this thesis, an importance sampling method known as Quick Simulation is used to find the BER of a spread spectrum system. The Quick Simulation method is compared to two other methods for estimating the BER.

CHAPTER 1

INTRODUCTION

In this thesis we examine a direct sequence multiple access spread spectrum communication system and try to obtain a good estimate for the average probability of bit error via the method of quick simulation developed from the theory of importance sampling using the principles of large deviation theory [3]. One of the most important parameters in judging a digital communication system is its bit error rate, BER. In evaluating the BER one is often required to evaluate the expected value of functions of random vectors. Due to the complex stochastic nature, nonlinearities, and the additive noise (e.g., Gaussian) that corrupt these systems, it is very difficult to obtain exact closed form expressions to evaluate the BER. As a result, one usually resorts to numerical techniques (simulations) to estimate these expectations (i.e., error probabilities). The most widely used simulation method for this purpose is the Monte Carlo simulation.

The idea of the Monte Carlo simulation technique is to count the number of times a certain event of interest occurs in N trials, giving an estimate of the expected value or the probability of that event. These estimates will converge to the actual value as the number of simulation runs N tends to infinity. For our purpose the event of interest is the probability of decoding a bit erroneously at the receiver of the communication system. Let K be the number of bits decoded erroneously, then the estimated value for the probability of bit error is $\mathrm{P}_e = \rho \triangleq K/N$. An error is expected approximately every ρ^{-1} runs. A typical digital communication system (DCS) has BER in the range of $10^{-4} - 10^{-8}$. Therefore, in order to obtain a good estimate for the BER, N must be taken very large. The number of simulation trials must be taken at least 10 times ρ^{-1} to achieve an acceptable level of accuracy, hence an excessive number of simulation runs may be required to adequately estimate the BER. This could take an inordinate length of time. Furthermore, if N is taken very large (e.g., $N > 10^7$) the number of random variables required may exceed the period of the random number generator. The random number generator will no longer be random, and therefore it becomes imperative to use other more efficient methods, for the error estimation.

One way to circumvent these difficulties is to utilize importance sampling methods. The idea of importance sampling is to reduce the number of required simulation trials necessary to estimate the expectation or probability for events of interest by making the low probability events occur more frequently [13], [14]. One way to do that is to modify the probability density function (pdf) of the input random process by biasing the input signal probability distribution (via stretching or translation). This results in an increase in the occurrence of rare events, creating a biased estimator. The bias can be removed by proper weighting of the estimator to get the actual expectations or probabilities. The goal is to select a biased distribution in a smart way so that one obtains a good estimate with the smallest number of simulation runs. There are several methods used to implement this idea. One method used for this purpose is the quick simulation method developed from the principles of large deviation theory (LDT) [2], [3].

This thesis is divided into five chapters. Chapter II gives an introduction to spread spectrum communication systems: how a system is defined as a spread spectrum communication system; the basic idea of operation; and advantages and disadvantages. A development of the direct sequence spread spectrum communication system model used in this thesis is also presented. In Chapter III we derive the test statistics for determining the probability of bit error from our model, and present the first method for estimating the BER, which is based on conventional Monte Carlo simulation techniques. Chapter IV is devoted to the development of importance sampling and large deviation theory from which the method of quick simulation is developed. The second method for estimating the BER which is based on the quick simulation technique is presented. In Chapter V we develop our last method for estimating the BER which is based on numerical integration of the system pdf. The work is concluded by comparing the quick simulation technique with the other techniques, and summarizing the results obtained.

CHAPTER 2

General Concepts of Spread Spectrum Systems

2.1 Background

Spread spectrum is a means of transmission or modulation in which the bandwidth of the modulated signal is spread well beyond the bandwidth of the modulating signal and independently of the modulating signal bandwidth.

For a communication system to be defined as a spread spectrum system it must satisfy the following conditions [1]:

1) The modulated signal occupies a bandwidth much in excess of the minimum bandwidth necessary to send the information.
2) Spreading is accomplished by means of a spreading signal which is independent of the data. This spreading signal is called a Pseudo Random Code or a PN sequence.
3) At the receiver, despreading is accomplished by generating an exact synchronized replica of the PN sequence used to spread the information. This replica is correlated with the received spread signal to produce the original unspread signal.

Originally, spread spectrum systems were used for military applications, mainly to achieve highly jam-resistant communication systems. However, in recent years spread spectrum techniques have been used for civilian applications. Communication experts believe that spread spectrum systems have a very promising future in the field of communication systems.

Spread Spectrum communication systems have several applications such as [1]:

1) Anti-jam capability, particularly for narrow band jamming.
2) Interference rejection and selective calling.
3) Multiple-access capability.
4) Multipath fading protection, especially in mobile communications.
5) Covert communications in military operations and low probability of intercept.
6) Secure communications in civilian applications.
7) Improved spectral efficiency.
8) Determination of position location or providing range measuring capability.

There are several ways to classify spread spectrum systems. The most common way of classification is by modulation method. These modulation methods include [1]:

1) Direct sequence (pseudo noise).
2) Frequency hopping.
3) Time hopping.
4) Chirp.
5) Hybrid methods.

In this thesis we examine the technique of direct sequence spread spectrum and find an estimate for the probability of bit error as a function of signal-to- noise ratios (SNR) and other parameters of interest.

Direct Sequence Spread Spectrum System Model

2.2 Basic Operation

In this section a general model for the direct sequence spread spectrum system is presented with a brief description of its basic operations. In general there are several methods for designing these systems [1]. Figure (2.1) shows a typical design model that is used frequently in real life applications.

General block diagram of a direct sequence spread spectrum communication system

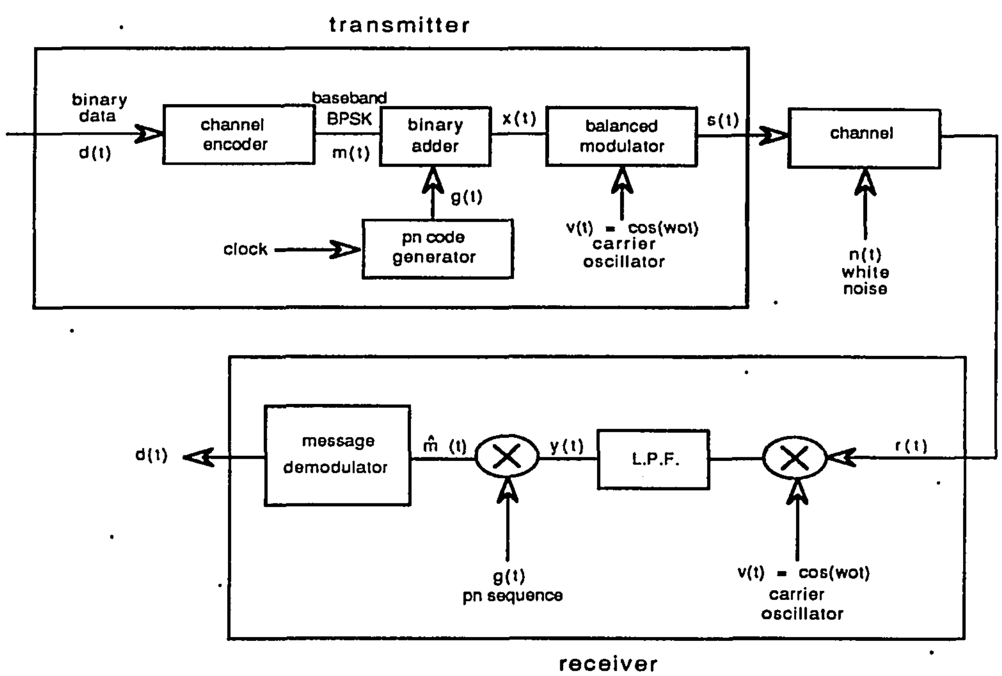


Figure (2.1)

In direct sequence spread spectrum systems any coherent digital communication modulation technique can be used to modulate the information string; however, Binary Phase Shift Keying (BPSK) is the most common method used for that purpose. The data bits are encoded using, for example, polar signaling, the resultant signal is then modulated by the pseudo noise (PN) sequence to produce a baseband direct sequence spread spectrum signal. From Figure (2.1) consider the following.

Let $d(t)$ represent a string of binary information bits that is

$$d(t) = \sum_{i=-\infty}^{\infty} a_i u(t - iT_b) \tag{2.1}$$

where a_i is either binary "1" or binary "0" and $u(t)$ is a pulse of unit height and width T_b. This information is encoded using polar signaling, the resultant signal is equivalent to baseband BPSK and can be represented mathematically as

$$m(t) = \sum_{n=-\infty}^{\infty} (2a_i - 1)u(t - nT_b) \tag{2.2}$$

where $m(t)$ represents the baseband BPSK message, and n is an index used to refer to the n^{th} information bit symbol. For convenience, binary "1" is represented by a positive pulse and binary "0" is represented by a negative pulse. $m(t)$ is multiplied by the pseudo noise (PN) sequence to produce a baseband spread spectrum signal. Note that the pseudo noise is not actually random at all [1]. The sequence is called "random" because it has the statistical properties of a sampled white noise and the sequence appears as a random signal to unfriendly listeners. The baseband spread spectrum signal can be represented mathematically as

$$x(t) = g(t) \cdot m(t) \tag{2.3}$$

where $g(t)$ represents the spreading code. In order to transmit $x(t)$ through the channel it is multiplied by a very high frequency (i.e., $f_o \gg \frac{1}{T_c}$) local carrier oscillator $v(t) = \sqrt{2p}\cos w_o t$, where p is the power in the signal $x(t)$. The resultant passband spread spectrum signal can be written as

$$s(t) = m(t) \cdot v(t) \cdot g(t). \tag{2.4}$$

This signal is transmitted through a channel. The channel adds in white Gaussian noise of infinite bandwidth and constant power spectral density over all frequencies. Since we are considering a multiple access system, other interfering direct sequence spread spectrum signals are added and hence considered as noise corrupting the desired signal. At the receiver the received signal is first multiplied by a synchronized replica of the carrier oscillator and then passed through a low pass filter with cutoff frequency high enough to maintain most of the energy in the PN sequence pulses. When the signal is passed through the LPF the signal with frequency component at twice the carrier frequency will be eliminated, and only baseband spread spectrum signals and finite energy Gaussian noise will remain. The output of the filter is multiplied by an exact replica of the spreading code of the desired signal and then passed through an integrate and dump filter which act as a demodulator to retrieve the original data.

Even though there are several different methods of modulation and design for spread spectrum systems, the direct sequence spread spectrum system is the most common method used by communication engineers because of its interesting features and advantages. Some of the direct sequence method advantages include [1]:

1) Best noise and anti-jam performance compared to other spread spectrum systems.
2) Direct Sequence method is the most difficult to detect because of its low probability of intercept compared to other spread spectrum systems.

3) Best discrimination against multipath with respect to other modulation methods.

4) Can be used for multiple-access and code division multiple-access communication systems.

On the other hand, like any other communication system, direct sequence spread spectrum systems suffer from several disadvantages, mainly [1]:

1) Spread spectrum systems require wide-band channel with little phase distortion.

2) Long acquisition time. Acquisition refers to the process of accomplishing synchronization between the locally generated signal and the received signal. Due to elapsed time since last communication, or the relative motion between transmitter and receiver, an error may occur in interpreting the right time or frequency of the code clock at the receiver. The time it takes the receiver to obtain maximum correlation, which indicates that synchronization is achieved, is called the acquisition time. This time is relatively long.

3) Fast code generator needed. In general very fast circuitry is needed to implement the PN sequence of the code generator. This circuit may be costly and difficult to design.

4) Near-far problem. In this problem very strong signals at the receiver swamp out the effect of weaker signals.

In the next chapter we present a mathematical derivation of the test statistics for estimating the probability of bit error for the direct sequence multiple access spread spectrum communication system

CHAPTER 3

Derivation Of The Test Statistic For The Multi-Channel Direct Sequence Spread Spectrum Model

3.1 System Analysis

Figure 3.1 shows the model for deriving the test statistics for our direct sequence multiple access spread spectrum communication system [3].

Block diagram of a direct sequence multiple-access spread spectrum communication system used to derive the test statistic

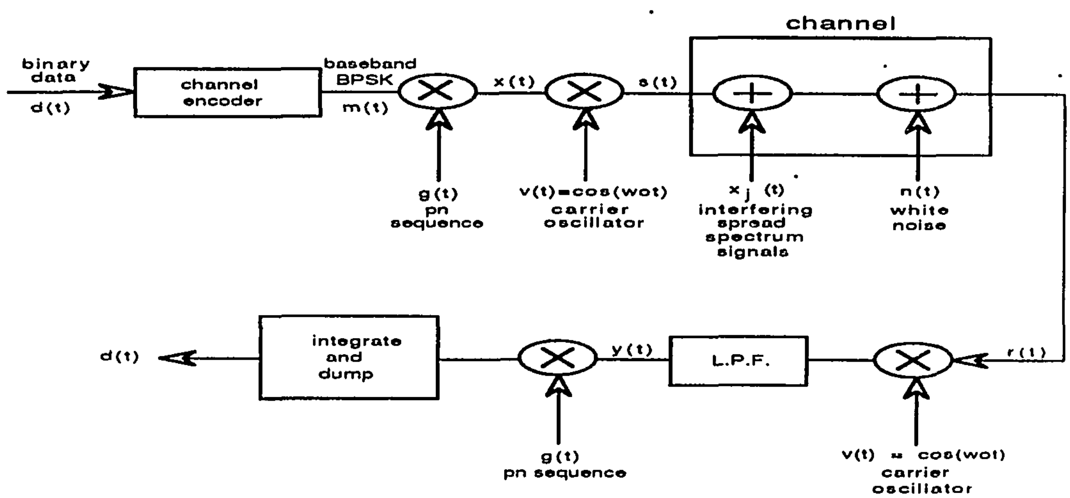


Figure (3.1)

Based on the model shown in Figure (3.1) consider the following.

$$\text{let } d(t) = \sum_{i=-\infty}^{\infty} a_i u(t - iT_b) \tag{2.1}$$

represents the digital data which is encoded using polar format to give the baseband BPSK signal. This BPSK signal is represented by

$$m(t) = \sum_{n=-\infty}^{\infty} (2a_i - 1)u(t - nT_b). \tag{2.2}$$

The PN sequence $g(t)$ modulates $m(t)$ to give the baseband spread spectrum signal which can be written as

$$x(t) = A\sum_{k=1}^{n} U_k p_c(t-(k-1)T_c). \tag{3.1}$$

This represents the equivalent baseband spread spectrum signal for one data bit having length T_b . In the expression for $x(t)$ we have

A is the amplitude of the baseband BPSK signal.

U_k is a discrete independent identically distributed (i.i.d.) random sequence with values of ±1, modelling the PN sequence.

$p_c(t)$ is a rectangular pulse of unit height and width T_c. Note that the summation of the product of U_k and $p_c(t-(k-1)T_c)$ represents the pseudo noise spreading code or the PN sequence length; i.e,

$$g(t) = \sum_{k=1}^{n} U_k P_c(t-(k-1)T_c) \quad \text{where}$$

T_c is the chip time of the pseudo noise code.

n is the PN sequence length or the ratio of the information bit T_b to the chip time T_c. Figure (3.2) on the next page shows a typical PN sequence.

Typical pseudo noise sequence

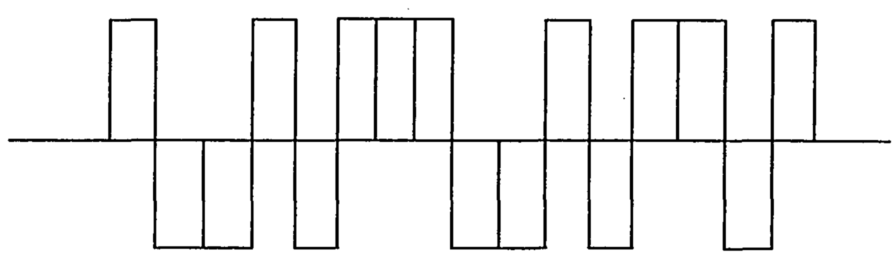


Figure (3.2)

Referring to Figure(3.1), let

$$\sum_{j=1}^{m} x_j(t) = \sum_{j=1}^{m} A_j \sum_{k=1}^{n} V_k^j p_c(t-(k-1)T_c) \tag{3.2}$$

represent some other direct sequence spread spectrum interfering signals, where

m is the number of interfering signals.

A_j is the amplitude of the baseband BPSK for the other interfering signals.

V_k^j is an i.i.d. random sequence with values ± 1, modelling the PN sequence of the j^{th} interfering spread spectrum signal.

$n(t)$ is Gaussian white noise with spectral density $N_o/2$.

$\cos w_o t$ is the carrier oscillator signal.

Each information bit is represented by a positive or negative pulse of amplitude A and time duration of T_b. The spreading signal has pulses of duration T_c where T_b is an integer multiple of T_c; i.e, $T_b = nT_c$. The received signal $r(t)$ consists of the original spread spectrum signal, plus m interfering signals and additive white noise [3]; i.e.,

$$r(t) = x(t)\cos w_o t + n(t) + \sum_{j=1}^{m} x_j(t)\cos(w_o t + \phi_j). \tag{3.3}$$

This received signal is multiplied by a replica of the carrier oscillator signal $\cos w_o t$, where it is assumed that the desired spread spectrum signal carrier is synchronized with the carrier signal. Hence, we have

$$2r(t)\cos w_o t = 2x(t)\cos^2 w_o t + 2n(t)\cos w_o t + 2\sum_{j=1}^{m} x_j(t)\cos(w_o t + \phi_j)\cos(w_o t).$$

Note that " 2 " is just a normalization factor.

This signal is passed through an ideal low pass filter (LPF) with cutoff frequency $f_c = \frac{j}{T_c}$ where j is an integer of the same order or equal to n. In general, the cutoff frequency of the filter should be large enough to accommodate all of the energy in the PN sequence pulses. The output of the filter is

$$y(t) = x(t) + 2n_o(t) + \sum_{j=1}^{m} x_j(t)\cos\phi_j,$$

where $n_o(t)$ is the result of passing $n(t)\cos w_o t$ through the LPF . $2n_o(t)$ is Gaussian noise with finite energy; however, since the bandwidth of the LPF is very large compared to the bandwidth of the PN sequence pulses, it is reasonable to model the bandlimited Gaussian noise as a white noise; i.e. , $2n_o(t) \simeq n(t)$. In general, it is a common assumption to analyze baseband system with additive white noise as a worse case analysis [1], [9]. Hence the output of the filter can be expressed as

$$y(t) = x(t) + \sum_{j=1}^{m} x_j(t)\cos\phi_j + n(t) \tag{3.4}$$

which represents the total received signal at baseband.

The next step is to multiply the received signal by an exact replica of the spreading code and pass it through the integrate and dump filter. Define Δ as the output of the integrator, then

$$\Delta = \int_0^{nT_c} g(t) \cdot y(t)\, dt \tag{3.5}$$

$$\Delta = \int_0^{nT_c} [x(t)g(t) + g(t)\sum_{j=1}^{m} x_j(t)\cos\phi_j + n(t)g(t)]\, dt.$$

Now partition the above into three separate integrals and analyze each one separately. First consider

$$\int_0^{nT_c} x(t)g(t)\, dt = \int_0^{nT_c} A\sum_{k=1}^{n}\sum_{l=1}^{n} U_k U_l p_c(t-(k-1)T_c)p_c(t-(l-1)T_c)\, dt.$$

$p_c(t)$ is a rectangular pulse of duration T_c and $p_c(t-(k-1)T_c)$ is $p_c(t)$ shifted by an amount T_c $(k-1)$, hence

$$U_k U_l p_c(t-(k-1)T_c)\cdot p_c(t-(l-1)T_c) = \begin{cases} {p_c}^2(t-(k-1)T_c), & \text{if } k=l; \\ 0, & \text{otherwise}. \end{cases}$$

Therefore

$$\int_0^{nT_c} x(t)g(t)\, dt = A\left[\int_0^{T_c} p_c^2(t)\, dt + \cdots + \int_{(n-1)T_c}^{nT_c} p_c^2(t-(n-1)T_c)\, dt\right].$$

Define $P \triangleq \int_0^{T_c} p_c^2(t)\, dt$ as the energy in pulse p_c. Since every shifted pulse has the same energy, that is

$$\int_0^{T_c} p_c^2(t)\, dt = \int_{T_c}^{2T_c} p_c^2(t-T_c)\, dt = \cdots = \int_{(k-1)T_c}^{kT_c} p_c(t-(k-1)T_c)\, dt,$$

we have

$$\int_0^{nT_c} x(t)g(t)\, dt = A\sum_{k=1}^{n}\int_{(k-1)T_c}^{kT_c} {p_c}^2(t-(k-1)T_c)\, dt$$

and since $x(t) = A\cdot g(t)$ then

$$A\int_0^{nT_c} g^2(t)\, dt = A\sum_{k=1}^{n}\int_0^{T_c} {p_c}^2(t)\, dt = \sum_{k=1}^{n} P\cdot A = nPA.$$

Next consider the second integral in the Δ expression; i.e.,

$$\int_0^{nT_c} n(t)g(t)\, dt = \int_0^{nT_c} n(t)\sum_{k=1}^{n} U_k p_c(t-(k-1)T_c)\, dt$$

$$= \sum_{k=1}^{n} U_k \int_{(k-1)T_c}^{kT_c} n(t) p_c(t-(k-1)T_c)\, dt.$$

But

$$\int_0^{T_c} n(t)p_c(t)\,dt = \int_{T_c}^{2T_c} n(t)p_c(t-T_c)\,dt = \ldots = \int_{(n-1)T_c}^{nT_c} n(t)p_c(t-(n-1)T_c)\,dt.$$

Define

$$N_k \triangleq \int_{(k-1)T_c}^{kT_c} n(t)p_c(t-(k-1)T_c)\,dt.$$

N_k is a Gaussian random variable with mean equal to

$$E[N_k] = \int_{(k-1)T_c}^{kT_c} E[n(t)]p_c(t-(k-1)T_c)\,dt.$$

Since $n(t)$ is zero mean, $E[N_k] = 0$. The variance of N_k is

$$\mathrm{Var}[N_k] = E[{N_k}^2] - E^2[N_k] = E[{N_k}^2]$$

$$= \int_{(k-1)T_c}^{kT_c} \int_{(k-1)T_c}^{kT_c} E[n(t)n(\tau)p_c(t-(k-1)T_c)p_c(\tau-(k-1)T_c)d\tau\, dt].$$

Since the autocorrelation of the white noise is $\Phi(t-\tau) \triangleq E[n(t)n(\tau)] = \frac{N_o}{2}\delta(t-\tau)$,

$$\mathrm{Var}[N_k] = \int_{(k-1)T_c}^{kT_c} \int_{(k-1)T_c}^{kT_c} \frac{N_o}{2}\delta(t-\tau)p_c(t-(k-1)T_c)p_c(\tau-(k-1)T_c)d\tau\, dt$$

$$= \int_{(k-1)T_c}^{kT_c} \frac{N_o}{2}{p_c}^2(t-(k-1)T_c)\,dt = \frac{N_o}{2}\int_{(k-1)T_c}^{kT_c} {p_c}^2(t-(k-1)T_c)\,dt = \frac{N_o}{2}P.$$

So finally we have

$$\int_0^{nT_c} n(t)g(t)\,dt = \sum_{k=1}^{n} U_k N_k$$

where N_k is a Gaussian random variable with zero mean and variance $\frac{N_o}{2}P$. The last part of the Δ expression is

$$\int_0^{nT_c} g(t)\sum_{j=1}^{m} x_j(t)\cos\phi_j\,dt = \sum_{j=1}^{m}\cos\phi_j \int_0^{nT_c} g(t)x_j(t)\,dt$$

$$= \sum_{j=1}^{m} A_j \cos\phi_j \int_0^{nT_c} \sum_{k=1}^{n} U_k p_c(t-(k-1)T_c) \sum_{k=-\infty}^{+\infty} V_k^j p_c(t-(k-1)T_c - \tau_j)$$

where V_k^j is an i.i.d. discrete random sequence with values of ± 1. The V_k^j represent the PN sequence of the j^{th} interfering spread spectrum signal and τ_j represent the phase shift in the PN sequence due to lack of synchronization. The above integral can be written as

$$\sum_{j=1}^{m} A_j \cos\phi_j \sum_{k=1}^{n} \sum_{l=1}^{n} U_k V_l^j \int_0^{nT_c} p_c(t-(k-1)T_c) p_c(t-(l-1)T_c - \tau_j)\, dt.$$

Recall that $0 \le \tau_j \le T_c$ therefore the product of the over lapping pulses will be zero for $t \ge T_c$ and $t \le 0$ Figure (3.3) shows the product of these overlapping pulses.

Product of two overlapping PN sequence pulses

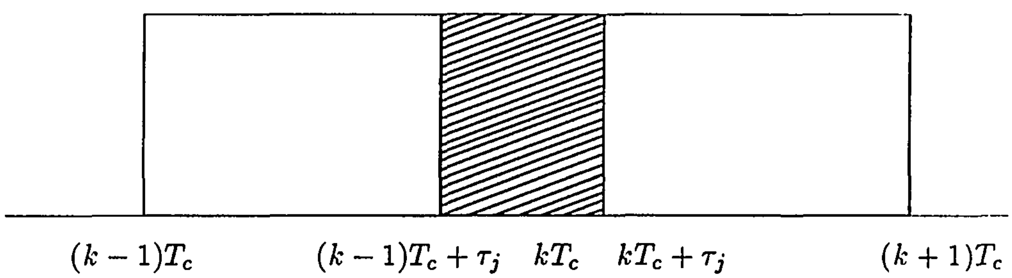


Figure (3.3)

So

$$\int_0^{nT_c} p_c(t-(k-1)T_c) p_c(t-(l-1)T_c - \tau_j)\, dt = \int_{(k-1)T_c+\tau_j}^{kT_c} 1^2\, dt + \int_{kT_c}^{kT_c+\tau_j} 1^2\, dt.$$

The first integral on the right has the value τ_j and the second integral has the value $T_c - \tau_j$. Recall that $p_c(t)$ is a rectangular pulse of unit amplitude and width T_c. Hence,

$$P = \int_0^{T_c} p^2{}_c(t)\, dt = \int_0^{T_c} (1)^2\, dt = T_c.$$

One can rewrite τ_j and $T_c - \tau_j$ as follows :

$$\tau_j = P \cdot \frac{\tau_j}{T_c} \quad \text{and} \quad T_c - \tau_j = P(1 - \frac{\tau_j}{T_c}).$$

So

$$\int_0^{nT_c} g(t)x_j(t)\,dt = A_j \sum_{k=1}^{n}\sum_{l=1}^{n} U_k V_l^j \left[P\tau_j/T_c + P(1-\tau_j/T_c)\right]$$

and

$$\sum_{j=1}^{m} \cos\phi_j \int_0^{nT_c} g(t)x_j(t)\,dt = \sum_{j=1}^{m} A_j \cos\phi_j \sum_{k=1}^{n}\sum_{l=1}^{n} U_k V_l^j [Pa_{\tau_j} + Pb_{\tau_j}]\,dt,$$

where $a_{\tau_j} = (1 - \frac{\tau_j}{T_c})$ and $b_{\tau_j} = \frac{\tau_j}{T_c}$. Note that b_{τ_j} is obtained due to the shift in the pulse with coefficient V_{k-1}^j and $a\tau_j$ is obtained due to the pulse with coefficient V_k^j. From the previous analysis we have

$$\sum_{k=1}^{n}\sum_{l=1}^{n} U_k V_l^j = \begin{cases} \sum_{k=1}^{n} U_k V_k^j, & \text{if } k = l; \\ 0, & \text{otherwise.} \end{cases}$$

So

$$\sum_{k=1}^{m} \cos\phi_j \int_0^{nT_c} g(t)x_j(t)\,dt = \sum_{j=1}^{m} A_j \cos\phi_j \sum_{k=1}^{n} \left[a\tau_j P U_k V_k^j + b\tau_j P U_k V_{k-1}^j\right]$$

$$= A \cdot P \sum_{k=1}^{n}\sum_{j=1}^{m} \frac{A_j}{A} \cos\phi_j \left[(1 - \frac{\tau_j}{T_c}) U_k V_k^j + \frac{\tau_j}{T_c} U_k V_{k-1}^j\right].$$

Collect all terms at the output of the integrator and define S_n as the result of that collection; i.e.,

$$S_n = \sum_{k=1}^{n} A \cdot P + \sum_{k=1}^{n} U_k N_k + A \cdot P \sum_{k=1}^{n}\sum_{j=1}^{m} \frac{A_j}{A} \cos\phi_j \left[a_{\tau_j} U_k V_k^j + b_{\tau_j} U_k V_{k-1}^j\right].$$

Define

$$Z_k = A \cdot P + U_k \cdot N_k + A \cdot P \sum_{j=1}^{m} \frac{A_j}{A} \cos\phi_j \left[a_{\tau_j} U_k V_k^j + b_{\tau_j} U_k V_{k-1}^j\right]$$

and

$$S_n = \sum_{k=1}^{n} Z_k.$$

For convenience we normalize S_n by a factor

$$\frac{\sqrt{2}}{\sqrt{n}AP.}$$

Let $\widetilde{S_n} \triangleq \frac{S_n.\sqrt{2}}{\sqrt{n}AP}$ then

$$\widetilde{S_n} = \sum_{k=1}^{n} \frac{\sqrt{2}}{\sqrt{n}AP} AP \left[1 + \frac{U_k N_k}{AP} + \sum_{j=1}^{n} \frac{A_j}{A} \cos\phi_j [a_{\tau_j} U_k V_{kj} + b_{\tau_j} U_k V_{k-1}^j] \right].$$

Define $\widetilde{N_k} = \frac{N_k\sqrt{2}}{\sqrt{n}AP}$ then

$$\widetilde{S_n} = \sum_{k=1}^{n} \sqrt{\frac{2}{n}} \left[1 + U_k \widetilde{N_k} + \sum_{j=1}^{m} \frac{A_j}{A} \cos\phi_j [a_j U_k V_k^j + b_j U_k V_{k-1}^j] \right] \tag{3.6}$$

Note that $E[\widetilde{N_k}] = 0$ and $\text{Var}[\widetilde{N_k}] = \frac{N_o}{nA^2P}$. Define the SNR for the spread spectrum signal as $\frac{nA^2P}{N_o}$, then the variance of $\widetilde{N_k}$ is $\frac{1}{SNR}$. Since P represents the energy in the pseudo noise pulse and the period of the BPSK is $T_b = nT_c$, the energy in each information bit is $nA^2P = E_b$.

We are interested in estimating the probability of bit error. Assuming the binary bit "1" was sent we need to find P{receiving "0"|"1" is sent}. The test statistic S_n is compared to threshold value 0. This is done by implementing a detector which performs the following test. If $\widetilde{S_n} > 0$, say "1" was sent and if $\widetilde{S_n} < 0$ say "0" was sent. In the next section we present the first method for estimating the BER using the conventional Monte Carlo technique based on the test statistics and the threshold detector developed above.

3.2 Estimation via the Monte Carlo method

In this section the first method for estimating the BER is presented using the conventional Monte Carlo technique. All random functions are simulated using computer random function generators. The code emulates the test statistics, and estimate the BER by finding the number of bits which are decoded erroneously in the total number of simulation runs. The variance of the output estimator, S_n is also computed. Later this variance is compared to the variance of the quick simulation method. The code is written to simulate the performance of the communication system as realistically as possible. The probability of error is computed for different random values of time shifts τ_j and phase shifts ϕ_j. The results are averaged to obtain an overall P_e.

The procedure is repeated for different values of n and different values of SNR's. The tables shown on the next two pages show the results for estimating the BER and the variance using the conventional Monte Carlo method. A plot of the results for the BER is shown in Figure (3.4). The pros and cons of the conventional Monte Carlo method are discussed at the conclusion in Chapter (6). It is very difficult to determine how accurate and reliable the results obtained are for several reasons. First, we are simulating the output S_n, and it is nearly impossible to determine the exact values for the BER since an easily evaluated closed form expression for P_e does not exist . Second, to our knowledge no published results exist which analyze the exact same system using the conventional monte carlo technique, therefore it is very difficult to compare the results obtained. Third, the monte carlo method is a counting method. Therefore, to get better results one must increase the size of the simulation trials in order for the probabilities to converge to their actual values. Hence for high values of SNR and n, this method may not be very reliable, since the event of interest may not occur frequently enough to get good estimate. Forth, the output S_n is a random function and contains random parameters. To obtain better results one must run the simulation for many different values of ϕ_j and τ_j and then average the results. In our analysis we used fifty different random values of ϕ_j and τ_j and averaged the results. Despite all the above impairments, we think the results obtained are reasonably accurate. The values are compared to the other methods of estimation (see [12]) and to the quick simulation method. The results obtained are within the bounds for the BER under different values of SNR and n. More about the results obtained are presented in Chapter 4 when compared with the quick simulation results.

TABLE 3.1
Estimate for the BER using Monte Carlo method n=10

Estimating the BER and the variance for n=10		
SNR	probability of error	estimator variance
0 dB	9.2166×10^{-2}	3.9437×10^{-1}
5 dB	1.9164×10^{-2}	2.3421×10^{-2}
10 dB	3.5323×10^{-3}	2.1283×10^{-3}
15 dB	8.6079×10^{-4}	3.5619×10^{-4}
20 dB	2.3228×10^{-4}	6.3724×10^{-5}

TABLE 3.2

Estimate for the BER using Monte Carlo method n=20

Estimating the BER and the variance for n=20		
SNR	probability of error	estimator variance
0 dB	8.6221×10^{-2}	6.8631×10^{-1}
5 dB	1.0572×10^{-2}	1.7581×10^{-2}
10 dB	3.6824×10^{-4}	2.1585×10^{-4}
15 dB	4.4239×10^{-5}	1.6038×10^{-5}
20 dB	3.6410×10^{-6}	7.1667×10^{-7}

TABLE 3.3

Estimate for the BER using Monte Carlo method n=30

Estimating the BER and the variance for n=30		
SNR	probability of error	estimator variance
0 dB	8.3729×10^{-2}	9.6505×10^{-1}
5 dB	9.5042×10^{-3}	2.2025×10^{-3}
10 dB	1.3072×10^{-4}	7.6127×10^{-5}
15 dB	2.9210×10^{-6}	9.4790×10^{-7}
20 dB	4.0125×10^{-7}	1.2230×10^{-7}

TABLE 3.4

Estimate for the BER using Monte Carlo method n=40

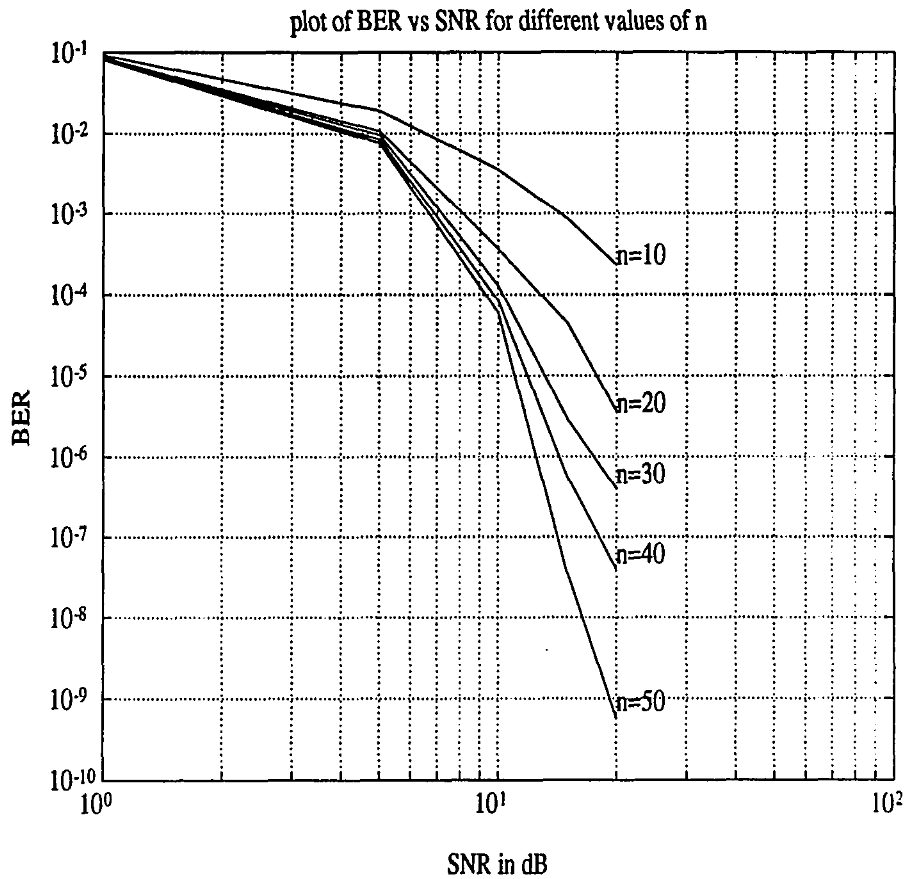
Estimating the BER and the variance for n=40		
SNR	probability of error	estimator variance
0 dB	8.1909×10^{-2}	1.2274
5 dB	8.3712×10^{-3}	2.3992×10^{-2}
10 dB	8.4610×10^{-5}	5.9709×10^{-5}
15 dB	1.6102×10^{-7}	1.8246×10^{-8}
20 dB	4.0125×10^{-8}	9.3675×10^{-9}

TABLE 3.5

Estimate for the BER using Monte Carlo method n=50

Estimating the BER and the variance for n=50		
SNR	probability of error	estimator variance
0 dB	8.13089×10^{-2}	1.5131
5 dB	7.5695×10^{-3}	6.2314×10^{-2}
10 dB	6.0159×10^{-5}	5.1545×10^{-5}
15 dB	7.7210×10^{-8}	5.4790×10^{-9}
20 dB	5.6125×10^{-9}	3.2230×10^{-10}

Figure (3.4) on the next page shows a plot of the BER vs SNR for different values of n using conventional Monte Carlo method.



plot of BER vs SNR for different values of n
BER
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
10^{-10}
10^{0}
10^{1}
10^{2}
n=10
n=20
n=30
n=40
n=50
SNR in dB

Figure (3.4)

CHAPTER 4

Quick Simulation Of Direct Sequence Spread Spectrum Systems

4.1 Introduction

In this chapter, we present the second method for estimating the BER using the quick simulation technique. The quick simulation method is developed from basic concepts of large deviation theory (LDT) and importance sampling [2] and [3]. Before we present this method, some of the basic concepts of importance sampling and LDT are introduced.

Often in statistical analysis, one is interested in determining the expected value of some random function; i.e., $E[g(X)]$ where X is an M-dimensional random vector with distribution P and $g(\cdot)$ is a function of the random vector X. In many applications such as evaluating the probability of bit error of a DCS, this expectation can not be evaluated directly. Either the underlying probability density function is unknown, or the expectation may be too difficult to analytically compute. Consequently one is left with attempting to bound this value from above and below, or to estimate it from some set of sample data. In the later approach one usually resorts to simulation, such as Monte Carlo techniques, to get a good estimate. As was discussed before, there are serious problems when applying this technique directly. These problems manifest themselves in the unrealistic large amount of time and computation required to adequately estimate the BER. In some cases, one way to overcome this problem is to utilize a notion from sampling theory known as importance sampling [3].

Importance sampling is a monte carlo simulation technique in which simulation data is generated using a probability distribution different from the true underlying distribution. One way to do this is via stretching or translating the density function of the input random variables [3]. In the process of stretching, the probability density function (pdf) is stretched by increasing the variance; in the process of translation the mean of the pdf is shifted in the direction of the event of interest by increasing the mean. Both cases result in an increase in the relative frequency of occurrence for events of interest ("important" events) and a possible reduction in the estimator variance.

Since the event of interest occurs more frequently the estimate of the expected value, the probability will be biased. One therefore must scale the importance sampling estimator by a weight function to get the unbiased value of the estimate [3]. This weight function is expressed as a likelihood ratio of the true underlying distribution to the biasing distribution. The above can be expressed mathematically as follows [2], [8].

Let $(X_1, X_2, \cdots, X_n)$ be a set of n independent identically distributed random variables with common density function $p(\cdot)$. We wish to estimate the probability that this set falls in a specific observation space E_n in the n-dimensional sample space. When the Monte Carlo technique is used to estimate the probability, L independent realizations of $(X_1, X_2, \cdots, X_n)$ are generated. L is a large integer representing the number of simulation runs. The estimate is

$$\mathrm{P_e} = \frac{1}{L}\sum_{l=1}^{L} 1_{E_n}(X_1^l, X_2^l, \cdots X_n^l) \tag{4.1}$$

where $(X_1^l, X_2^l, \cdots, X_n^l)$ is the l^{th} independent realization generated via simulation and 1_{E_n} represent the indicator function defined as follows

$$1_{E_n}(x_1, \cdots, x_n) \triangleq \begin{cases} 1 & \text{if } (x_1, \cdots, x_n) \in E_n; \\ 0 & \text{if } (x_1, \cdots, x_n) \notin E_n. \end{cases}$$

To apply importance sampling to the above estimate we generate a new biasing density $q(\cdot)$ that is different from the true underlying density $p(\cdot)$ [2], [3]. The weight function $W(\cdot)$ is defined as the Radon-Nikodym derivative of the unbiased distribution to the biased distribution; i.e.,

$$W(x) \triangleq \frac{p(x)}{q(x)}. \tag{4.2}$$

The importance sampling probability estimate becomes

$$\hat{\mathrm{P}}_e = \frac{1}{L}\sum_{l=1}^{L} 1_{E_n}(X_1^l, X_2^l, \cdots X_n^l) W(X_1^l, X_2^l, \cdots X_n^l). \tag{4.3}$$

The purpose of the weight function $W(\cdot)$ is to produce an unbiased estimator of the probability of the event. This can be proven easily by taking the expectation of $\hat{\mathrm{P}}_e$ with respect to the new density $q(\cdot)$; i.e., $E[\hat{\mathrm{P}}_e] = \mathrm{P}_e$. Since $p(x) \geq 0$ and $q(x) \geq 0$, $W(\cdot)$ is always nonnegative. An important parameter to examine when applying importance sampling techniques is the variance of the estimator which is easily computed to be

$$\mathrm{Var}[\hat{\mathrm{P}}_e] = \frac{1}{L}(\eta_{E_n}(q) - \mathrm{P}_e^2) \tag{4.4}$$

where

$$\eta_{E_n}(q) = E\left[\left(1_{E_n}(\cdot)W(\cdot)\right)^2\right] \tag{4.5}$$

$$= \int_{E_n} \left[\frac{p}{q}(x_1, x_2, \cdots, x_n)\right]^2 q(x_1, x_2, \cdots x_n) dx_1 dx_2 \cdots dx_n. \tag{4.6}$$

For importance sampling to be useful the variance of the importance sampling estimator in eq (4.4) must be less than the variance of the conventional Monte Carlo estimator. This is because the theory of importance sampling shows that the size of the variance is directly related to the size of simulation trials. The smaller the size of the variance, the smaller the number of simulation trials are required to obtain a predetermined estimator variance. Therefore one seeks to minimize the variance of the biased estimator with respect to the unbiased estimator.

Close examination of eq(4.4) shows that, one can minimize Var[$\hat{\mathrm{P}}_e$] by minimizing $\eta_{E_n}(q)$ since L and $\mathrm{P_e}$ are constant quantities. The expression for $\eta_{E_n}(q)$ in eq(4.6) shows that by proper choice of a biasing distribution $q(\cdot)$ one can minimize $\eta_{E_n}(q)$ and hence minimize Var[$\hat{\mathrm{P}}_e$]. The solution can be obtained by use of Jensen's inequality [2]

$$\eta_{E_n}(q) \geq \left[\int 1_{E_n}(x_1, x_2, \cdots x_n) \frac{p}{q}(x_1, \cdots x_n) q(x_1, \cdots x_n) dx_1 \cdots dx_n \right]^2 .$$

Hence,

$$\eta_{E_n}(q) \geq \mathrm{P}_e^2$$

with $\eta_{E_n}(q) = \mathrm{P_e}^2$ if and only if $1_{E_n} \cdot \frac{p}{q}$ is constant with respect to q; i.e.,

$$p(\cdot) = c \cdot q(\cdot)$$

where c is some constant. Under this condition one can achieve a unique optimal minimization for the estimate variance by selecting

$$q(x_1, x_2, \cdots, x_n) = \mathrm{P}_e^{-1} \cdot 1_{E_n}(x_1, x_2, \cdots, x_n) p(x_1, x_2, \cdots, x_n).$$

However this solution is not practical because it involves precisely the quantity we are trying to estimate. Therefore, one must seek another biasing density $q(\cdot)$ which results in a solution as close as possible to the optimal solution and minimizes the variance of the importance sampling estimator. Three possible methods for achieving that goal are frequently used. These methods differ in their techniques to bias the distribution of the input random variables, consequently they differ in the selection of the weight function $W(\cdot)$ which will lead to an optimum solution. These three methods are [2]:

1) mean bias method: this technique seeks to increase the occurrence of $\{S_n \leq T\}$ by adding a small mean value, ξ to the input random variables.
2) variance scaling method: this technique seeks to increase the occurrence of $\{S_n \leq T\}$ where S_n is the test statistics and T is the threshold, by multiplying the variance of the inputs random variables by ξ^2.
3) quick simulation method.

Quick simulation is a method developed from the principles of large deviation theory and is based upon the observation that for a fixed number of simulation runs the variance of the important sampling estimator in eq (4.3) vanishes exponentially fast as n approaches infinity [2], [3].

Quick simulation is utilized in this thesis when applying the method of importance sampling to estimate the BER. The goal here is to calculate the exponential rate of decrease at which the estimator variance decays using the tools of LDT. Once the rate function is calculated, one can obtain the desired biased density which is used in the weight function to remove the bias of the estimator and obtain the probability estimate with minimum number of simulation runs. To understand how this is done, we first start by recalling the Weak Law of Large Numbers [17]. Let $X_1, X_2, X_3, ..., X_n$ be independent random variables having a common distribution with mean μ and finite variance. Define $S_n = X_1 + X_2 + X_3 + ... + X_n$. Then for any $\delta \geq 0$,

$$\lim_{n\to\infty} P\Big(\Big|\frac{S_n}{n} - \mu\Big| \geq \delta\Big) = 0; \tag{4.7}$$

i.e., $\frac{S_n}{n}$ converges in probability to μ as n increases. By adapting a result of Cramér's work [2] one can show that these probabilities decay with an exponential rate as n increases; i.e.,

$$P(|\frac{S_n}{n} - \mu| \geq \delta) \leq \exp(-n \cdot M) \tag{4.8}$$

for some $M > 0$ and all n greater than some N_o. This can be stated using LDT terms [2] as

$$\lim_{n \to \infty} \frac{1}{n} \log P\Big(|\frac{S_n}{n} - \mu| \geq \delta\Big) = -I(\delta). \tag{4.9}$$

where $I(\cdot) = M$ is the so-called large deviation rate function defined as the Legendre/Fenchel transform of the moment generating function $c(\cdot)$ [2], [4]

$$I(x) \triangleq \sup_{\theta}\{\theta \cdot x - c(\theta)\} \tag{4.10}$$

and

$$c(\theta) \triangleq \lim_{n \to \infty} \frac{1}{n} \log E\big[exp(\theta \cdot X_n)\big]. \tag{4.11}$$

When the quick simulation technique is applied to importance sampling methods, certain conditions must be satisfied [2]. First, the sequence $\sum_{k=1}^{n} X_k$, which consisted of independent and identically distributed (i.i.d) random variables, is converted into a set of dependent random variables. This dependency is represented by a first order finite state space irreducible Markov chain with transitional probabilities $q(\cdot|\cdot)$ and a steady state stationary distribution $Q(\cdot)$. Second, it is required that the moment generating function $c(\cdot)$ exist. Furthermore it is assumed that the probabilistic settings in eq (4.8) and eq (4.9) hold when the quick simulation method is applied.

When the conditions above are met, the first step in the quick simulation technique is to determine the rate function. Once the ratė function is determined one can apply it to importance sampling strategies, and that will allow one to design simulations which will speed up the computations for events of interest. To see how this is done one proceeds as follows.

Recall that we are interested in finding

$$\mathrm{P}_e \triangleq \mathrm{P}\{S_n < 0\}.$$

It can be shown that $\mathrm{P_e}$ obeys a large deviation principle [2]; i.e., this probability decays with an exponential rate given precisely by the rate function $I(\cdot)$ as

$$\lim_{n\to\infty} \frac{1}{n} \log \mathrm{P_e} = -I(T)$$

where T is a threshold set by the system designer. Note that when the quick simulation is applied the new density is represented by a first order Markov process with transition density $q(\cdot|\cdot)$. From eq(4.10) it can be shown that the rate function can be expressed as [3]

$$I(T) = \theta_o T - \log \lambda(\theta_o) \tag{4.12}$$

where $\lambda(\theta_o)$ is the largest positive eigenvalue of the first order Markov chain transition density matrix, which represents the new altered biased transition density, which will optimize the rate function, and θ_o is the value which yield the largest eigenvalue for that matrix. Once the rate function of the transition density matrix is determined, one can predict the precise rate at which $\hat{\mathrm{P}}_e$ and the estimator variance decays. Since the rate function controls the exponential rate of decrease for the variance of the importance sampling estimator, one would like to maximize it to achieve optimum results. To maximize the rate function consider the following. Define the " speed factor " [2] $\mathrm{SF}(\hat{\mathrm{P}}_e)$ as

$$SF(\hat{\mathrm{P}}_e) = \lim_{n\to\infty} \frac{1}{n} \log(L\mathrm{Var}\{\hat{\mathrm{P}}_e\}).$$

The speed factor is maximized when $K\text{Var}\{\hat{\text{P}}_e\}$ vanishes at the fastest rate. The objective of the quick simulation is to select a new transition density matrix $q(\cdot|\cdot)$ to maximize $I(T)$ such that $\text{Var}\{\hat{\text{P}}_e\}$ vanishes at the greatest possible exponential rate. This is equivalent to maximizing the speed factor which from eq (4.4), can be written as

$$\eta_{E_n}(q) = \text{P}_\text{e}{}^2 + L \cdot \text{Var}\hat{\text{P}}_\text{e}.$$

Since $\text{P}_\text{e}{}^2$ is constant, $\eta_{E_n}(q)$ is proportional to $L \cdot \text{Var}\{\hat{\text{P}}_e\}$. Therefore maximizing the speed factor is equivalent to minimizing $\lim_{n\to\infty} \frac{1}{n}\log(\eta_{E_n}(q))$. Since

$$\eta_{E_n}(q) = \text{P}_\text{e}{}^2 + L \cdot \text{Var}\{\hat{\text{P}}_e\}$$

$$\text{and} \;\; I(T) = \frac{-1}{n}\log(\text{P}_\text{e}),$$

one can show that

$$\lim_{n\to\infty} \frac{1}{n}\log(\eta_{E_n}(q)) \geq -2I(T).$$

Thus the best we can hope for is [2]

$$\lim_{n\to\infty} \frac{1}{n}\log(\eta_{E_n}(q)) = -2I(T).$$

Under this condition the transition simulation density will be termed "efficient".

The definition of efficiency is based upon the asymptotics on n as the number of simulation runs L remains fixed [2]. However, the asymptotic analysis is pertinent to simulation implementation. From LDT principles, as n increases the importance sampling estimator vanishes exponentially fast, however n is usually fixed by the system designer. Therefore it seems reasonable that better estimations require more simulation runs as n increases. A big question which arises in this kind of simulation is how many simulation runs L are required to adequately estimate the BER. Consider the relative precision $\sqrt{\text{Var}\{\hat{\text{P}}_\text{e}\}}/\text{P}_\text{e}$ and define $K_n(q)$ as the total number of simulation trials to estimate P_e to some fixed a *priori* precision. One can show [2] that $K_n(q) \sim \exp(Rn)$ for $R \geq 0$ and let

$$R \triangleq 2 \cdot I(T) + \lim_{n\to\infty} \frac{1}{n} \log\left(\eta_{E_n}(q)\right)$$

recall that $2I(T) = -\lim_{n\to\infty} \frac{1}{n} \log\left(\eta_{E_n}(q)\right)$ under optimum conditions, in that case $R = 0$ which implies that an optimal result has been achieved and $q(\cdot|\cdot)$ is termed efficient. Therefore, by proper choice of $q(\cdot|\cdot)$, the exponential growth factor can be avoided in the simulation and thus speedup the estimate of P_e.

In the following section the principles of LDT are applied to the direct sequence spread spectrum communication system of interest and we show how to develop the test statistics under the new conditions. In the last section of this chapter we give an estimate to the BER and the variance of the estimator using the quick simulation method.

4.2 System Analysis Using Quick Simulation

In this section the principles and tools of large deviation theory are applied to the direct sequence multiple access communication system of interest to develop the test statistics for estimating the BER. The theorems from which these principles are developed are beyond the scope of this thesis (but see [2], [3]). Therefore the formulas used in the derivation will be given without proof. In order to apply LDT, the communication system must be modeled as a Markov chain; that is, U_k and V_k^j which are i.i.d. random sequences, will no longer be i.i.d. Due to this dependency of the random variables in the system, it is convenient to model the system as a first-order discrete state Markov chain. The new biased distribution of the input random variables will be represented by a modification of the state transition matrix of the chain.

The goal of the quick simulation is to select a density function in a smart way to maximize the exponential rate of decrease in the variance of the importance sampling estimator. As was discussed in the previous section this will lead to the desired rate function which in essence will speedup the process of computing the BER with minimum number of simulation runs, and minimize the variance of the estimator. As we mentioned above when the system is modeled as a first-order Markov chain, the altered density $q(\cdot)$ is represented by a transition matrix. The size of that matrix depends on the number of signals accessing the system. For our problem we chose two interferences; i.e., $m = 2$. The system is then modeled as a $2^{m+1} = 8$ [4] state first order Markov chain. The eight-state markov chain is described by an 8×8 transition matrix. To derive this matrix we proceed as follows. Recall that U_k correspond to the PN sequence of the desired signal. V_k^j corresponds to the PN sequence of the j^{th} interference signal. Since we have two interferences, let $V_k^{(1)}$ correspond to the first interference and $V_k^{(2)}$ correspond to the second interference. Define the following vectors [4]

$$\vec{X} \triangleq (x_1, x_2, \cdots, x_{m+1}) \text{ and } \vec{Y} \triangleq (y_1, y_2, \cdots, y_{m+1})$$

where x_1 and y_1 correspond to U_k and x_j, y_j where $j > 1$ correspond to the j^{th} interference V_k^j. Applying this to our model we have

$$\vec{X} = (U_k, V_k^{(1)}, V_k^{(2)}) \text{ and } \vec{Y} = (U_k, V_k^{(1)}, V_k^{(2)}).$$

Since U_k, $V_k^{(1)}$, $V_k^{(2)}$ each have two possible values ± 1 there are a total of eight different combinations for X and eight different combinations for Y. Therefore we have an 8×8 transition matrix. Define the transition density matrix $p_{x|y}(s)$ as follows [4]

$$p_{x|y}(s) \triangleq 2^{-(m+1)} \exp(\frac{s^2 \sigma^2}{2}) \exp(s \cdot g_{y|x})$$

where

$$g_{y|x} = \sqrt{\frac{2}{n}} \left[1 + y_{m+1} \Big(\sum_{j=1}^{m} (\frac{A_j}{A}) \cos(\phi_j) [(1 - \frac{\tau_j}{T_c}) y_j + \frac{\tau_j}{T_c} x_j] \Big) \right]$$

and $\sigma^2 = \mathrm{Var}(\widetilde{\mathrm{N}}_k) = \frac{1}{\mathrm{SNR}}$. The transition matrix $||p_{x|y}(s)||$ is a function of the variable s. To find the transition density which maximizes the rate function $I(\cdot)$ one needs to find the value of s which gives the largest positive eigenvalue of the transition matrix. Denote this value as s^* and let $\lambda_{max}(s^*)$ represent the largest eigenvalue and $r(\cdot, s^*)$ represent the right eigenvector which correspond to s^*. Define the twisted distribution $q_{x|y}$ as the transition matrix which maximizes the rate function; i.e., $q_{x|y}$ represents the new biased density of the input random variables. $q_{x|y}$ can be written as

$$q_{x|y} = \frac{2^{-(m+1)}}{\lambda_{max}(s^*)} \frac{r(y, s^*)}{r(x, s^*)} \exp(\frac{(s^*)^2 \sigma^2}{2}) \exp(s^* \cdot g_{y|x}).$$

From the transition matrix $q_{x|y}$ one can find the steady state distribution denoted by $q(X_1(j))$ by taking the left eigenvector of $q_{x|y}$ which correspond to the largest eigenvalue of the matrix. Once the steady state distribution is found, one can establish the test statistic for the importance sampling estimator as follows. Recall that the weight function is the ratio of the original density to the new biased density $q_{x|y}$. The original density consisted of i.i.d. random variables which can be modeled by a first order Markov chain as a transition matrix with equal valued entries; i.e., $p_{x_i|y_j} = \frac{1}{8}$ for $i, j = 1, \cdots, 8$. The weight function becomes [4]

$$W(\cdot) = \frac{p(X_1^{(j)}, X_2^{(j)}, \cdots, X_n^{(j)})}{q(X_1^{(j)}, X_2^{(j)}, \cdots, X_n^{(j)})}$$

where

$$\frac{p(X_1^{(j)}, X_2^{(j)}, \cdots, X_n^{(j)})}{q(X_1^{(j)}, X_2^{(j)}, \cdots, X_n^{(j)})} = \frac{2^{-(m+1)}}{Q(X_1^{(j)})} \prod_{l=1}^{n-1} \frac{2^{-(m+1)}}{Q(X_{l+1}^{(j)}|X_l^{(j)})}$$

and the test statistic

$$S_n = \sum_{k=1}^{n} Z_k$$

where

$$Z_k = \sqrt{\frac{2}{n}}\left[1 + \sum_{j=1}^{m}(\frac{A_j}{A})\cos(\phi_j)[(1 - \frac{\tau_j}{T_c})U_k V_k^{(j)} + \frac{\tau_j}{T_c} U_k V_{k-1}^{(j)}]\right] + U_k \tilde{N}_k.$$

Where the values for U_k and $V_k^{(j)}$ will depend on the present state of $q_{x|y}$. That is in going from previous state to present state the value of $(U_k, V_k^{(1)}, V_k^{(2)})$ will correspond to the number of which the present state will be in. Since there are 8-states, each state will correspond to one of the 8-different combinations $(U_k, V_k^{(1)}, V_k^{(2)})$ can take. For example if the present state is in five then that corresponds to the sequence (1,-1,-1) that is $U_k = 1$, $V_k^{(1)} = -1$, $V_k^{(2)} = -1$ and so on. Table (4.1) shown in the next page shows the correspondence between the sequences and the states.

TABLE 4.1

Relationship between the states of $q_{x|y}$ and the sequences

Correspondence between sequences and states		
Sequence	U_k, V_k^1, V_k^2	state
(-1,-1,-1)	(-1,-1,-1)	1
(-1,-1,1)	(-1,-1,1)	2
(-1,1,-1)	(-1,1,-1)	3
(-1,1,1)	(-1,1,1)	4
(1,-1,-1)	(1,-1,-1)	5
(1,-1,1)	(1,-1,1)	6
(1,1,-1)	(1,1,-1)	7
(1,1,1)	(1,1,1)	8

Once the weight function is found one can establish the quick simulation estimate for the BER. The probability of error estimator is written as

$$\hat{P}_e = \frac{1}{K}\sum_{j=1}^{k} 1_{\{S_n^{(j)}<0\}} \frac{p(X_1^{(j)}, X_2^{(j)}, \cdots, X_n^{(j)})}{q(X_1^{(j)}, X_2^{(j)}, \cdots, X_n^{(j)})}$$

where

$$1_{\{S_n^{(j)}<0\}} = \begin{cases} 1 & \text{if } (x_1, \cdots, x_n) \in S_n < 0; \\ 0 & \text{if } (x_1, \cdots, x_n) \notin S_n < 0. \end{cases}$$

The quick simulation method is used to estimate the BER and the variance of the importance sampling estimator. Just like we did for the Monte Carlo estimation, a computer code is written to implement the test statistic and compute the probability of error and the variance. The code is written in such a way to make the simulation as realistic as possible; i.e., it is run for different values of τ_j and different values of ϕ_j, the results then being averaged. The code is run for different values of n and different values of SNR. Plots of the BER and the estimator variance are shown at the end of this chapter. The results for some values of SNR are also presented in the tables shown in the next few pages. These tables will be used for comparison purposes. The results obtained seem to fall within the bounds of the expected values of the BER [11], [12]. As in the monte carlo technique it is very difficult to determine how accurate the results obtained are. To our knowledge no one has analyzed the exact system using this technique. Better estimates would require more simulation trials for different values of τ_j and ϕ_j. Since these values play an important role in determining the BER and the variance, the reader is referred to the conclusion in Chapter Six for more details. The variance of the quick simulation method is several order of magnitude smaller than the variance of the monte carlo method, especially for high values of SNR and n. The decrease in the variance is expected based on the importance sampling theory. The decrease was not significant for small values of SNR and n, the reader is referred to the tables in the following two pages. Recall that the variance of the importance sampling estimator is expressed as follows

$$\mathrm{Var}[\hat{\mathrm{P}}_e] = \frac{1}{L}(\eta_{E_n}(q) - \mathrm{P}_e^2) \tag{4.4}$$

where

$$\eta_{E_n}(q) = \int_{E_n} \left[\frac{p}{q}(x_1, x_2, \cdots x_n)\right]^2 q(x_1, x_2, \cdots x_n) dx_1 dx_2 \cdots \, dx_n. \tag{4.6}$$

For conventional monte carlo technique $p(\cdot) = q(\cdot) \Rightarrow \eta_{E_n}(q) = \mathrm{P_e}$ so the variance of the monte carlo method becomes

$$\mathrm{Var}[\hat{\mathrm{P}}_\mathrm{e}] = \frac{1}{K}[\mathrm{P_e} - \mathrm{P_e}^2]$$

which is approximately $\mathrm{P_e}$. Hence, for the conventional monte carlo technique, the variance is expected to be of the same order of magnitude as the BER. The results obtained in Chapter Three confirm this except for the case of $SNR = 0dB$ and high values of n. In this case, we are not sure why the variance is so high. For other values of SNR and n, the variance is of the same order of magnitude as the BER. The variance of the quick simulation method however is always smaller than the BER as expected. A plot of the variance of the quick simulation vs the SNR for different values of n appears at the end of this chapter. We believe the results obtained are fairly accurate [11], [12] the following table shows a comparison of the quick simulation, and the method used in [12] to estimate the BER.

4.3 Estimation Results Using Quick Simulation Method

TABLE 4.2

Estimating BER using quick simulation and other method of esimation

Comparison between two methods for estimating BER n=31		
SNR	other method	quick simulation
4 dB	1.66×10^{-2}	1.63×10^{-2}
6 dB	4.77×10^{-3}	4.74×10^{-3}
8 dB	9.11×10^{-4}	8.98×10^{-4}
10 dB	1.21×10^{-4}	1.39×10^{-4}
12 dB	1.29×10^{-5}	3.52×10^{-5}
14 dB	4.49×10^{-7}	1.29×10^{-6}

For small values of SNR the two methods provide very close results. For higher values of SNR the two methods give slightly different results. We are not sure which method is more accurate. Another method for estimating the BER [11] gives results which fall within the bounds of the above values.

The following tables show the probability of bit error for different values of SNR and n.

TABLE 4.3

Estimates for the BER via quick simulation method for $n = 10$

Estimating the BER and the variance for n=10		
SNR	prob of error	estimator variance
0 dB	9.5104×10^{-2}	7.7829×10^{-2}
5 dB	1.8912×10^{-2}	1.3108×10^{-2}
10 dB	3.1728×10^{-3}	5.8110×10^{-4}
15 dB	7.5719×10^{-4}	3.2384×10^{-5}
20 dB	3.2446×10^{-4}	5.6715×10^{-6}

TABLE 4.4

Estimates for the BER via quick simulation method for $n = 20$

Estimating the BER and the variance for n=20		
SNR	prob of error	estimator variance
0 dB	8.6551×10^{-2}	7.5068×10^{-2}
5 dB	1.2076×10^{-2}	9.8395×10^{-3}
10 dB	4.9861×10^{-4}	1.0181×10^{-5}
15 dB	7.7343×10^{-5}	4.7901×10^{-7}
20 dB	8.9236×10^{-6}	6.2786×10^{-9}

TABLE 4.5

Estimates for the BER via quick simulation method for $n = 30$

Estimating the BER and the variance for n=30		
SNR	prob of error	estimator variance
0 dB	8.3276×10^{-2}	7.3968×10^{-2}
5 dB	9.8124×10^{-3}	8.3499×10^{-3}
10 dB	2.0431×10^{-4}	4.7541×10^{-4}
15 dB	3.0178×10^{-6}	3.8156×10^{-8}
20 dB	6.0126×10^{-7}	2.1768×10^{-10}

TABLE 4.6

Estimates for the BER via quick simulation method for $n = 40$

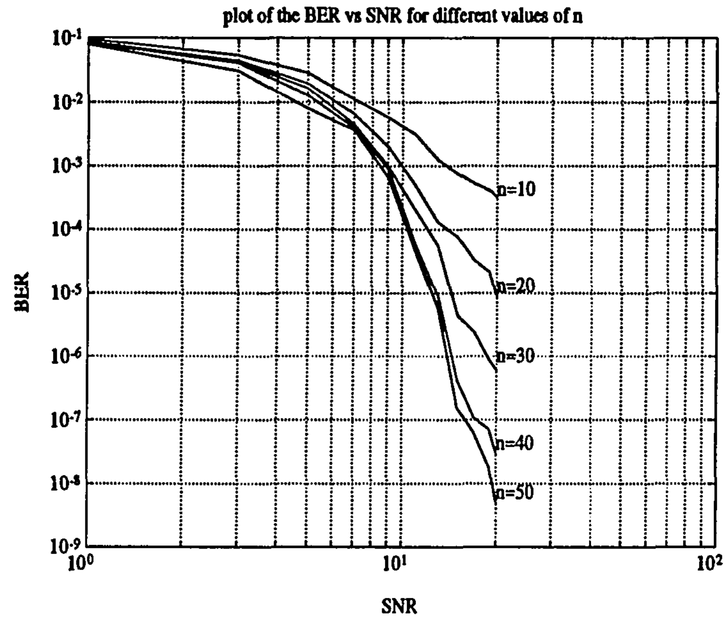
Estimating the BER and the variance for n=40		
SNR	prob of error	estimator variance
0 dB	8.2922×10^{-2}	3.9437×10^{-2}
5 dB	8.4951×10^{-3}	3.9437×10^{-4}
10 dB	4.9925×10^{-5}	4.4213×10^{-6}
15 dB	4.2182×10^{-7}	7.2291×10^{-11}
20 dB	3.0602×10^{-8}	3.1216×10^{-14}

TABLE 4.7

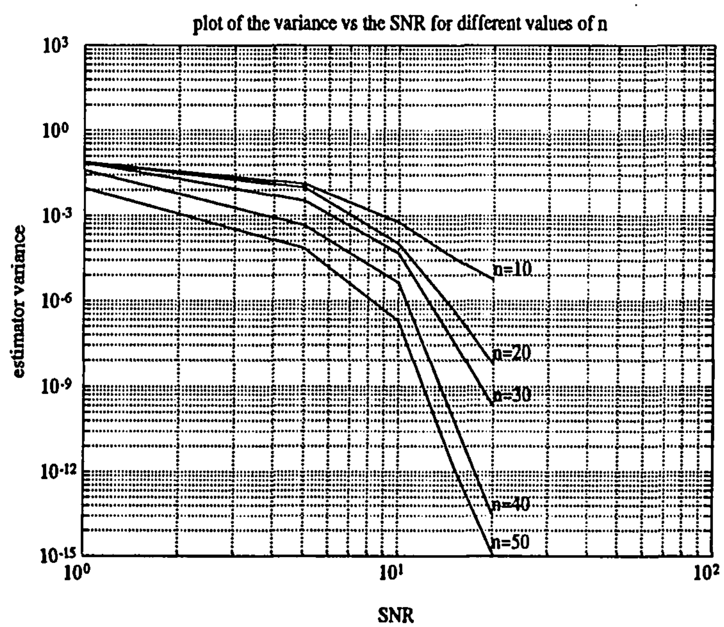
Estimates for the BER via quick simulation method for $n = 50$

Estimating the BER and the variance for n=50		
SNR	prob of error	estimator variance
0 dB	7.6143×10^{-2}	9.1438×10^{-3}
5 dB	7.8912×10^{-3}	7.1989×10^{-5}
10 dB	3.5167×10^{-5}	1.9437×10^{-7}
15 dB	1.5638×10^{-7}	1.6437×10^{-12}
20 dB	4.936×10^{-9}	1.3756×10^{-15}

Figure (4.1) shows a plot of the BER vs the SNR, and Figure (4.2) shows aplot of the variance vs the SNR :



Figure(4.1)



Figure(4.2)

CHAPTER 5

ESTIMATING BER VIA NUMERICAL INTEGRATION

In this chapter the third and final method for estimating the BER is presented. This method is based on numerical integration [5]. The approach is to find the probability density function (pdf) of the test statistic signal. Once the pdf is found one can use it to determine the probability of bit error, by integrating the pdf over the proper region in which a bit is decoded erroneously. In this chapter we show how this is done.

First recall that the probability of bit error is defined as the probability of receiving a binary "0" given that binary "1" is transmitted; i.e.,

$$\mathrm{P}\{S_n < 0 | \text{given that "1" is transmitted}\}$$

where

$$S_n = \sum_{k=1}^{n} Z_k$$

is the test statistic and

$$Z_k = AP \sum_{j=1}^{m} \frac{A_j}{A} \cos\phi_j \left[(1 - \frac{\tau_j}{T_c}) U_k V_k^j + \frac{\tau_j}{T_c} U_k V_{k-1}^j \right] + U_k N_k + AP.$$

Furthermore,

ϕ_j are i.i.d. uniform continuous random variables in $[0, 2\pi]$

$\frac{\tau_j}{T_c}$ are i.i.d. uniform continuous random variables in $[0, 1]$

N_k are i.i.d. Gaussian $N(0, \frac{N_o}{2} P)$,

U_k, V_k^j, and V_{k-1}^j are i.i.d. discrete random variables with values ± 1,

A is the amplitude of the baseband BPSK signal,

P is the power in the baseband spread spectrum signal $x(t)$.

The probability of bit error can be expressed as follows

$$P\{S_n < 0|\ \text{“1” is transmitted}\} = \int_{-\infty}^{\infty} \mathbf{Q}\left[\sqrt{2 \cdot SNR}(1+\frac{x}{n})\right] f_{\mathrm{X}}(x)\,dx \qquad (5.1)$$

where $\mathbf{Q}(x)$ represent the complementary error function, and $f_{\mathrm{X}}(x)$ is the pdf of the random function X defined as

$$\mathrm{X} = \sum_{j=1}^{m} \frac{A_j}{A} \cos\phi_j \left[(1-\frac{\tau_j}{T_c})R_o + \frac{\tau_j}{T_c}R_1\right],$$

where

$$R_o \triangleq \sum_{k=1}^{n} U_k V_k^j \quad \text{and} \quad R_1 \triangleq \sum_{k=1}^{n} U_k V_{k-1}^j.$$

The reader is referred to the appendix for a complete derivation of eq (5.1) [5]. As it stands, R_o and R_1 are discrete random numbers. Both R_o and R_1 are the sum of n products of discrete random variables taking the values ± 1. Hence, each product will take the values ± 1. Assume n is an even integer. Then, when we sum the products, both R_o and R_1 will take on any of the following possible values: $0, \pm 2, \pm 4, \cdots, \pm n$.

Based on eq(5.1) one can estimate the probability of bit error if $f_{\mathrm{X}}(x)$ is determined. To determine $f_{\mathrm{X}}(x)$ one proceeds as follows. From the law of total probability, $f_{\mathrm{X}}(x)$ can be written as

$$f_{\mathrm{X}}(x) = \sum_{q=-n}^{n} \sum_{l=-n}^{n} f_{\mathrm{X}}(x|R_o = l, R_1 = q) \cdot \mathrm{P}\{R_o = l, R_1 = q\} \qquad (5.2)$$

where we condition on R_o and R_1, and the double summation shows all possible combinations R_o and R_1 can have. $f_X(x)$ can be written as

$$f_X(x) = \sum_{q=-n}^{n} \sum_{l=-n}^{n} f_X(x|l,q) \cdot \mathrm{P}_{lq} \tag{5.3}$$

where $\mathrm{P}_{lq} \triangleq \mathrm{P}\{R_o = l, R_1 = q\}$,

$$\mathrm{P}_{lq} = \mathrm{P}\{\sum_{k=1}^{n} U_k V_k^j = l, \sum_{k=1}^{n} U_k V_{k-1}^j = q\} \tag{5.4}$$

where l and q can take any of the possible values listed previously. Using Bayes rule this can be written as

$$\mathrm{P}_{lq} = \sum_{\text{over all u's}} \mathrm{P}\{R_o = l, R_1 = q|\vec{U} = \vec{u}\} \cdot \mathrm{P}\{\vec{U} = \vec{u}\}. \tag{5.5}$$

The vector $\vec{U} = (u_1, u_2, \cdots, u_n)$ is an n-tuple random vector with i.i.d. random elements taking the values ± 1 with equal probabilities; i.e.,

$$\mathrm{P}\{u_i = 1\} = \mathrm{P}\{u_i = -1\} = \frac{1}{2}.$$

Since there are 2^n different possibilities the vector $\vec{U}$ can have, $\mathrm{P}\{\vec{U} = u\} = \frac{1}{2^n} = 2^{-n}$ so

$$\mathrm{P}_{lq} = 2^{-n} \sum_{\text{over all u's}} \mathrm{P}\{R_o = l, R_1 = q|\vec{U} = \vec{u}\}.$$

As discussed before, if n is an even integer then both R_o and R_1 can take any of the values shown before. That is, both l and q take on any of the values $0, \pm 2, \pm 4, \cdots, \pm n$. In order to find P_{lq} we need to exhaust all possible combinations of l and q when conditioned on the vector $\vec{U} = \vec{u}$, this is a combinatorics problem which can be solved using a counting technique. The table shown below gives all possible combinations l and q can have for any possible n-tuple $\vec{U}$.

The table (5.1) gives all possible combinations l and q for an n-tuple where n is an even integer

TABLE (5.1)

All possible combinations (l, q) can have for even integer n

$(-n, -n)$	$(-n, -n+2)$	$\cdots$	$(-n, n-2)$	$(-n, n)$
$(-n+2, -n)$	$(-n+2, -n+2)$	$\cdots$	$(-n+2, n-2)$	$(-n+2, n)$
$(-n+4, -n)$	$(-n+4, -n+2)$	$\cdots$	$(-n+4, n-2)$	$(-n+4, n)$
$(-n+6, -n)$	$(-n+6, -n+2)$	$\cdots$	$(-n+6, n-2)$	$(-n+6, n)$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$(n-6, -n)$	$(n-6, -n+2)$	$\cdots$	$(n-6, n-2)$	$(n-6, n)$
$(n-4, -n)$	$(n-4, -n+2)$	$\cdots$	$(n-4, n-2)$	$(n-4, n)$
$(n-2, -n)$	$(n-2, -n+2)$	$\cdots$	$(n-2, n-2)$	$(n-2, n)$
$(n, -n)$	$(n, -n+2)$	$\cdots$	$(n, n-2)$	(n, n)

From the table it is easy to see that there are a total of $(n+1)^2$ different combinations that l and q can have as pairs. In the next page we find all possible combinations l and q can have for a specific n.

For this part of the simulation, n is taken to be ten. A computer code was written to find all possible combinations l and q can have for that specific n. The code also counts the number of times each combination can occur. Table (5.2) gives all the possible combinations l and q can have for $n = 10$.

TABLE (5.2)

All possible combinations (l, q) can have for $n = 10$:

(10,8)	(10,-8)	(-10,8)	(-10,-8)	(8,10)	(8,-10)
(-8,10)	(-8,-10)	(8,6)	(8,-6)	(-8,6)	(-8,-6)
(6,8)	(6,-8)	(-6,8)	(-6,-8)	(10,4)	(10,-4)
(-10,4)	(-10,-4)	(4,10)	(4,-10)	(-4,10)	(-4,-10)
(8,2)	(8,-2)	(-8,2)	(-8,-2)	(2,8)	(2,-8)
(-2,8)	(-2,-8)	(6,4)	(6,-4)	(-6,4)	(-6,-4)
(4,6)	(4,-6)	(-4,6)	(-4,-6)	(4,2)	(4,-2)
(-4,2)	(-4,-2)	(2,4)	(2,-4)	(-2,4)	(-2,-4)
(10,0)	(-10,0)	(0,10)	(0,-10)	(6,0)	(-6,0)
(0,6)	(0,-6)	(2,0)	(-2,0)	(0,2)	(0,-2)

The probability P_{lq} is calculated as follows: l and q are the values which vectors R_o and R_1 can take respectively. Since these vectors are formed from n-tuple digits and every digit can take any of the values ± 1, there are 2^n ways to do that. Therefore the probability that the pair l and q will take on any of the possible combinations in the table is simply the number of times that specific combination occurs for a given $\vec{U} = \vec{u}$ divided by 2^n. Since there are 2^n different combinations for $\vec{U}$, the probability for any combination of the pair (l,q) can be expressed as follows

$$P_{lq} = \frac{\text{number of times (l,q) occurs}}{2^{2n}}. \tag{5.6}$$

Table (5.3) shown on the next page shows the number of times each combination occurs and the probability of occurrence; i.e., P_{lq}. Note that in counting the number of times each combination occurs, a certain pattern occurs. For every pair, all related combinations occur the same number of times. For example, the number of times the combination $(8,10)$ occurs is the same as the number of times the combination $(8,-10)$ occurs and likewise its the same as $(-8,10)$, $(-8,-10)$, $(10,8)$, $(10,-8)$, $(-10,8)$, $(-10,-8)$. A similar result is obtained for other combinations.

TABLE (5.3)

Actual combinations for $n = 10$ with times of occurance and the probability of occurance

list of possible combinations for (l,q) for $n = 10$		
combination	times of occurence	probability of occurrence
(10,8)	20	1.90735×10^{-5}
(8,6)	900	8.58307×10^{-4}
(6,4)	10800	9.15527×10^{-3}
(4,2)	50400	4.806519×10^{-2}
(10,4)	240	2.28882×10^{-4}
(8,2)	4200	4.00543×10^{-3}
(10,0)	504	4.806519×10^{-4}
(6,0)	22680	2.16293×10^{-2}
(2,0)	105840	1.009369×10^{-1}

The next step in determining $f_X(x)$ is to find $f_X(x|R_o = l, R_1 = q)$. A detailed derivation for finding the conditional density $f_X(x|R_o = l, R_1 = q)$ is presented in the appendix [5]. The result of this derivation is given by

$$f_X(x|R_o = l, R_1 = q) = \begin{cases} \frac{1}{|q-l|\pi C_j} \cdot \left[\ln \left| \frac{q}{l} \frac{C_j + \sqrt{C_j^2 - (x/q)^2}}{C_j + \sqrt{C_j^2 - (x/l)^2}} \right| \right], & \text{if } |x| \leq lC_j; \\ \frac{1}{|q-|\pi C_j} \left[\ln \left| \frac{C_j + \sqrt{C_j^2 - (x/q)^2}}{(x/q)} \right| \right], & \text{if } lC_j \leq |x| \leq qC_j; \\ 0 & \text{otherwise.} \end{cases}$$

Plots of the various individual probability density functions and the the convolution of these densities with themselves are shown in the appendix for different combinations of l and q. An example of these plots appears on the next page.

Figure (5.1) shows a plot of the density function for the combination (8, 10):

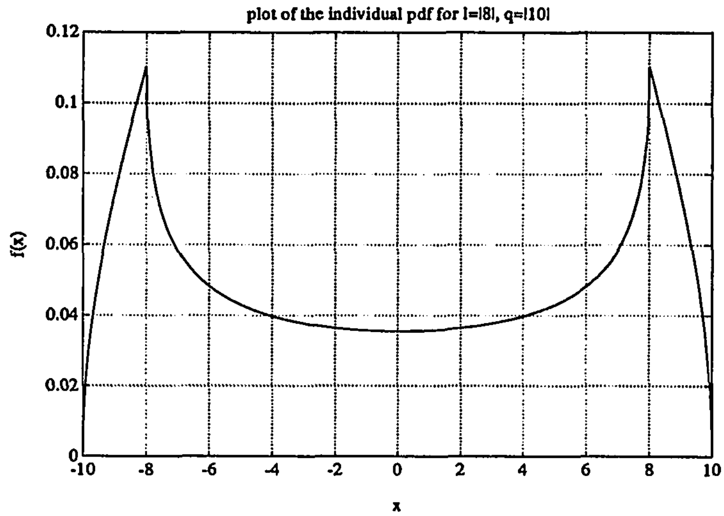


Figure (5.1)

Figure (5.2) shows a plot of the convolution of the density function with itself:

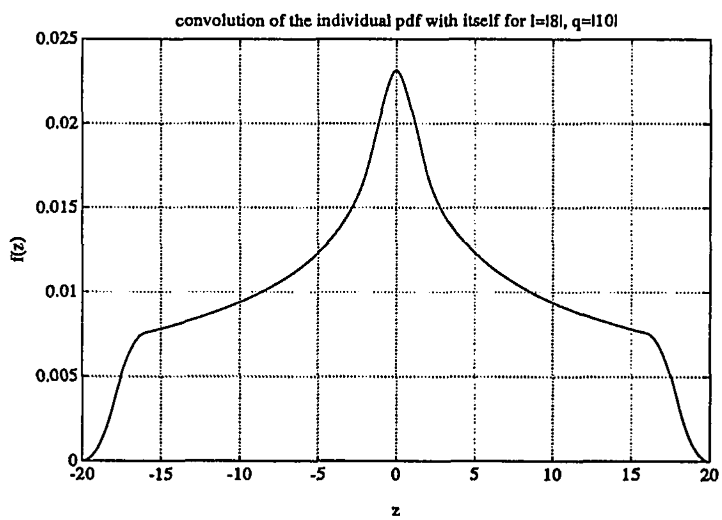


Figure (5.2)

When the conditional density $f_X(x|l,q)$ is determined, from eq(5.2) one can find the density $f_X(x)$ as follows

$$f_X(x) = \sum_{q=-n}^{n} \sum_{l=-n}^{n} f_X(x|R_o = l, R_1 = q) \cdot \mathrm{P}\{R_o = l, R_1 = q\} \qquad (5.2)$$

Figure (5.3) shown below present a plot of the probability density function $f_X(x)$.

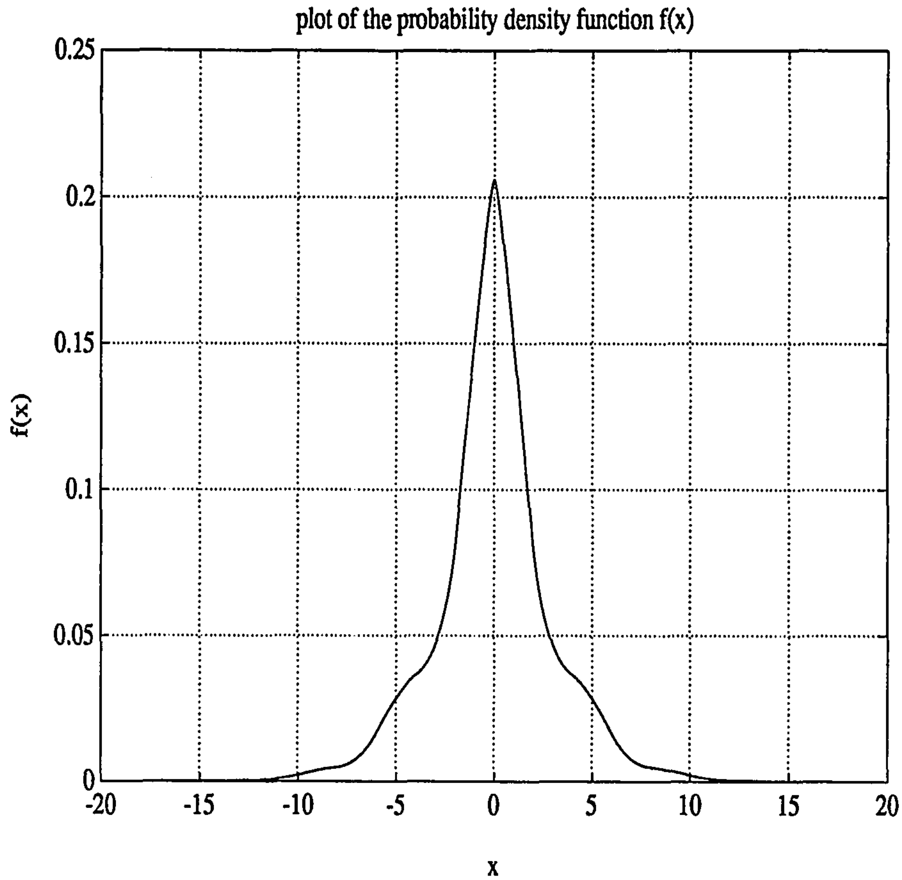


Figure (5.3)

The table (5.4) shows the BER for different SNR's with $n = 10$:

TABLE (5.4)

Estimating the BER using numerical integration for $n = 10$

SNR	BER
0 dB	9.531×10^{-2}
5 dB	2.265×10^{-2}
10 dB	6.552×10^{-3}
15 dB	3.795×10^{-3}
20 dB	3.103×10^{-3}

Note that not too many data points are taken. The results obtained are for $n = 10$. Only this value is used due to the tremendous amount of computation involved. The reader is referred to the conclusion in Chapter Six for the pros and cons of this method.

CHAPTER 6

CONCLUSION

In this thesis we estimated the BER of a direct sequence multiple access spread spectrum communication system. In particular we tried to develop an efficient and reliable method to achieve our objective. This is not an easy task, because digital communication systems are quite complex requiring stochastic methods for analysis. Therefore, except for simple cases, it is very difficult to develop a closed form error expression. Therefore one usually resorts to numerical methods; i.e., simulation to approximate the probability of error and other statistics of interest. In this thesis we relied on simulation techniques to estimate the BER. Basically we used three methods for estimating the BER. Two of these methods involved simulation, while the other method is based on numerical integration. Each method has its own advantages and disadvantages. We will now compare these methods and show which method is the most efficient and reliable one.

The first method we used to estimate the BER is the conventional Monte Carlo method. The advantage of this method is the simplicity of its concept and implementation. Basically in this method we simulated the output of the spread spectrum communication system in a computer code. Recall that

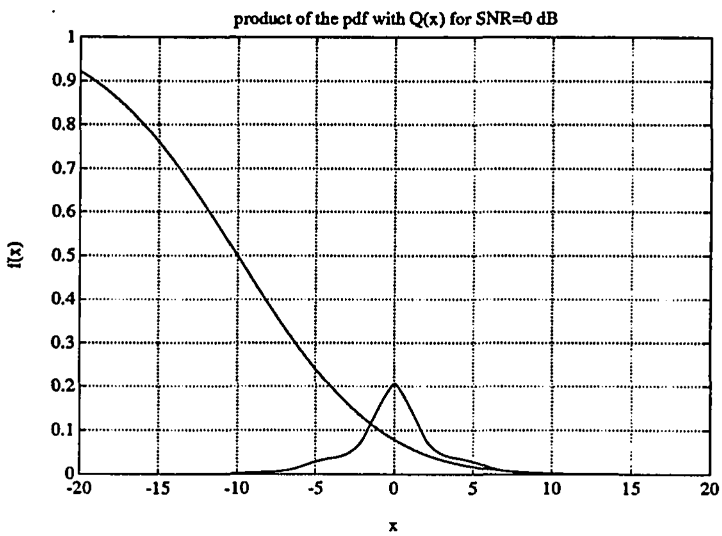
$$S_n = \sum_{k=1}^{n} AP \sum_{j=1}^{m} \frac{A_j}{A} \cos\phi_j \left[(1 - \frac{\tau_j}{T_c}) U_k V_k^j + \frac{\tau_j}{T_c} U_k V_{kj-1} \right] + U_k N_k + AP.$$

Where ϕ_j, $\frac{\tau_j}{T_c}$,U_k,V_k^j and V_{kj-1} are all random quantities. These random parameters are simulated using computer random number generators. The simulation is ran for different values of these random parameters and the number of times a bit is decoded as an error is counted and the result is averaged. Unfortunately, the

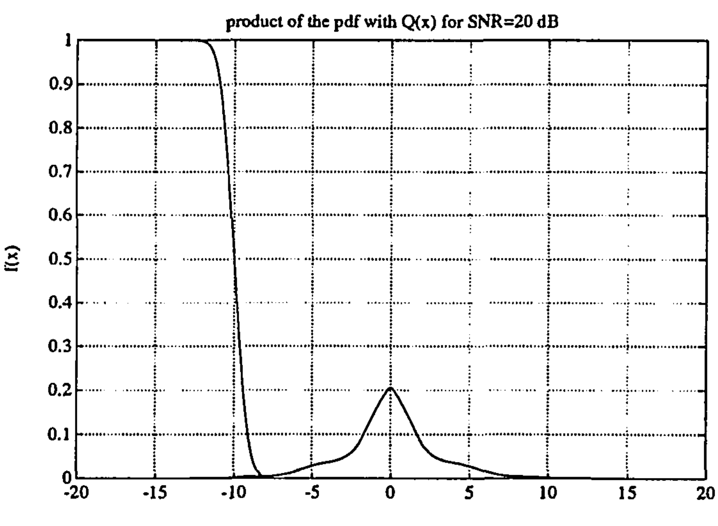
Monte Carlo method suffers from a major drawback in that a very large number of simulation runs are required before an acceptable relative precision in estimation is obtained which takes a very long time on the computer. One of the major problems we had with this method is the tremendous amount of computations needed to obtain a good estimate for the BER when the SNR is relatively large and n is relatively moderate. For example, we had to use 5 million runs for every random value of τ and ϕ when the SNR = 20 dB and $n = 40$. These values are not very big for a typical spread spectrum system. So one can imagine the extremely large amount of time and computation required to achieve relative accuracy for high values of n and SNR.

The second method is the quick simulation method developed from the notion of importance sampling and the principles of LDT. The advantage of the quick simulation method is that a much smaller amount of time and computations are required to estimate the BER with good precision, making this technique considerably more efficient. The disadvantages of the quick simulation method is the different steps required to obtain the most efficient distribution or density function which maximizes the rate of decrease in the estimator variance. Once this is obtained, the density is substituted in the weight function formula to estimate the probability of error. In our case, due to the modeling of our system as a Markov chain, the new density is implemented in the form of a transition matrix. The process of obtaining the matrix which maximizes the rate function is little bit involved especially when the matrix is large. Recall that the size of the matrix is proportional to 2^{m+1} where m is the number of other interfering spread spectrum signals in the multiple access system. So it becomes more difficult to obtain the optimum transition matrix which maximizes the rate function as the number of users increases. Another disadvantage with this method is that more simulation runs are required as n and the signal-to-noise ratio increase. For example, we used 40,000 trials to estimate the BER for SNR = 20 dB and $n = 40$ while we used 10,000 trials the BER for SNR = 20 dB and $n = 10$.

The third method used in this thesis for evaluating the BER is based on numerical integration. The concept of this method is simple and is developed from fundamental concepts of probability theory. Basically, the pdf of the output S_n, is determined and integrated in the proper region corresponding to a bit error, thus giving an estimate of the BER. The advantage of this method is that it is not required to repeat the process of computation as we did in the previous two methods. This is because the pdf is a deterministic function, so once it is found one would integrate it once to obtain the desired result. However this method for estimating the BER is the least attractive of the three methods for several reasons. First the method involves a lot of mathematical analysis and computations. Except for simple cases; i.e., small n (e.g., $n \leq 10$) and small m (e.g., $m = 2$) it is very difficult to obtain a closed form expression for the output S_n. Recall that in finding the probability of occurrence for the (l, q) combinations we had to go through 2^{2n} possible combinations (see the denominator for the expression of P_{lq}). So for $n = 20$, one needs 2^{40} computations to get the result. Clearly this is undesirable, especially as n increases. Also, in finding the density $f_X(x)$ one must convolve the densities of the interfering signals. So, the higher the number of interfering signals the more troublesome it is to convolve them. Second, except for small SNR (e.g., $SNR < 5dB$) this method does not provide a good estimate for the BER. The reason for this can be seen from the two graphs shown on the next page.



Figure(6.1)



Figure(6.2)

Figure (6.1) shows the product of the density $f_X(x)$ with $Q(x)$ for SNR=0 dB and Figure (6.2) shows the product of the two for SNR=20 dB. In the case of 0 dB the $Q(x)$ overlap all over the probability density function and the product is non-zero all over the density region. For the case of 20 dB the product of

$Q(x)$ with the pdf is non-zero only for a small part of the density region and it is approximately zero for most of the density region. This is due to the sharp cutoff in the $Q(x)$ function, since the variable x is a function of the SNR (see the expression for the $Q(\cdot)$ in Chapter 5). So, one can see that for higher SNR values, $Q(x)$ will have sharper cutoff and the product will be almost zero for most of the overlapping area between the pdf and $Q(x)$. Therefore one concludes that for moderate to high values of SNR; (i.e., $SNR \geq 5$ dB) this method is not reliable in estimating the BER. However for small values of SNR, this method gives good results especially in simple cases; i.e., small n and m. Table (6.1) shows the results of estimating the BER using the three methods for small values of SNR's, while table (6.2) shows the results of estimation for high values of SNR's:

TABLE 6.1

Comparison between the three methods for small values of SNR

Estimating the BER using the three methods for n=10			
SNR	quick simulation	monte carlo	numerical integration
0 dB	9.51×10^{-2}	9.53×10^{-2}	9.53×10^{-2}
1 dB	7.34×10^{-2}	7.29×10^{-2}	7.54×10^{-2}
2 dB	5.43×10^{-2}	5.41×10^{-2}	5.78×10^{-2}
3 dB	3.92×10^{-2}	3.93×10^{-2}	4.34×10^{-2}
4 dB	2.95×10^{-2}	2.84×10^{-2}	3.20×10^{-2}

TABLE 6.2

Comparison between the three methods for high values of SNR

Estimating the BER using the three methods for n=10			
SNR	quick simulation	monte carlo	numerical integration
0 dB	9.51×10^{-2}	9.22×10^{-2}	9.53×10^{-2}
5 dB	1.89×10^{-2}	1.91×10^{-2}	2.27×10^{-2}
10 dB	3.18×10^{-3}	3.53×10^{-3}	6.55×10^{-3}
15 dB	7.57×10^{-4}	8.6×10^{-4}	3.81×10^{-3}
20 dB	3.24×10^{-4}	2.32×10^{-4}	3.1×10^{-3}

There are several factors that control the value of the BER for the direct sequence spread spectrum communication system. Based on the results obtained, one observes the following. First, the BER is inversely proportional to the SNR; i.e., the higher the SNR the smaller the BER. Second, n which is the ratio of the chip rate T_c and the information bit T_b, is also inversely proportional to the BER, so for high values of n the BER is small. Third, the choice of the phase shift ϕ_j and the time shift τ_j effect the test statistic. Depending on the values of these random variables, one can increase or decrease the number of hits.

The bulk of this thesis concentrates on the quick simulation method. We believe that this method is the most attractive method of the three. In this technique we were able to obtain a good estimate (see [11] and [12]) for the BER with a fairly small number of simulation runs compared to the Monte Carlo method. There was an increase in the occurrence of event of interest; i.e., $S_n \leq 0$, compared to the Monte Carlo method. This is due to the change in the density function of the input random variables. The new biased density is expressed as a first-order Markov chain transition density matrix $q(\cdot|\cdot)$ and steady state distribution $Q(\cdot)$. This transition density is generated using the principles of large deviation theory.

The increase in the number of hits is due to this twisted transition matrix which creates more hits by increasing the transitional probabilities for the event of interest. That makes these events occur more frequently. The true estimate for the BER is obtained by the use of the weight function which is the ratio of the original density to the biased density. The factors discussed above which effect the BER play an important role in generating the new transition matrix. As a matter of fact we noticed that the matrix is extremely sensitive to the slightest change in the parameters above; i.e., SNR, n, ϕ_j and τ_j.

An example to demestrate this consider the following results for $n = 20$ and $SNR = 5dB$. Some of the results we obtained for the BER and the variance are the following. For $\phi_1 = 57.37°$, $\phi_2 = 219.928°$, $\tau_1 = 0.59197$, and $\tau_2 = .328153$, the $BER = 9.751 \times 10^{-}3$ and the Var $= 8.667 \times 10^{-}3$. For other values of ϕ and τ (like $\phi_1 = 45.61°$, $\phi_2 = 97.98°$, $\tau_1 = 0.4573$, $\tau_2 = 0.3037$) the $BER = 1.099 \times 10^{-2}$, and the Var $= 7.447 \times 10^{-3}$.

Consider the case where $n = 40$ and $SNR = 20dB$. For $\phi_1 = 201.56°$, $\phi_2 = 176.84°$, $\tau_1 = 0.1223$, and $\tau_2 = 0.84179$, the $BER = 2.675 \times 10^{-7}$ the Var $= 1.3066 \times 10^{-11}$. For $\phi_1 = 72.14°$, $\phi_2 = 28.42°$, $\tau_1 = 0.7804$, and $\tau_2 = 0.3237$, the $BER = 2.841 \times 10^{-15}$ and the Var $= 7.095 \times 10^{-26}$.

The above results show the variation in the values of the BER and the variance for different values of the random parameters, SNR, and n. The above parameters control the transitional probabilities for the Markov chain. That makes certain states occur more frequently than others. For our problem, the states that occur more often are the ones which will produce a bit error; i.e., $(S_n < 0)$. Furthermore, we obtained different results when using double precision versus single precision. The variance of the quick simulation is compared to the variance of the Monte Carlo technique. In the Monte Carlo method, the variance was of the same order of magnitude as the BER, which is what is expected [4]. In the quick simulations the variance is expected to decrease with an exponential rate as the number of simulation trials increases. For this particular problem, the ratio of the chip rate to the information bit n controls this exponential rate of decrease. For this system we noticed that the rate of decrease depends on the value of the SNR. For small values of SNR; i.e., $SNR \leq 10dB$, the variance of the quick simulation estimator was of the same order of magnitude as the the BER. For higher values of SNR, the variance starts to decrease rapidly as SNR and n both increase.

In general, what determines how fast the variance of the estimator decreases can vary from one system to another. For this particular system, it is n and the SNR that determine the rate of decrease. As a matter of fact, there are some communication systems for which the quick simulation method cannot apply [4]. We believe the results obtained are reasonable and fall within the expected range for the BER of the system studied . A final remark concerning the number of simulation runs is that more simulation trials improves the estimate is going to be, especially for high values of n and SNR. The question is, how can we get an acceptable level of accuracy and precision in the evaluation of the BER with minimum number of simulation trials. We believe that the quick simulation technique provides a good answer to this question. That, is it gives good results with fairly small number of simulation trials.

APPENDIX A: DERIVATIONS

In this section a complete derivation for obtaining a closed form expression for the output of the direct sequence spread spectrum communication system is presented [5]. Recall that the probability of bit error is defined as

$$P\{S_n < 0|\text{given that "1" is transmitted}\}$$

where

$$S_n = \sum_{k=1}^{n} Z_k$$

is the test statistic and

$$Z_k = AP\sum_{j=1}^{m} \frac{A_j}{A}\cos\phi_j[(1-\frac{\tau_j}{T_c})U_k V_k^j + \frac{\tau_j}{T_c}U_k V_{k-1}^j] + U_k N_k + AP.$$

Furthermore,

ϕ_j are i.i.d. uniform continuous random variables between $[0, 2\pi]$

$\frac{\tau_j}{T_c}$ are i.i.d. uniform continuous random variables between $[0, 1]$

N_k are i.i.d. Gaussian $N(0, \frac{N_o}{2}P)$

U_k, V_k^j, and V_{k-1}^j are i.i.d. discrete random variables with values ± 1

$\frac{A_j}{A}$ is the ratio of the amplitude of the desired signal to the amplitude of the j^{th} interfering signal Let

$$X_k^j = \frac{A_j}{A}\cos\phi_j[(1-\frac{\tau_j}{T_c})U_k V_k^j + \frac{\tau_j}{T_c}U_k V_{k-1}^j] \tag{A.1}$$

then S_n can be written as

$$S_n = AP\sum_{k=1}^{n}\sum_{j=1}^{m} X_k^j + \sum_{k=1}^{n} U_k N_k + \sum_{k=1}^{n} AP. \tag{A.2}$$

Now examine every term in S_n individually. First consider

$$\sum_{k=1}^{n} AP$$

which is just a summation of a constant, and can be written as

$$\sum_{k=1}^{n} AP = nAP. \tag{A.3}$$

The second term examined in S_n is $\sum_{k=1}^{n} U_k N_k$ which is a product of two i.i.d. random variables. Since U_k takes on the values ± 1 with equal probabilities, and N_k is a Gaussian random variable with zero mean and variance $\frac{N_o P}{2}$ it is easy to show that the sum of their product would be also Gaussian with zero mean and variance $= \frac{N_o}{2} nP$, therefore

$$\sum_{k=1}^{n} U_k N_k \quad \text{is distributed as} \quad N(0, \frac{N_o}{2} nP).$$

For convenience define

$$\mathrm{N} \triangleq \sum_{k=1}^{n} U_k N_k. \tag{A.4}$$

The last term in S_n is

$$AP\sum_{k=1}^{n}\sum_{j=1}^{m} X_k^j.$$

Define

$$\mathrm{X} \triangleq \sum_{j=1}^{m}\sum_{k=1}^{n} X_k^j \tag{A.5}$$

so S_n can be written as $S_n = AP\mathrm{X} + \mathrm{N} + nAP$ where

$$\mathrm{X} = \sum_{j=1}^{m}\sum_{k=1}^{n}[\frac{A_j}{A}\cos\phi_j((1-\frac{\tau_j}{T_c})U_k V_k^j + \frac{\tau_j}{T_c}U_k V_{k-1}^j)]$$

$$X = \sum_{j=1}^{m} \frac{A_j}{A}(1 - \frac{\tau_j}{T_c})\cos\phi_j \sum_{k=1}^{n} U_k V_k^j + \sum_{j=1}^{m} \frac{A_j}{A}\frac{\tau_j}{T_c}\cos\phi_j \sum_{k=1}^{n} U_k V_{k-1}^j.$$

Define the following random vectors

$$\vec{U} = (U_1, U_2, \cdots, U_n)$$

$$\vec{V^j} = (V_1^j, V_2^j, \cdots, V_n^j)$$

and

$$\vec{V_o^j} = (V_o^j, V_1^j, \cdots, V_{n-1}^j).$$

Let

$$R_o \triangleq \sum_{k=1}^{n} U_k V_k^j = \vec{U} \cdot \vec{V^j}$$

and

$$R_1 \triangleq \sum_{k=1}^{n} U_k V_{k-1}^j = \vec{U} \cdot \vec{V_o^j},$$

then X can be simplified by writing it as

$$X = \sum_{j=1}^{m} \frac{A_j}{A} R_o \cos\phi_j (1 - \frac{\tau_j}{T_c}) + \sum_{j=1}^{m} \frac{A_j}{A} R_1 \cos\phi_j \frac{\tau_j}{T_c}. \qquad (A.6)$$

Recall that an error occurs whenever $S_n < 0$ given that "1" is transmitted, that is

$$P\{S_n < 0\} = P\{[APX + N + nAP] < 0\}. \qquad (A.7)$$

From the total probability theorem this can be written as

$$P\{S_n < 0\} = \int_{-\infty}^{+\infty} P\{[APX + N + nAP] < 0 | X = x\} f_X(x)\, dx \qquad (A.8)$$

where $f_X(\cdot)$ is the pdf of X.. Note that

$$P\{APX + N + nAP < 0 | X = x\} = P\{[APx + N + nAP] < 0\}.$$

The random function X takes on a specific constant value x under the conditional probability, hence $\mathrm{N} + AP\mathrm{x} + APn$ is a Gaussian random variable shifted by the constant $APx + APn$.

We have

$$E[\mathrm{N} + APx + APn] = APx + APn$$

and

$$\mathrm{Var}(\mathrm{N} + APx + APn) = E[(\mathrm{N} + AP(\mathrm{x} + n))^2] - E^2[\mathrm{N} + AP(\mathrm{x} + n)].$$

Let $y = AP(x + n)$ then

$$\mathrm{Var}(\mathrm{N} + y) = E[(\mathrm{N} + y)^2] - E^2[\mathrm{N} + y]$$

$$E[(\mathrm{N} + y)^2] = E[\mathrm{N}^2] + E[2y\mathrm{N}] + E[y^2]$$

but

$$E[2y\mathrm{N}] = 2yE[\mathrm{N}] = 0 \text{ and } E^2[\mathrm{N} + y] = E^2[y]$$

hence

$$\mathrm{Var}(\mathrm{N} + AP(x + n)) = E[\mathrm{N}^2] + E[y^2] - E^2[y]$$

and since $E[y^2] = E^2[y]$ then

$$\mathrm{Var}[\mathrm{N} + AP(x + n)] = E[\mathrm{N}^2] = \frac{N_o}{2}nP.$$

Therefore S_n is a Gaussian random variable with mean and variance as shown

$$S_n \quad \text{is distributed as} \quad \mathrm{N}(AP(x + n), \frac{N_o}{2}nP), \qquad (A.9)$$

and hence

$$P\{\mathrm{N} + AP(x + n) < 0\} = P\{\mathrm{N}(AP(x + n), \frac{N_o}{2}nP) < 0\}. \qquad (A.10)$$

This probability can be easily evaluated as follows

$$P\{\mathrm{N}(AP(x+n), \frac{N_o}{2}nP) < 0\} = \int_{-\infty}^{0} \frac{1}{\sqrt{2\pi}\sqrt{\frac{N_o}{2}nP}} \exp[-\frac{(\xi - AP(x+n))^2}{2\frac{N_o}{2}nP}]\, d\xi.$$

Letting $u = \frac{\xi - AP(x+n)}{\sqrt{\frac{N_o}{2}nP}}$, a change of variables yields

$$\int_{-\infty}^{\frac{-AP(x+n)}{\sqrt{\frac{N_o}{2}nP}}} \frac{1}{\sqrt{2\pi}} \exp[-\frac{u^2}{2}]\, du$$

or

$$P\{\mathrm{N}(AP(x+n), \frac{N_o}{2}nP) < 0\} = \int_{\frac{AP(x+n)}{\sqrt{\frac{N_o}{2}np}}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp[-\frac{u^2}{2}]\, du. \tag{A.11}$$

Define the complementary error function as follows

$$\mathbf{Q}(x) \triangleq \int_{x}^{\infty} \frac{1}{\sqrt{2\pi}} \exp[-\frac{x^2}{2}]\, dx \tag{A.12}$$

and thus

$$P\{\mathrm{N}(AP(x+n), \frac{N_o}{2}nP) < 0\} = \mathbf{Q}\left(\frac{AP}{\sqrt{\frac{N_o}{2}nP}}(x+n)\right)$$

$$P\{\mathrm{N}(AP(x+n), \frac{N_o}{2}nP) < 0\} = \mathbf{Q}\left(\frac{APn}{\sqrt{\frac{N_o}{2}nP}}(1+\frac{x}{n})\right). \tag{A.13}$$

Since $E_b = nA^2P$ and $\frac{E_b}{N_o} = SNR$, then

$$\frac{APn}{\sqrt{\frac{N_o}{2}nP}} = \sqrt{\frac{2A^2P^2n^2}{N_oPn}} = \sqrt{\frac{2E_b}{N_o}} = \sqrt{2\cdot SNR} \tag{A.14}$$

therefore

$$P\{\mathrm{N}\big(AP(x+n), \frac{N_o}{2}nP\big) < 0\} = \mathbf{Q}\left(\sqrt{2\cdot SNR}(1+\frac{x}{n})\right). \tag{A.15}$$

Substituting the above into the probability of error expression we have

$$P\{S_n < 0|\text{given “1” is transmitted}\} = \int_{-\infty}^{\infty} \mathbf{Q}\left[\sqrt{2\cdot SNR}(1+\frac{x}{n})\right] f_X(x)\, dx.$$

The second derivation we present in this appendix is the formula for the conditional probability density function $f_X(x|l,q)$ this can be found as follows [5]. The output of the direct sequence spread spectrum communication system receiver is expressed as

$$S_n = \sum_{k=1}^{n} AP \sum_{j=1}^{m} \frac{A_j}{A} \cos\phi_j [(1 - \frac{\tau_j}{T_c}) U_k V_k^j + \frac{\tau_j}{T_c} U_k V_{k-1}^j] + U_k N_k + AP.$$

As shown before in this chapter this is simplified to

$$S_n = AP\mathrm{X} + \mathrm{N} + nAP \qquad (A.16)$$

where nAP is constant, N is Gaussian distributed random variable, and X is a random variable which represent the interfering signals see eq (A.6). Note that all random components in S_n are independent, furthermore the interference signals in X are also independent, therefore the density $f_{\mathrm{X}}(x|l,q)$ can be found by convolving the density function of all interfering spread spectrum signals [16]. To do this, first one determines the density function for every individual interfering spread spectrum signal; this is done as follows. Let

$$\mathrm{X}^j{}_{l,q} = \frac{A_j}{A} \cos\phi_j [(1 - \frac{\tau_j}{T_c}) l + \frac{\tau_j}{T_c} q]$$

represent the j^{th} interference conditioned on certain l and q where ϕ_j is a continuous uniform random variable on [0,2π] and $\frac{\tau_j}{T_c}$ is a continuous uniform random variable on [0,1].

Let $\frac{A_j}{A} = C_j$ where C_j is a constant greater than zero. For mathematical convenience we rewrite the random function $\mathrm{X}_{l,q}$ as a product of two random functions as shown

$$\mathrm{Z} = \mathrm{W} \cdot \mathrm{Y} \qquad (A.17)$$

where

$$W = C_j \cos\phi_j \quad \text{and} \quad Y = l + (q-l)\frac{\tau_j}{T_c}.$$

To find the density of Z first we find the density of W and Y. The distribution of W is

$$F_W(w) = P\{W \leq w\} = P\{C_j \cos\phi_j \leq w\} = P\{\phi_j \leq \cos^{-1}(\frac{w}{C_j})\}$$

hence

$$F_W(w) = 2\int_{\cos^{-1}(\frac{w}{C_j})}^{\pi} \frac{1}{2\pi} d\theta = 1 - \frac{1}{\pi}\cos^{-1}(\frac{w}{C_j})$$

and thus the density $f_W(w)$ is

$$f_W(w) = \frac{1}{C_j \pi} \frac{1}{\sqrt{1 - (\frac{w}{C_j})^2}}$$

or

$$f_W(w) = \frac{1}{\pi} \frac{1}{\sqrt{C_j^2 - w^2}} \tag{A.18}$$

this density is valid for $|w| \leq C_j$ and it is zero for $|w| > C_j$. To compute the density of $Y = l + (q-l)\frac{\tau_j}{T_c}$ we have

$$F_Y(y) = P\{Y \leq y\} = P\{l + (q-l)\frac{\tau_j}{T_c} \leq y\} = P\{\frac{\tau_j}{T_c} \leq \frac{y-l}{q-l}\}$$

$$F_Y(y) = \int_0^{\frac{y-l}{q-l}} 1 \cdot dx = \frac{y-l}{q-l}$$

so the density is

$$f_Y(y) = \frac{1}{q-l}. \tag{A.19}$$

This density is valid on $[l, q]$ where $q > l$ and $q \neq l$ and is zero otherwise.

Since $Z = W \cdot Y$, use total probability theorem to conclude

$$F_Z(z) = P\{WY \leq z\} = \int_{-\infty}^{\infty} P\{WY \leq z | Y = y\} f_Y(y) dy \tag{A.20}$$

$$F_Z(z) = \int_{-\infty}^{\infty} P\{\mathrm{Wy} \leq z | \mathrm{Y} = \mathrm{y}\} f_Y(y) dy = \int_{-\infty}^{\infty} P\{\mathrm{W} \leq \frac{z}{y}\} f_Y(y) dy.$$

Therefore

$$F_Z(z) = \int_{-\infty}^{\infty} F_W(\frac{z}{y}) \cdot f_Y(y) dy.$$

By differentiating $F_Z(z)$ we get the density

$$f_Z(z) = \int_{-\infty}^{\infty} \frac{1}{|y|} f_W(\frac{z}{y}) f_Y(y) dy. \quad (A.21)$$

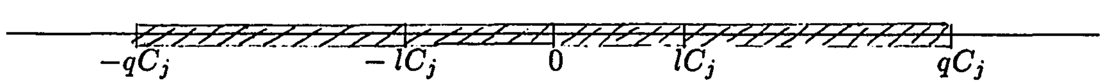
or

$$f_Z(z) = \frac{1}{(q-l)} \int_l^q \frac{1}{|y|} f_W(\frac{z}{y}) dy. \quad (A.22)$$

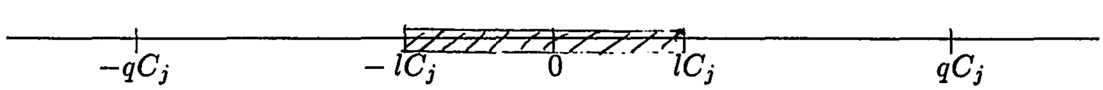
This density has several shapes depending on the values of l and q [5] as shell be presented. In analysing these conditions for convenience we will make the following assumption. We will let q to be equal to the largest of the two integers in magnitude, so whenever appropriate one can swap l and q to make it compatible with this assumption. First consider the case where $0 < l < q$ then

$$f_Z(z) = \frac{1}{q-l} \cdot \int_l^q \frac{1}{\pi y} \cdot \frac{dy}{\sqrt{C_j^2 - (z/y)^2}}$$

where $|z| \leq C_j y$, so z can take any value on the line shown below.



Examine the case where $|z| \leq lC_j$ as shown in the line below

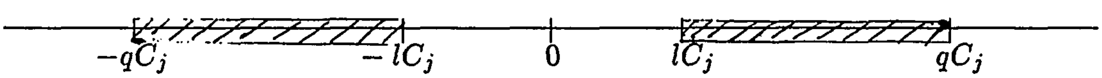


Let $u = \frac{z}{y}$ then $dy = \frac{-y^2}{z} du$. By substituting the result into the density for z we have

$$f_Z(z) = \frac{-1}{\pi(q-l)} \int_{z/l}^{z/q} \frac{1}{u\sqrt{C_j^2 - u^2}} \cdot du$$

$$f_Z(z) = \frac{1}{\pi(q-l)C_j} \cdot \left[\ln \left| \frac{q}{l} \frac{C_j + \sqrt{C_j^2 - (z/q)^2}}{C_j + \sqrt{C_j^2 - (z/l)^2}} \right| \right]$$

This density is only valid for $|z| \le lC_j$. Next examine the case where $lC_j < |z| < qC_j$ or $l < \left| \frac{z}{C_j} \right| < q$ as shown below:



Doing the same change of variables we obtain

$$f_Z(z) = \frac{-1}{(q-l)\pi} \int_{C_j}^{z/q} \frac{1}{u\sqrt{C_j^2 - u^2}} du = \frac{1}{(q-l)\pi} \int_{z/q}^{C_j} \frac{1}{u\sqrt{C_j^2 - u^2}} du.$$

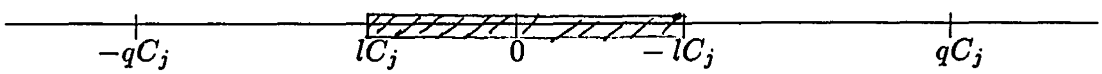
Simplifying the result we get

$$f_Z(z) = \frac{1}{(q-l)\pi C_j} \left[\ln \left| \frac{C_j + \sqrt{C_j^2 - (z/q)^2}}{(z/q)} \right| \right].$$

This result is valid only for $lC_j \le |z| \le qC_j$. The density is zero elsewhere. One can show that the same result can be obtained for negative values of l and q that is $l < 0$ and $q < 0$.

Now consider the case when one of the integers l or q is negative and the other is positive. For simplicity we will consider l to be the negative integer and q to be the positive integer i.e., $l < 0 < q$. The analysis is presented in the next page

If $0 < z < -lC_j$ then as shown in the line below $0 < \dfrac{z}{C_j} < -l$ *or* $l < \dfrac{-z}{C_j} < 0$



Substitute the result into the density $f_Z(z)$ expression we get

$$f_Z(z) = \frac{1}{(q-l)\pi}\int_{l}^{\frac{-z}{C_j}} \frac{1}{y}\cdot\frac{dy}{\sqrt{C_j^2-(z/y)^2}} + \frac{1}{(q-l)\pi}\int_{\frac{z}{C_j}}^{q} \frac{1}{y}\cdot\frac{dy}{\sqrt{C_j^2-(z/y)^2}}$$

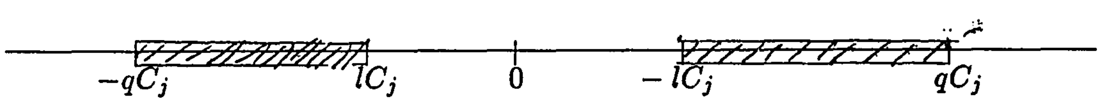
doing a change of variables by Letting $u = \frac{z}{y}$ we have

$$f_Z(z) = \frac{1}{(q-l)\pi}\left[\int_{-C_j}^{\frac{z}{l}} \frac{1}{u}\frac{du}{\sqrt{C_j^2-u^2}} + \int_{\frac{z}{q}}^{C_j} \frac{1}{u}\frac{du}{\sqrt{C_j^2-u^2}}\right]$$

simplifying the result we get

$$f_Z(z) = \frac{1}{\pi(q-l)C_j}\cdot\left[\ln\left|\frac{q}{l}\frac{C_j+\sqrt{C_j^2-(z/q)^2}}{C_j+\sqrt{C_j^2-(z/l)^2}}\right|\right]$$

The same result is obtained for negative values of z. Consider the case when $z > 0$ and $-lC_j < z < qC_j \Rightarrow -l < \frac{z}{C_j} < q$ as shown below



then we have

$$f_Z(z) = \frac{1}{(q-l)\pi}\int_{\frac{z}{C_j}}^{q} \frac{1}{y}\frac{dy}{\sqrt{C_j^2-(\frac{z}{y})^2}}$$

doing the same change of variables and simplifying the results we get

$$f_Z(z) = \frac{1}{(q-l)\pi C_j}\left[\ln\left|\frac{C_j + \sqrt{C_j^2 - (z/q)^2}}{(z/q)}\right|\right].$$

The same result is obtained when $z < 0$. For other combinations of l and q one can exchange l and q where appropriate. In summary, the density $f_Z(z)$ can be written as follows.

$$f_Z(z) = \begin{cases} \frac{1}{|q-l|\pi C_j} \cdot \left[\ln\left|\frac{q}{l}\frac{C_j+\sqrt{C_j^2-(z/q)^2}}{C_j+\sqrt{C_j^2-(z/l)^2}}\right|\right], & \text{if } |z| \leq lC_j; \\ \frac{1}{|q-l|\pi C_j}\left[\ln\left|\frac{C_j+\sqrt{C_j^2-(z/q)^2}}{(z/q)}\right|\right], & \text{if } lC_j \leq |z| \leq qC_j; \\ 0 & \text{otherwise.} \end{cases}$$

Plots of the various individual probability density functions and the the convolution of these densities with themselves are shown in the next few pages for different combinations of l and q. Note that we set $C_j = 1.0$ since we assume all signals have equal powers.

In the following pages we present a plot of the conditional density functions for different values of l and q and the convolution of each individual density with it self:

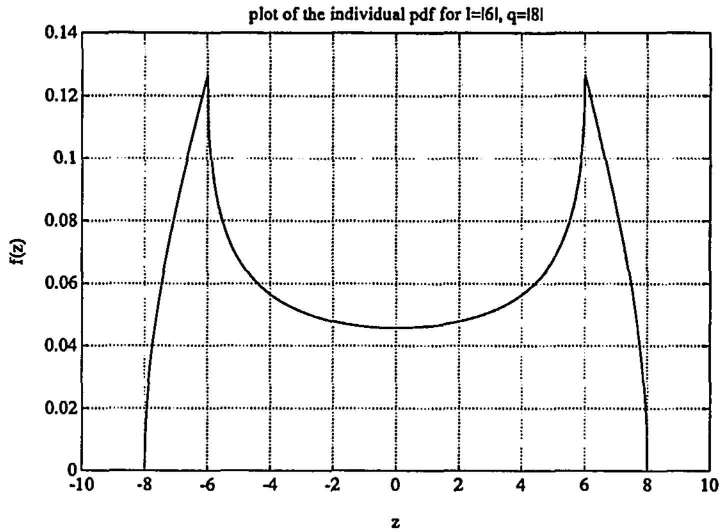


Figure (A.1)

The next plot shows the convolution of the density function above with itself:

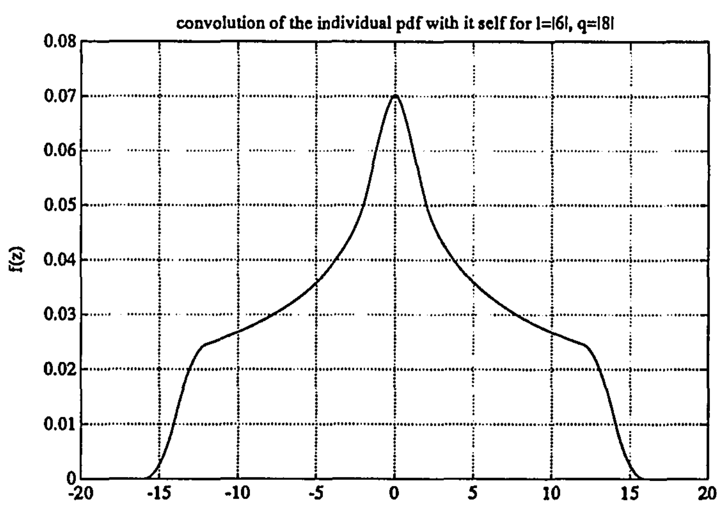


Figure (A.2)

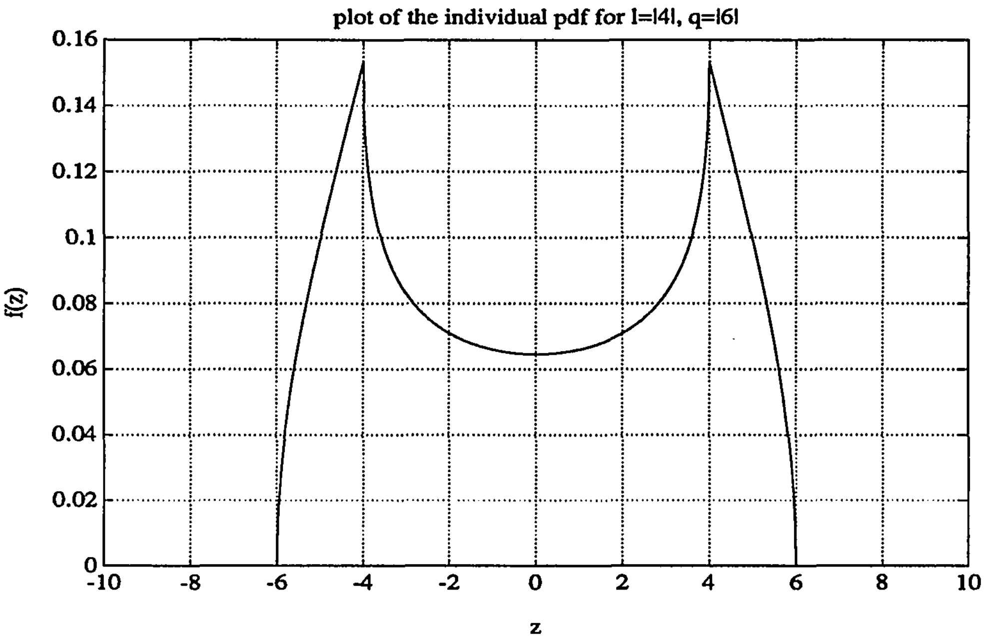


Figure (A.3)

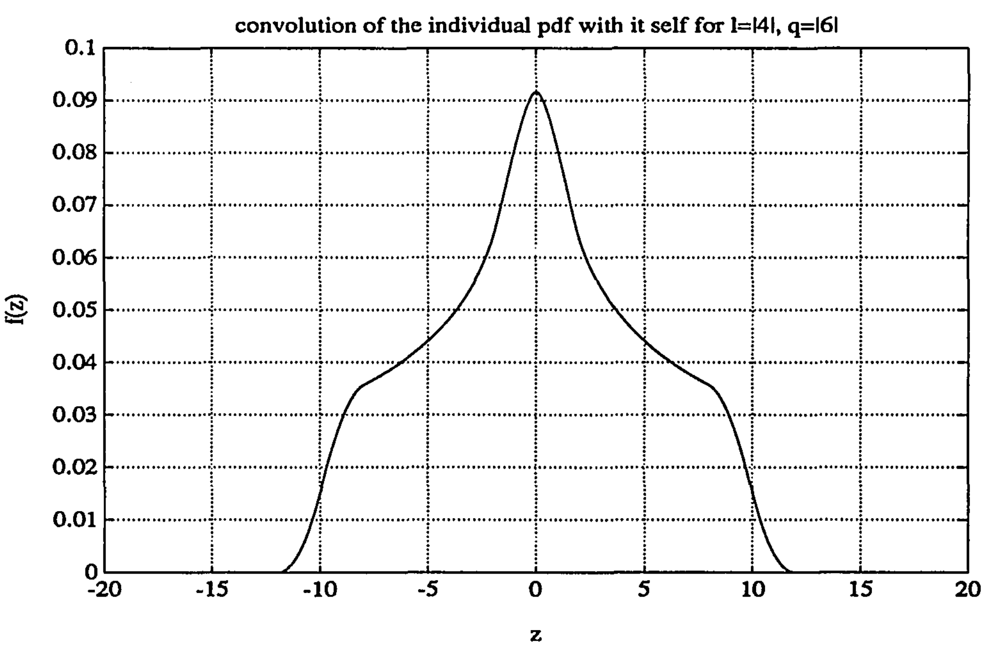


Figure (A.4)

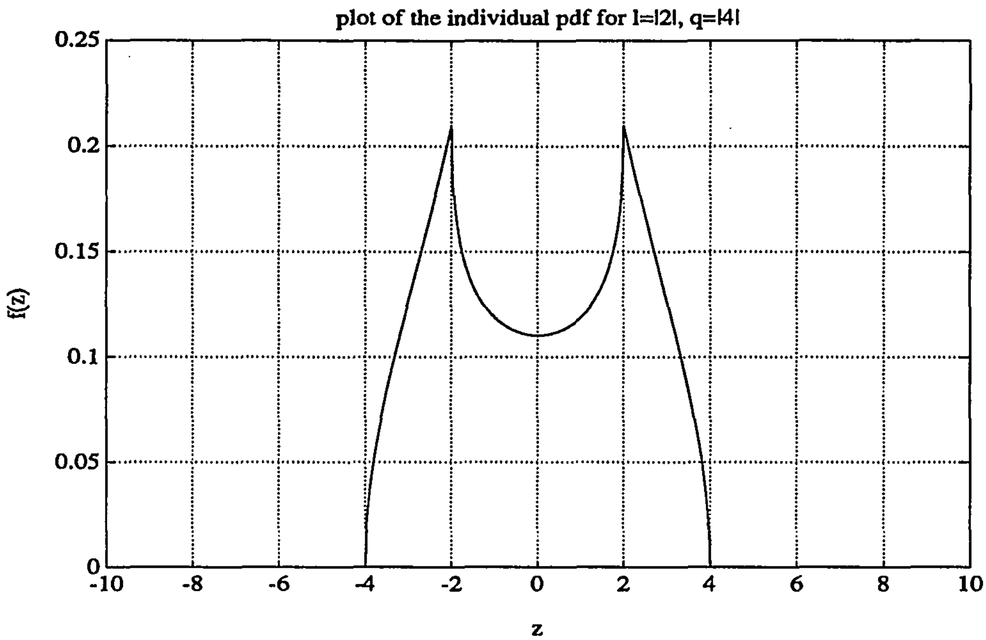


Figure (A.5)

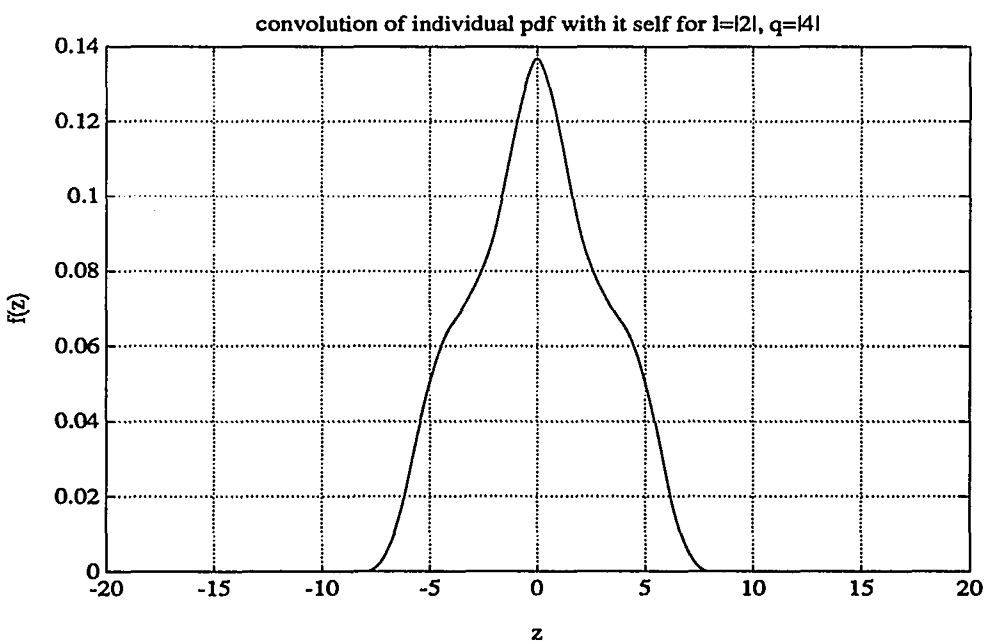


Figure (A.6)

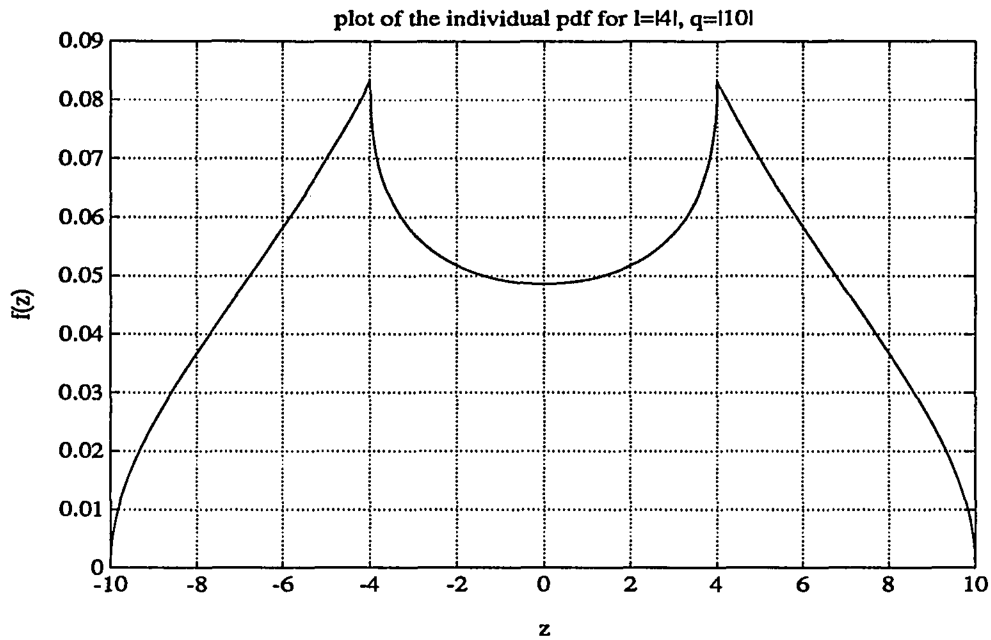


Figure (A.7)

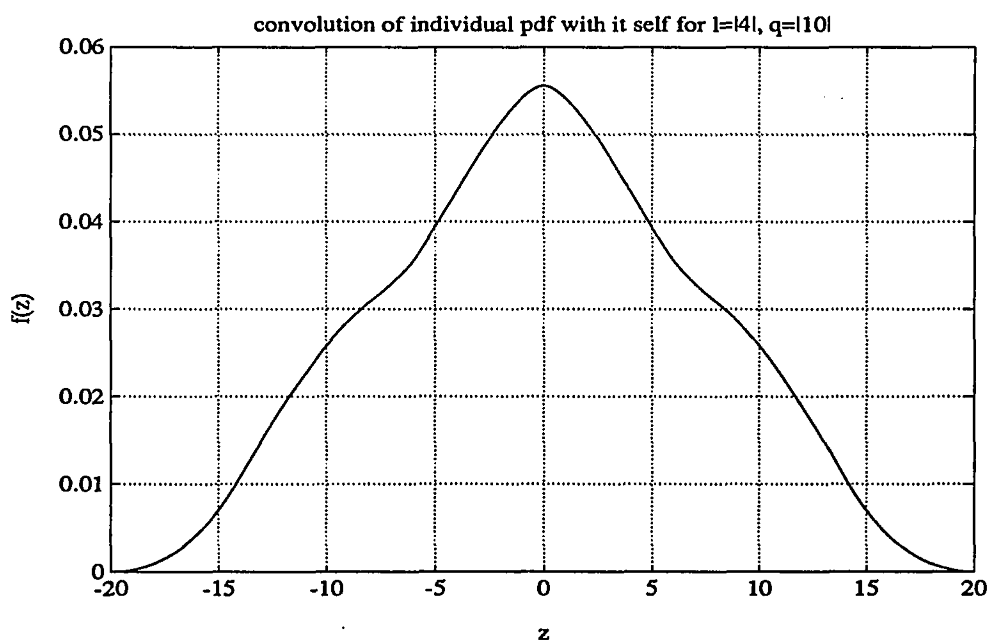


Figure (A.8)

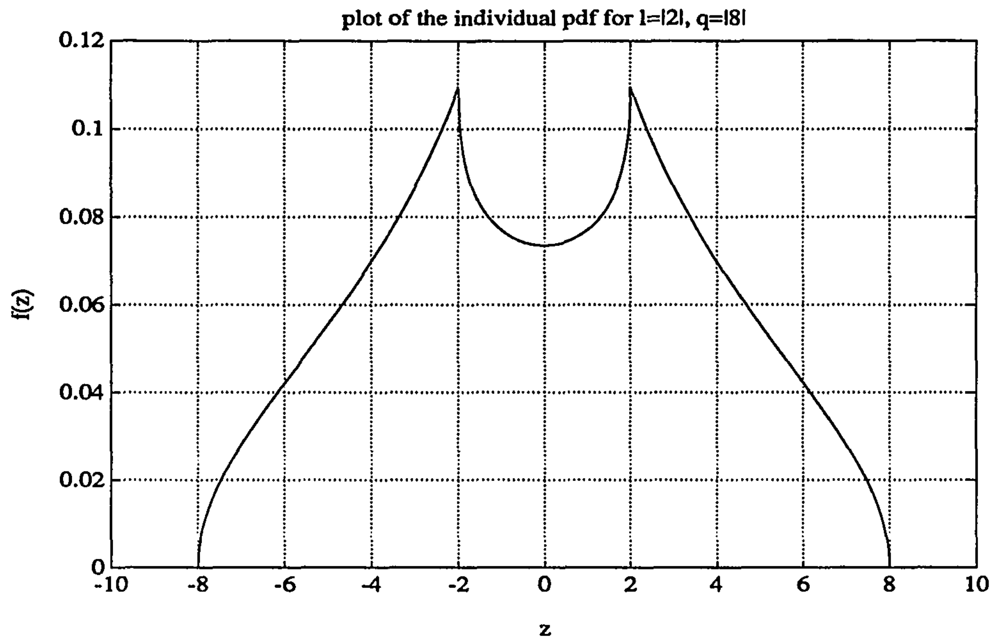


Figure (A.9)

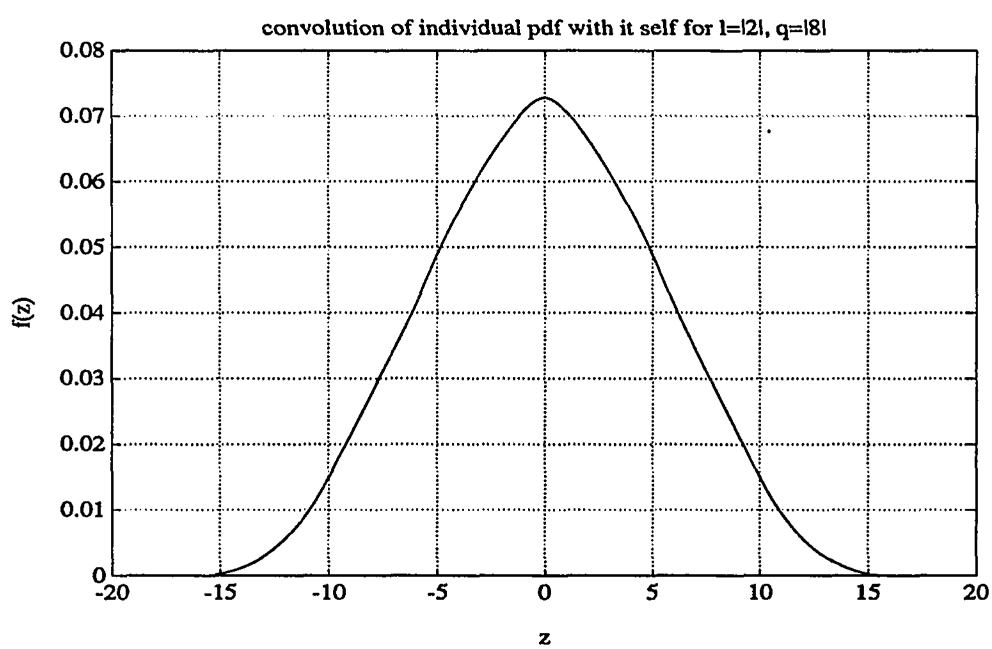


Figure (A.10)

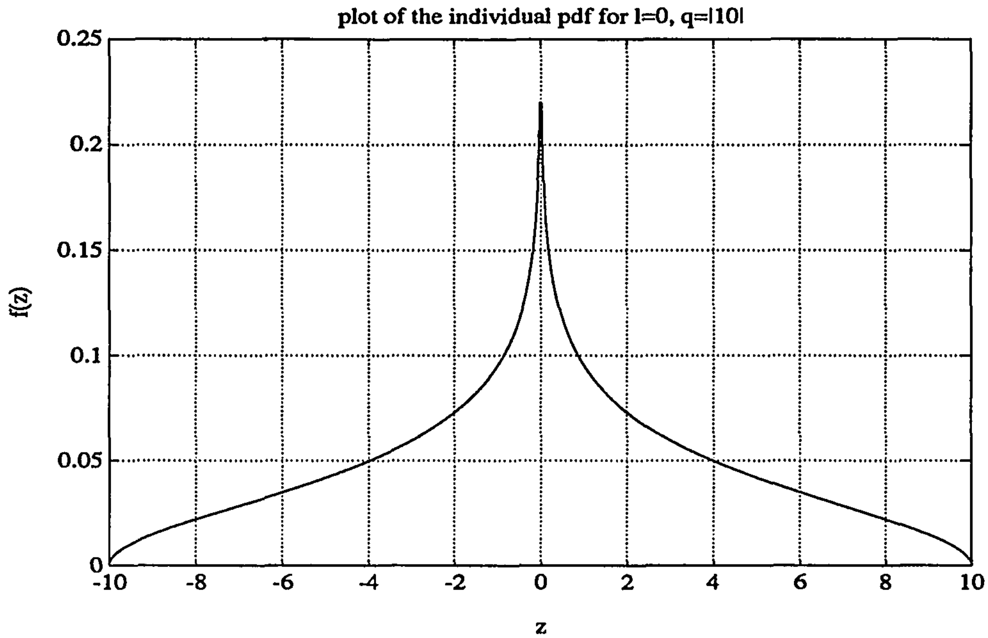


Figure (A.11)

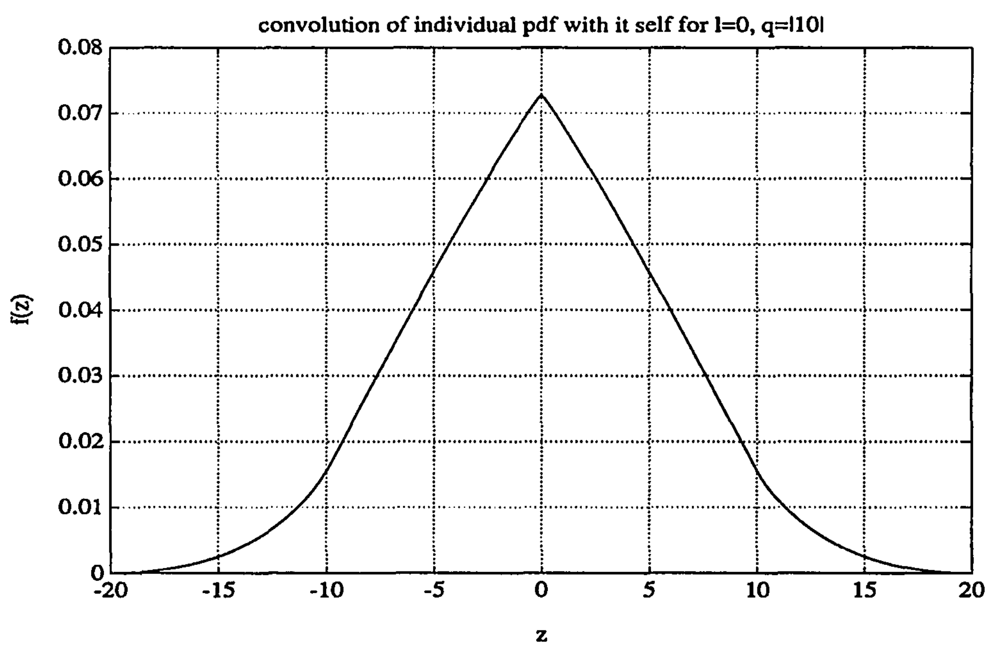


Figure (A.12)

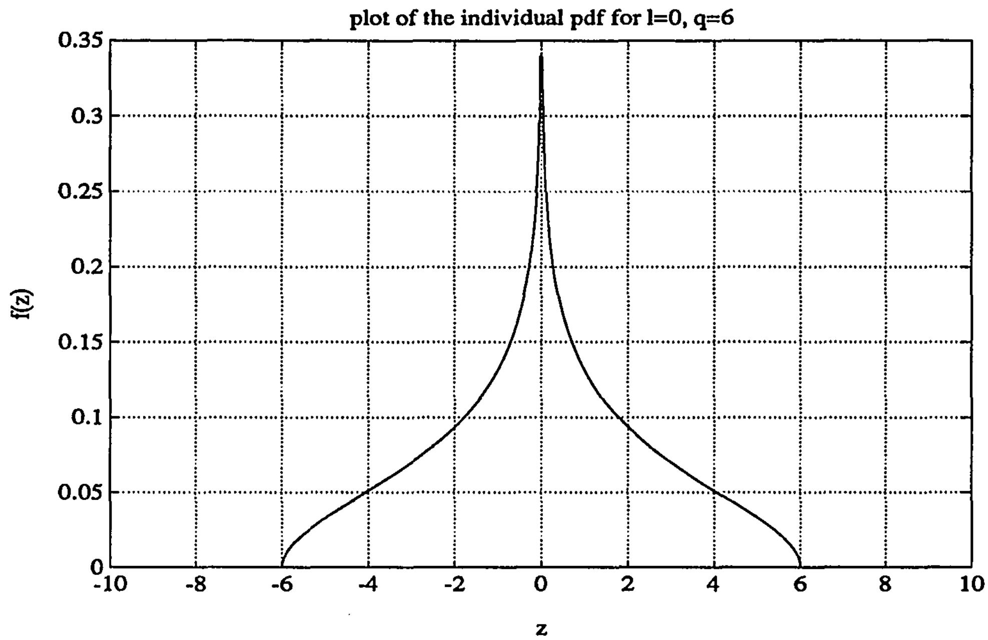


Figure (A.13)

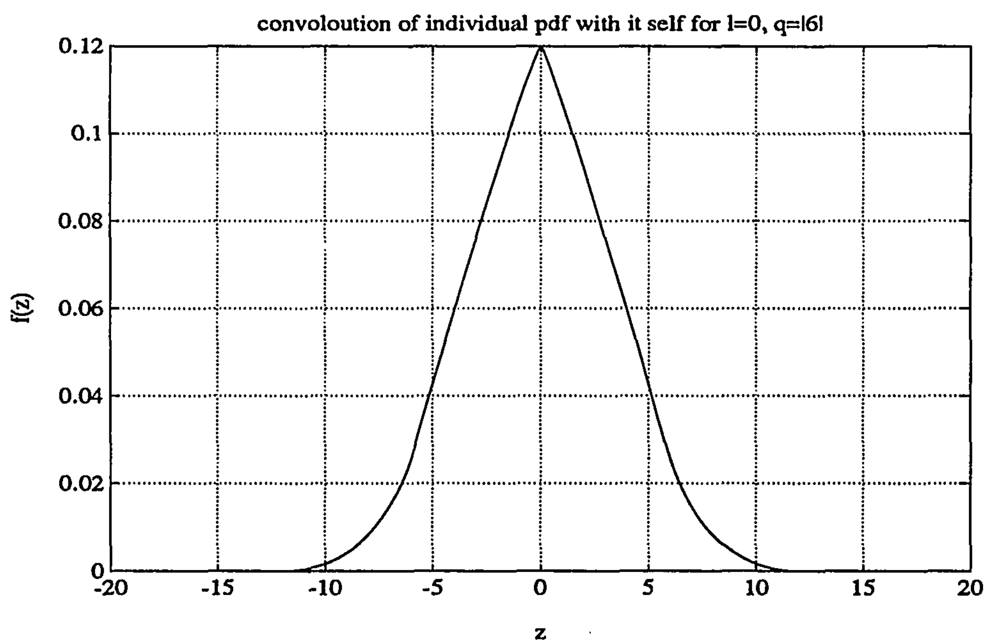


Figure (A.14)

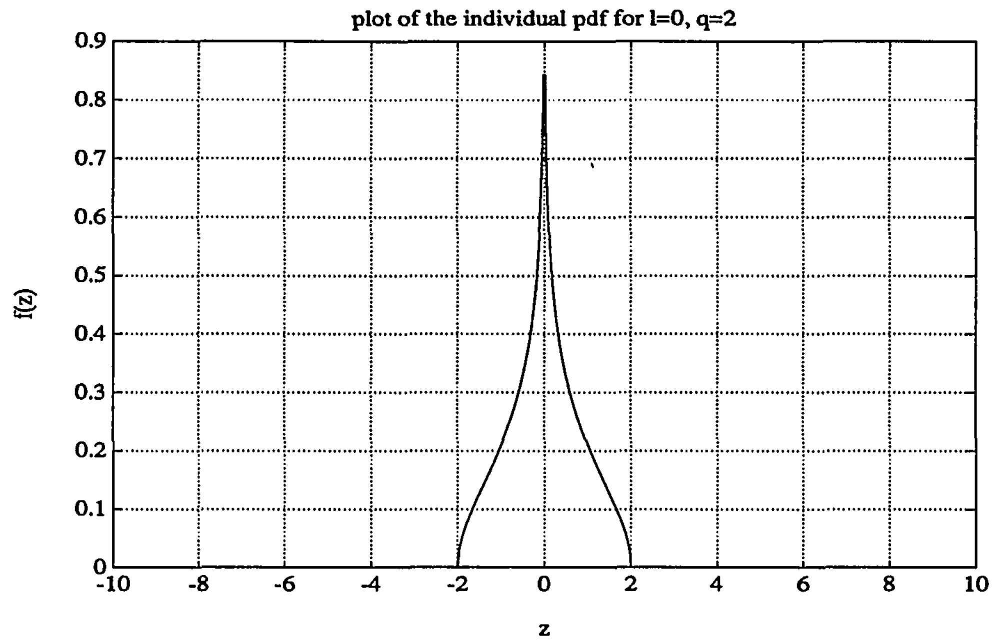


Figure (A.15)

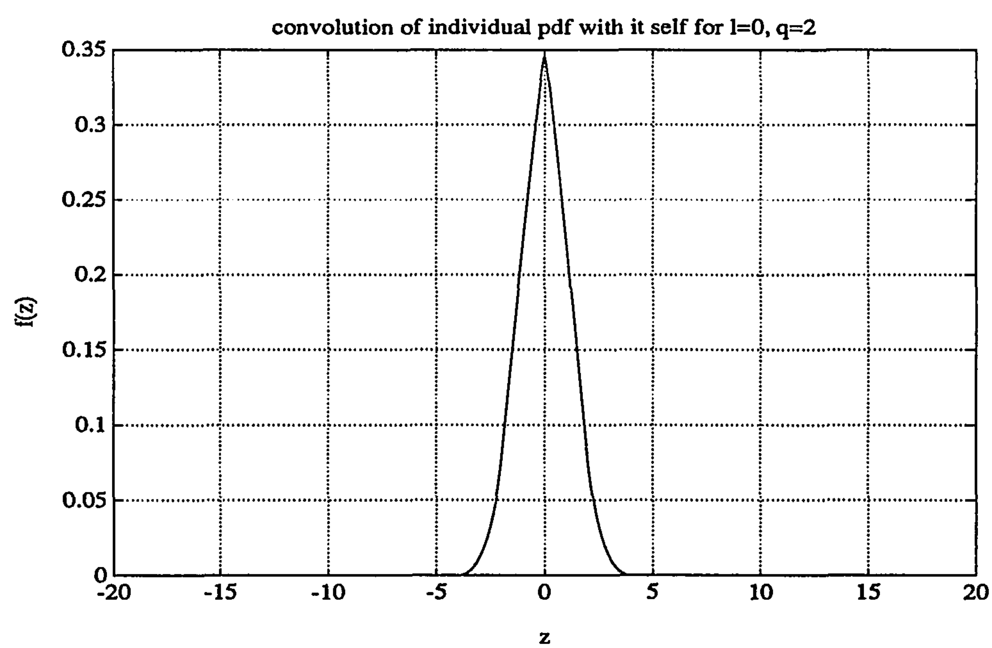


Figure (A.16)

References

[1] G. R. Cooper and C. D. McGillem, Modern Communications and Spread Spectrum. McGraw-Hill, New York, 1988.

[2] R. K. Bahr and J. A. Bucklew, " Fast Simulation of Detector Error Probabilities in the Presence of Memory and Non-Linearity," to appear in IEEE Transactions on Communications.

[3] J. A. Bucklew, Large Deviation Techniques in Decision, Simulation, and Estimation. Piggly–Wiggly Press, St. Oyster, MN., 1990.

[4] R. K. Bahr, private communications on spread spectrum systems and large deviations theory.

[5] R. K. Bahr private communications on numerical integration for estimating the BER.

[6] P. Ney and E. Nummelin, "Markov Additive Processes 1. Eigenvalue Properties and Limit Theorems," The Annals of Probability., Vol-15, No. 2, pp. 561-592, 1987.

[7] P. Ney and E. Nummelin, "Markov Additive Processes 2. Eigenvalue Properties and Limit Theorems," The Annals of Probability., Vol-15, No. 2, pp. 593-609, 1987.

[8] J. S. Sadowsky and J. A. Bucklew, "Large Deviations Theory Techniques in Monte Carlo Simulation". Winter Simulation Conference, 1989.

[9] J. G. Proakis, Digital Communications, Second Edition. McGraw-Hill, New York, 1989.

[10] R. E. Ziemer and W. H. Tranter, Principles of Communications Systems, Modulation, and Noise, Third Edition. Houghton Mifflin, Boston, MA, 1990.

[11] J. S. Lehnert and M. B. Pursley, " Error Probabilities for Binary Direct-Sequence Spread-Spectrum Communications with Random Signature Sequences," IEEE Transactions on Communications, Vol. Com-35, No.1, pp. 87-98, Jan. 1987.

[12] E. A. Geraniotis and M. B. Pursley, " Error Probability for Direct Sequence Spread Spectrum Multiple Access Communications," IEEE Transactions on Communications, Vol. Com-30, No. 5, pp. 985-995, May 1982.

[13] D. Lu and K. Yao, "Improved Importance Sampling Technique for efficient Simulation of Digital Communication Systems," IEEE Journal of Selected Areas in Communications, Vol. 6, No. 1, pp. 67-75, Jan. 1988.

[14] P. M. Hahn and M. C. Jeruchim, " Developments in the Theory and Application of Importance Sampling," IEEE Transactions on Communications, Vol. Com-35, No. 7, pp. 706-714, July 1987.

[15] A. W. Lam and F. M. Ozluturk, "Performance Bounds for Direct-Sequence Spread-Spectrum Communications with Complex Signature Sequences". Tenth Annual International Phoenix Conference on Computers and Communications. Scottsdale, Arizona. March 27-30 1991, pp 408-414.

[16] A. Papoulis, Probability, Random Variables, and Stochastic Processes, Second Edition. McGraw-Hill, New York, 1984.

[17] Hoel, Port, and Stone, Introduction to Probability Theory. Houghton Mifflin, Boston, MA.

[18] Hoel, Port, and Stone, Introduction to Stochastic Processes. Houghton Mifflin, Boston, MA.

[19] E. Lee and Messerschmitt, Digital Communication. Kluwer Academic Publishers.