Orthogonal Short Codes for Code Division Multiple Access Networks

Khalil Kanj

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montréal, Québec, Canada

April 28, 1995

ISBN 0-612-05133-1

ABSTRACT

Orthogonal Short Codes for Code Division Multiple Access Networks

Khalil Kanj

Short codes are known to have better balance and spectral characteristics than their long counterparts. Extensive simulations have been conducted in order to compare a long code consisting of a sequence of length $2^{15} - 1$ chips, to a code consisting of nine short codes each having a length of $2^7 - 1$ chips. In the worst case, the system switches randomly among those short codes at each data bit. Many types of short concatenated codes have been investigated and compared to each other. However, only the case of asynchronous concatenated short codes has been considered for the purpose of comparision with long code.

Different fading channels have been considered in the simulation. It is shown that long code outperforms asynchronous short codes for large number of users under Rayleigh or Rician flat fading. On the other hand, they have approximately the same performance under Rayleigh or Rician frequency selective fading. Since error correcting codes are not used in the simulated cases, number of users has been taken in the range of 20 users in order to have low bit error rate. Adding white Gaussian noise, shadow, power control error to the fading channels and introducing orthogonal Walsh functions (orthogonal codes) in the proposed system, it is shown that the performance of short codes is approximately the same as long codes or even better in some cases. This result enhances the use of short codes which are known to have less synchronization time and implementation complexity than long codes.

To Mom, Dad and my family

for their encouragement and patience.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Professor A. K. Elhakeem for his guidance and support throughout the course of my thesis work. This work was born from his ideas and he provided constructive suggestions to enable me to demonstrate this new technique by mathematical analysis and computer simulation. I am grateful to him for his sincere efforts. I would like to acknowledge gratefully the valuable suggestions and constructive criticism of Professor Tho Le-Ngoc.

Throughout my educational efforts my family has provided me ample moral support and I am grateful to them for their encouragement and patience. Finally I would like to express my gratitude to my friends who cheered me up.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Spread Spectrum and Satellite Communications: an Introduction

Up until fairly recently, most interest in spread spectrum was dominated by the classical military scenario of intentional smart jamming [1]. However, once the threat of having to design a system which is capable of combatting an intelligent adversary is removed, one can consider ways of improving the receiver design to make the system more efficient for commercial applications.

In this introductory chapter, the three main techniques adopted in spread spectrum, i.e. direct sequence (DS), frequency hopping (FH), and time hopping (TH), are described. Satellite communications subsystems and satellite links are presented, since the base station for cellular mobile satellite channels would be the satellite itself.

1.1 Basic Principles and Features

There are several ways to implement a spread spectrum system. Each requires: 1) signal spreading by means of a code; 2) synchronization between pairs of users; 3) care to insure that some of the signals do not overwhelm the others (near-far problem); 4) source and channel coding to optimize performance and total throughput.

In general, there are three techniques: 1) spectrum spreading (spread spectrum); 2) time spreading (time hopping); and 3) hybrid spreading.

1.1.1 Spread Spectrum (SS) Techniques:

There are two greneral spread spectrum techniques, direct sequence (DS) and frequency hopping (FH).

1) Direct Sequence (DS) :

Direct sequence spread spectrum systems are so called because they employ high speed code sequences, which is used to modulate the data modulated carrier (such as BPSK or M-ary PSK).

Figs.1.1 and 1.2 show the block diagram of a simplified direct sequence system. The idea behind spread spectrum is to transform a signal with bandwidth B into a noise-like signal of much larger bandwidth B_{ss}. We see from Fig.1.3 that the spreading is accomplished by multiplying the modulated information-bearing signal by a binary $\{\pm 1\}$ baseband code sequence waveform, $PN_k(t)$, where T_c, duration of a chip period, is much less than T, duration of a data bit period.

The ratio B_{ss}/B, called the processing gain, is typically 10-30 dB. Hence the power of the radiated spread spectrum signal is spread over 10-1000 times the original bandwidth, while its power spectral density is correspondingly reduced by the same amount. The other requirement of the spread spectrum signal is that it be "noise-like". That is, each spread spectrum signal should behave as if it were "uncorrelated" with every other spread signal using the same band. In practice, the correlation used need not be zero, i.e. the variability of differential propagation delay differences between the paths from the central station to the various users may force these codes to practically become semiorthogonal rather than completely orthogonal.

The code sequence may be thought of as being (pseudo) randomly generated so that each binary chip can change (with probability = 1/2) every T_c sec. Thus the signal for the ith transmitter is:

$$S_i(t) = Ad(t)PN_i(t)\cos(\omega_0 t + \phi) \tag{1.1}$$

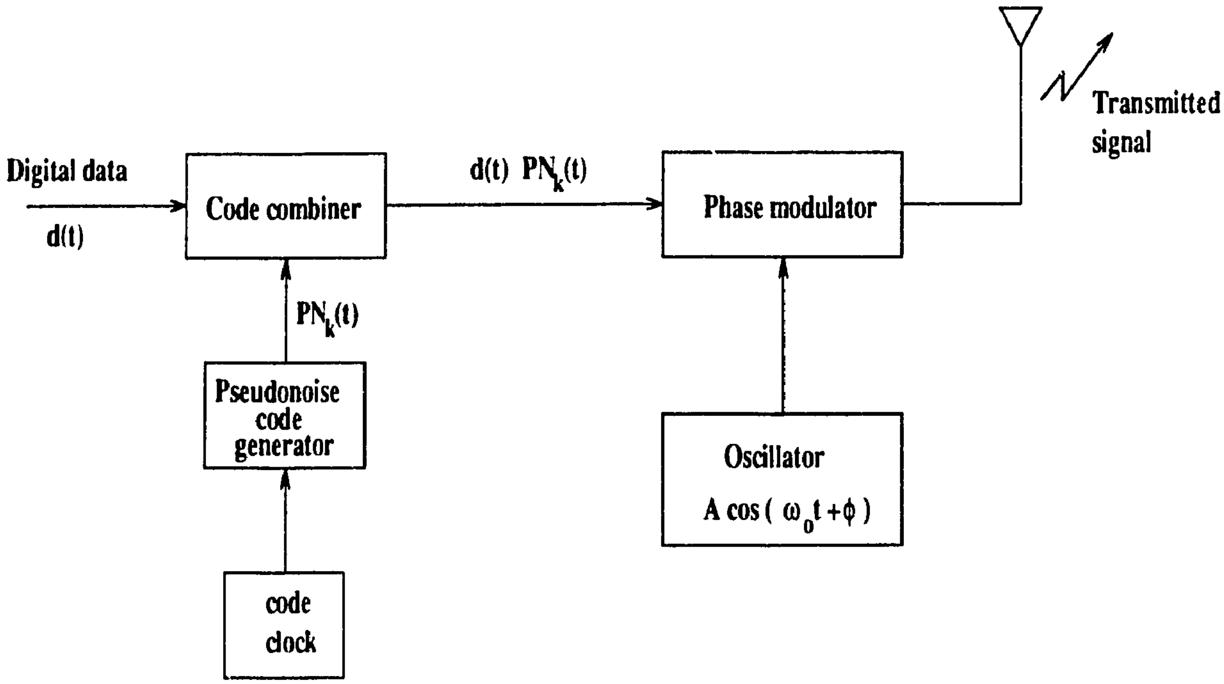


Figure 1.1: Block diagram of a direct sequence transmitter system.

where $d(t)$ is the data modulation (assumed to be ± 1 for BPSK signaling), A is the amplitude of the BPSK waveform, $f_0 = \frac{\omega_0}{2\pi}$ is the carrier frequency, and ϕ is a random phase. Since T_c is much less than T, the ratio of the spread bandwidth, B_{ss}, to the unspread bandwidth, B, is given by $B_{ss}/B = T/T_c = N$, the processing gain. It is clear that a receiver with access to $PN_i(t)$, and synchronized to the spread spectrum transmitter, can receive the data signal, $d(t)$, by a simple correlation. That is, in the interval $[0, T]$, if the data symbol is d_1, which can take on values ± 1, then

$$\frac{2}{T}\int_0^T PN_i(t)S_i(t)\cos(\omega_0 t + \phi)dt = Ad_1 = \pm A \tag{1.2}$$

A properly designed spreading sequence may have the following properties: 1) in a long sequence, about half the chips will be $+1$ and half will be -1; 2) a run of length r chips of the same sign will occur about $2^{-r}l$ times in a sequence of l chips; 3) the autocorrelation of the sequence $PN_i(t)$ and $PN_i(t+\tau)$ will be very small except in the vicinity of $\tau = 0$; 4) the crosscorrelation of any two sequences $PN_i(t)$ and $PN_j(t+\tau)$ will be small.

An important class of sequences called maximal length linear feedback shift register sequences are well known to exhibit properties 1) , 2) and 3). The code

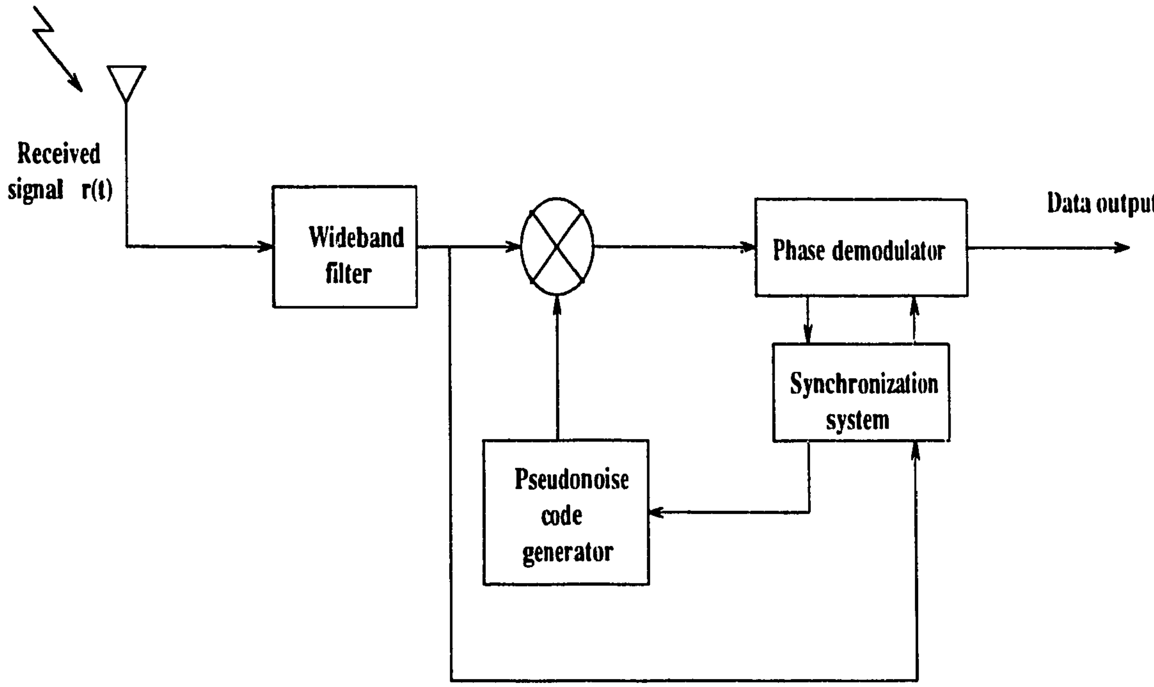


Figure 1.2: Receiver of a direct sequence system.

length L of any pseudonoise (PN) code generator is dependent upon the number of shift registers K,

$$L = 2^K - 1 \tag{1.3}$$

In particular, the autocorrelation function,

$$R_i(\tau) = \frac{1}{T_q}\int_0^{T_q} PN_i(t)PN_i(t+\tau)dt \tag{1.4}$$

is given by

$$R_i(\tau) = \begin{cases} 1 - \frac{\tau}{T_q}(1 + \frac{T_c}{T_q}) & ; 0 \leq \tau \leq T_c \\ -\frac{T_c}{T_q} & ; T_c \leq \tau \leq (N-1)T_c \\ \tau - \frac{T_q - T_c}{T_c}(1 + \frac{T_q}{T_c}) - \frac{T_q}{T_c} & ; (N-1)T_c \leq \tau \leq NT_c \end{cases} \tag{1.5}$$

where T_q is the period of the sequence and $R_i(\tau)$ is also periodic with period T_q. The enhancement in performance obtained from a DS spread spectrum signal through the processing gain and coding gain can be used to enable many DS spread spectrum signals to occupy the same channel bandwidth provided that each signal has its own distinct PN sequence. Thus it is possible to have a large community of relatively uncoordinated users transmit messages simultaneously over the same channel

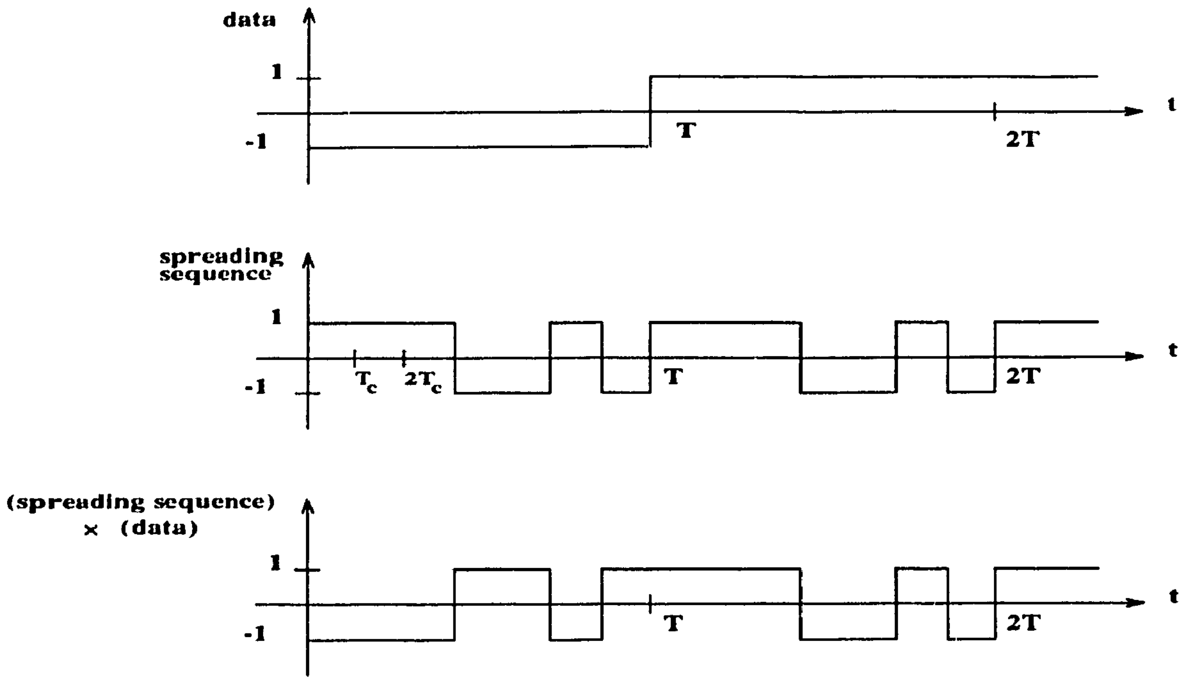


Figure 1.3: Time waveforms involved in generating a direct sequence signal.

bandwidth [3]. This type of digital communication in which each user (transmitter-receiver pair) has a distinct PN code for transmitting over a common channel bandwidth is called either *code division multiple access* (CDMA) or *spread spectrum multiple access* (SSMA).

In the demodulation of each PN signal, the signals from the other simultaneous users of the channel appear as an additive interference. The level of interference varies depending on the number of users at any given time. A major advantage of CDMA is that a large number of users can be accommodated if each transmits messages for a short period of time. In such a multiple access system it is relatively easy either to add new users or to decrease the number of users without disrupting the system [5].

If we assume that all signals have identical average powers, the desired signal-to-interference power ratio at a given receiver is

$$\frac{S_{av}}{I_{av}} = \frac{S_{av}}{(N_u - 1)S_{av}} = \frac{1}{N_u - 1} \tag{1.6}$$

where N_u is number of simultaneous users.

Since CDMA capacity is only interference limited (unlike FDMA and TDMA capacities which are primarily bandwidth limited), any reduction in interference converts directly and linearly into an increase in capacity. Voice activity and spatial isolation were shown to be sufficient to render CDMA capacity at least double that of FDMA and TDMA under similar assumptions for mobile satellite applications [4].

There are many attributes of direct sequence CDMA which are of great benefit to the cellular system [2]: 1) Voice activity cycles; 2) No equalizer needed; 3) One radio per site; 4) Soft handoff; 5) No guard time in CDMA; 6) Sectorization for capacity; 7) Less fading; 8) Easy transition; 9) Capacity advantage; 10) No frequency management or assignment needed; 11) Soft capacity; 12) Coexistence or ability to overlay; 13) For microcell and in-building systems.

Disadvantages of CDMA are: 1) Near-far effect; 2) Self-jamming; 3) Stringent synchronization requirements.

2)Frequency Hopping (FH):

In frequency hopping the modulated signal is first generated, then the spread spectrum technique consists of changing pseudo-randomly the center frequency of the transmitted signal every T_H sec, according to the output from a PN generator, so that the hop rate is $f_H = 1/T_H$ hops/sec. Figs. 1.4 and 1.5 show block diagrams of the transmitter and receiver for a frequency-hopped spread spectrum system. The modulation is either binary or M-ary FSK. For example, if binary FSK is employed, the modulator selects one of two frequencies corresponding to the transmission of either a 1 or a 0. The resulting FSK signal is translated in frequency by an amount that is determined by the output sequence from the PN generator which, in turn, is used to select a frequency that is synthesized by the frequency synthesizer. This frequency is mixed with the output of the modulator and the resultant frequency-translated signal is transmitted over the channel. for example, m bits from the PN generator may be used to specify $2^m - 1$ possible frequency translations.

At the receiver, we have an identical PN generator, synchronized with the received signal, which is used to control the output of the frequency synthesizer.

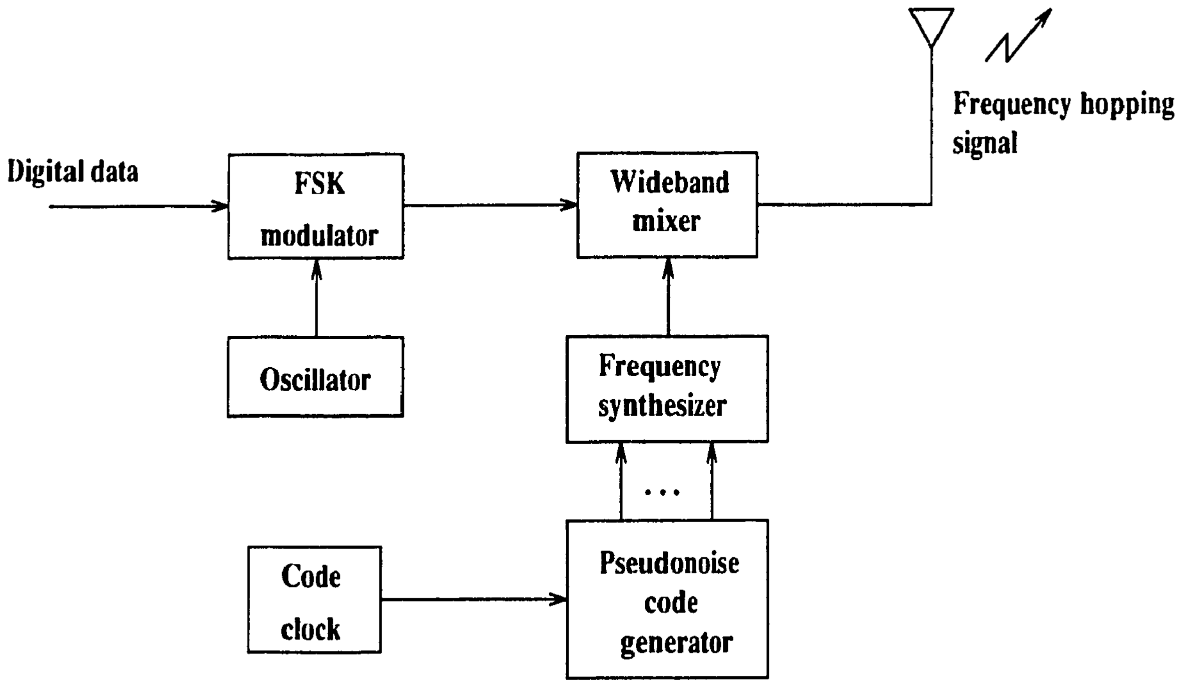


Figure 1.4: Block diagram of the transmitter of a frequency hopping system.

Thus the pseudo-random frequency translation introduced at the transmitter is removed at the receiver by mixing the synthesizer output with the received signal. The resultant signal is demodulated by means of an FSK demodulator. A signal for maintaining synchronism of the PN generator with the frequency-translated received signal is usually extracted from the received signal.

Although PSK modulation gives better performance than FSK in an AWGN channel, it is difficult to maintain phase coherence in the synthesis of the frequencies used in the hopping pattern and, also, in the propagation of the signal over the channel as the signal is hopped from one frequency to another over a wide bandwidth. Consequently FSK modulation with noncoherent detection is usually employed with FH spread spectrum signals.

Frequency hopping could be done slowly (one hop per many symbols) or fast (many hops per symbol). Due to limitation of today's technology, the FH is using a slow hopping pattern [2]. Fast frequency hopping is employed in anti-jamming (AJ) applications when it is necessary to prevent a type of jammer, called a *follower jammer*, from having sufficient time to intercept the frequency and retransmit it along with adjacent frequencies so as to create interfering signal components. However,

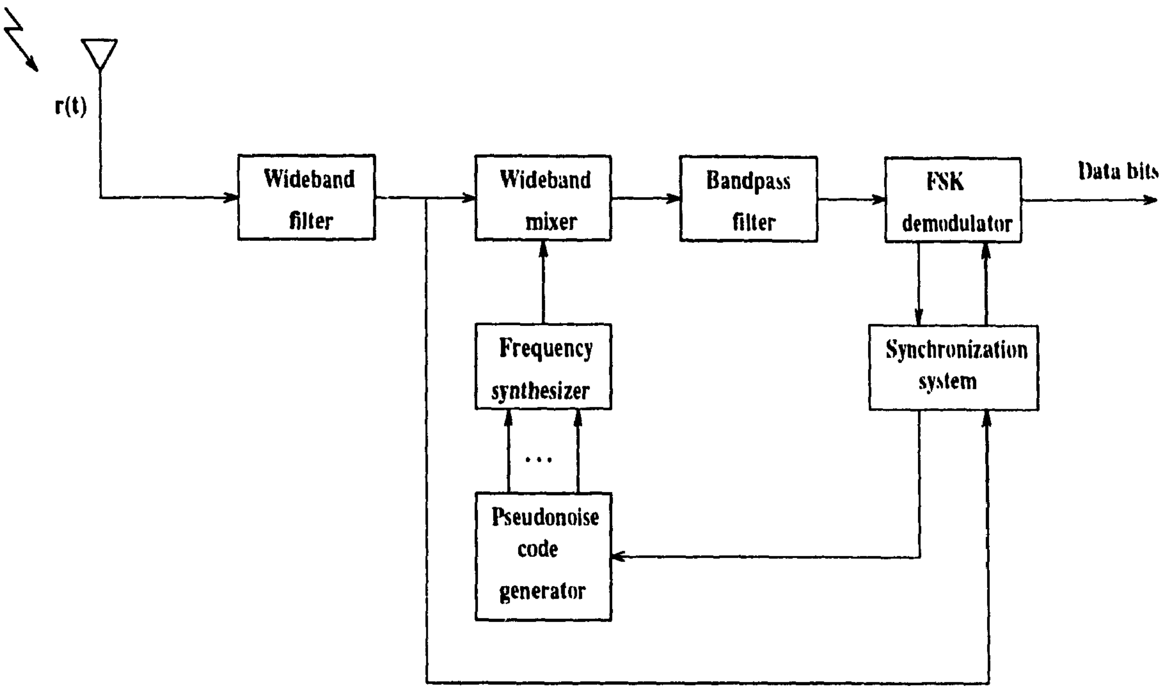


Figure 1.5: Block diagram of the receiver of a frequency hopping system.

there is a penalty incurred in subdividing a signal into several frequency-hopped elements because the energy from these seperate elements is combined noncoherently. Consequently the demodulator incurs a penalty in the form of a noncoherent combining loss.

The total frequency spread B_{ss} is equal to the total number N of distinct frequencies used for hopping times the bandwidth occupied by each frequency.

For slow frequency hopping, $N/T = B_{ss}$, or $N = B_{ss}/B$ where $B \cong 1/T$ and T is the duration of the data symbol. The spectrum spreading in FH is measured by the processing gain, PG, as $PG = 10 \log N$ (dB).

To make the FH signal "noise-like", the frequency hopping pattern is driven by a "pseudo-random" number generator having the property of delivering a uniform distribution for each frequency that is "independent" on each hop.

For a Direct Sequence (DS) (Frequency Hopping (FH)) spread spectrum system, binary PSK (binary FSK) or M-ary PSK (M-ary FSK) may be used for transmission but M-ary transmission has the advantage that more data per chip (hop) can be transmitted.

Frequency hopping system advantages: 1) Greater amount of spreading; 2)

Can be programmed to avoid portions of the spectrum (not requiring a contiguous band); 3) Relatively short aquisition time; 4) Less affected by near-far problem for highly loaded system.

Disadvantages of an FH system: 1) Complex frequency synthesizer; 2) Not useful for range and range rate measurement; 3) Error correction required.

1.1.2 Time Hopping (TH)

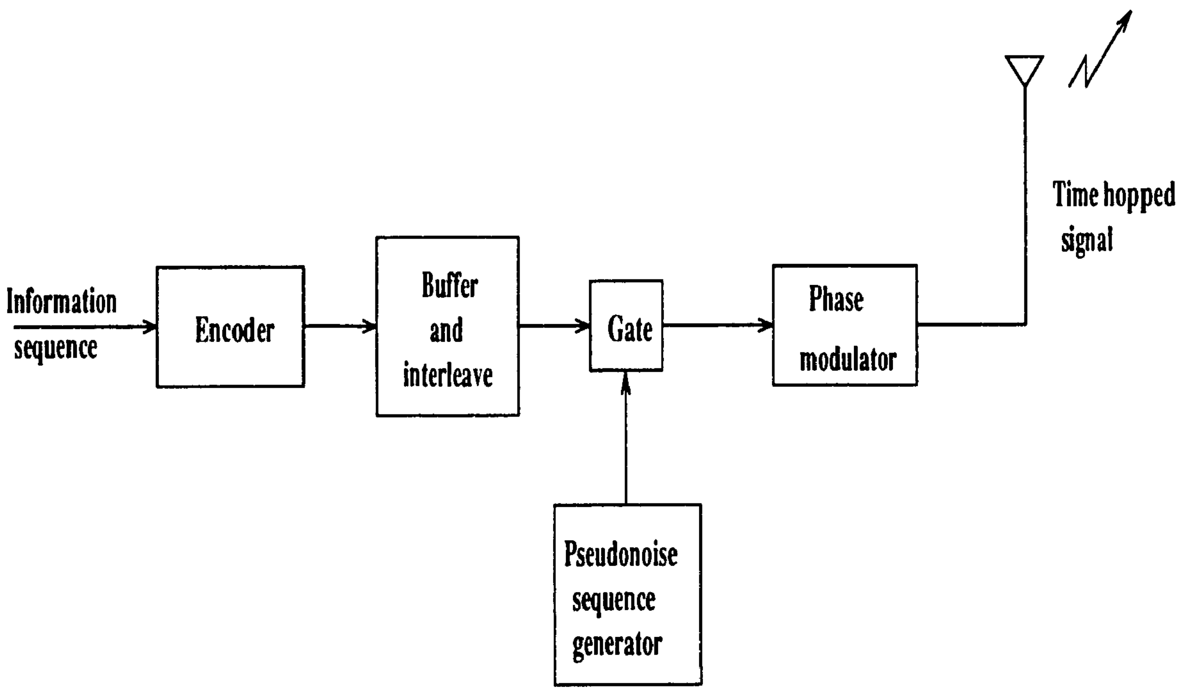


Figure 1.6: Generic transmitter of a time hopping system.

In time hopping systems spreading of the spectrum is achieved by compressing the information signal in the time domain. That is, the time hopping systems control their transmission time and period with a code sequence in the same way that frequency hoppers control their frequency. The message bit period, T, may be divided into a number of nonoverlapping subintervals. One of the subintervals is selected pseudo-randomly each bit interval, and a pulse is transmitted which conveys the value of the bit. The subinterval selected, τ, is independent of the bit value and is independent bit-to-bit. The position of the pulse is uniformly distributed over the bit interval, but could be predicted by a properly synchronized, friendly receiver.

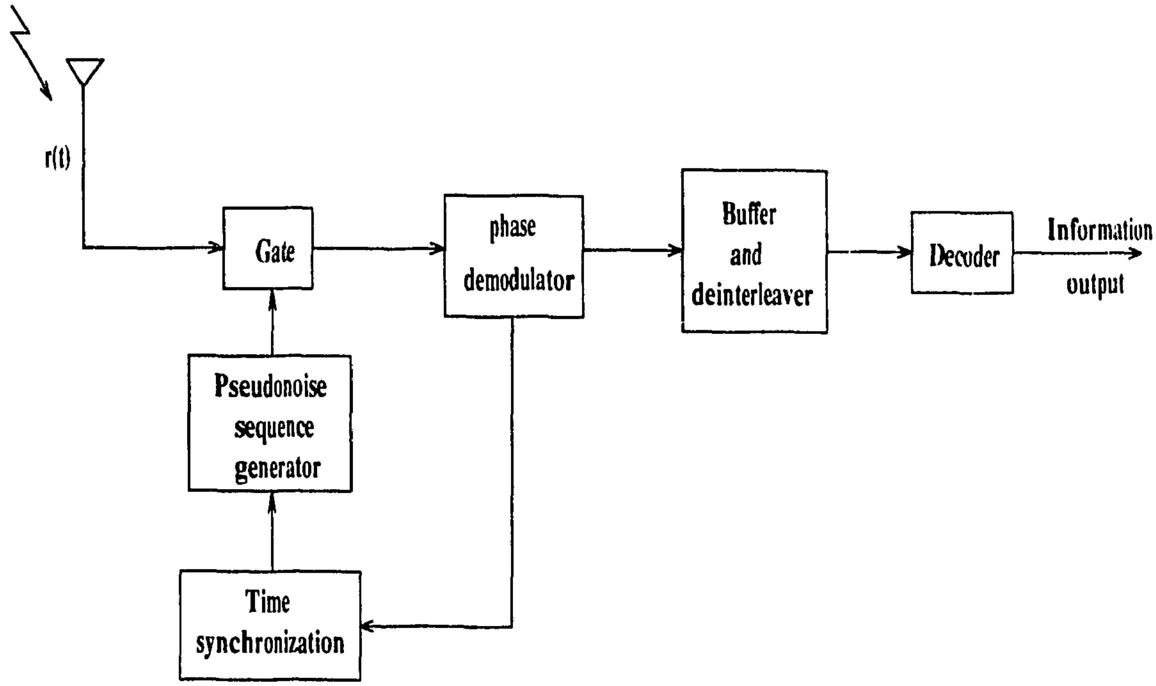


Figure 1.7: Generic receiver of a time hopping system.

In a sense, TH uses time slots or subintervals in a way which is analogous to the manner FH uses frequency cells. However, in TH the power level is increased to hold the energy per bit constant. Consequently, like FH, the signal power spectral density for TH signals is about the same as a conventional signal having the same energy per bit.

In the TH scheme described above, the pulse rate in TH is the same as the bit rate of the message it transmits. Rather than transmit one pulse in every bit interval, it is also possible to store up bits over several intervals and then transmit several bits together.

Figs. 1.6 and 1.7 show the block diagram of a time hopping system. Interference among simultaneous users in a time hopping system can be minimized by coordinating the times at which each user can transmit a signal, which also avoids the problem of very strong signals at a receiver swamping out the effects of weaker signals. In a non-coordinated system, overlapping transmission bursts will result in message errors, and for this it will normally require the use of error-correction coding to restore the proper message bits.

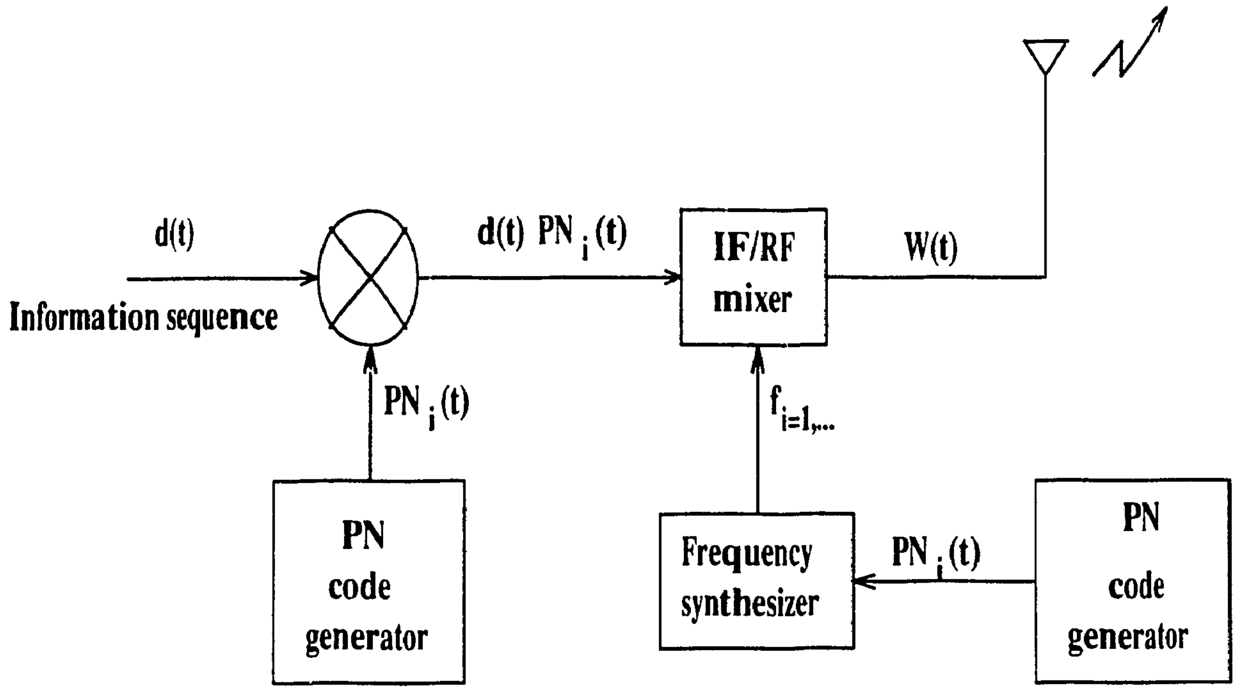


Figure 1.8: Block diagram of a typical DS/FH transmitter system.

Advantages of a time hopping system: 1) High bandwidth efficiency; 2) Implementation simpler than FH.

Disadvantages of a time hopping system: 1) Long acquisition time; 2) Error correction needed.

1.1.3 Hybrid spread spectrum systems

Hybrid spread spectrum systems are made up by combining two or more of, direct-sequence, frequency-hopping, time-hopping, to offer certain advantages of a particular technique while avoiding the disadvantages, or to offer very wide-band and/or very high processing gain, or to combine some of the advantages of two or three types of systems in a single system, and minimize the disadvantages of those types. Many different hybrid combinations are possible. Some of these are: 1) DS/FH; 2) DS/TH; 3) FH/TH; 4) DS/FH/TH. In this paper we will concentrate on only one hybrid system, namely DS/FH system.

DS/FH hybrid system:

Figs. 1.8 and 1.9 show the block diagrams of a typical DS/FH system. In

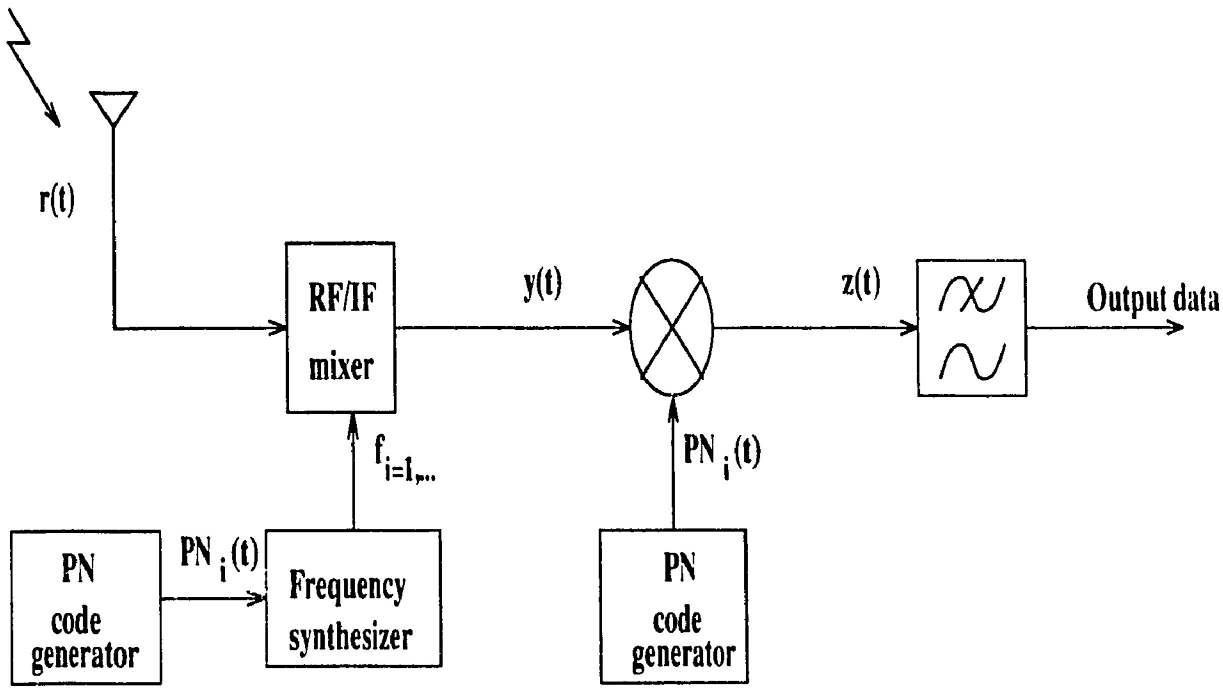


Figure 1.9: Generic receiver of a DS/FH system.

this system PN code is used to spread the signal to an extent limited by either code generator acquisition time or speed, and the frequency hopping would be used to increase the frequency spread. The DS code rate is normally much faster than the rate of frequency hopping, and the number of frequency channels available is usually much smaller than the number of code chips. The signal transmitted on a single hop consists of a DS spread spectrum signal which is demodulated coherently. However, the received signals from different hops are combined noncoherently (envelope or square-law combining). Since coherent detection is performed within a hop, there is an advantage obtained relative to a pure FH system. The hybrid DS/FH processing gain is a function of

$$BW_{RF}/R_{info} \tag{1.7}$$

where BW_{RF} is the RF bandwidth of the transmitted signal and R_{info} is the bit rate of the information. A high processing gain is easily achieved by the FH technique, and the signal keeps all the great advantages of the direct sequence modulation as the resistance to narrow band interference, and multiple path rejection. However, the price paid for the gain in performance is an increase in complexity, greater cost, and more stringent timing requirements.

1.2 Communication Satellites

A *communication satellite* is basically an electronic communication package placed in orbit. The prime objective of the satellite is to initiate or aid communication transmission from one point to another. Using satellites for commercial applications has become more frequent and useful these days if we take into consideration their wide coverage and good signal reception. Also they sometimes provide the only reliable solution for transmission and reception especially in the case of bad weather areas like the arctic part of Canada and US. The next section addresses briefly the communications subsystems, uplink channel, and downlink channel of a typical satellite.

1.2.1 Satellite Communications Subsystems

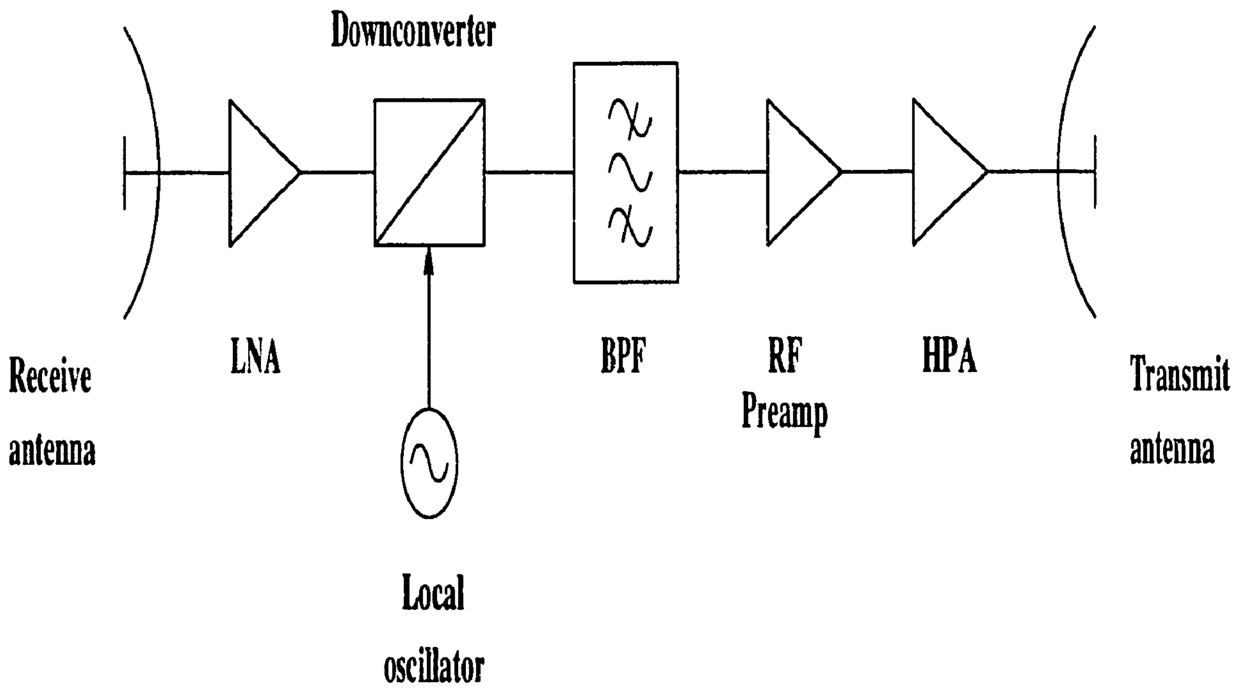


Figure 1.10: Simplified single-conversion transponder.

The communications subsystem is the major component of a communications satellite, and the remainder of the spacecraft is there only to support it. Usually the communications equipment is composed of one or more antennas, which receive and transmit over wide bandwidths at microwave frequencies, and a set of receivers

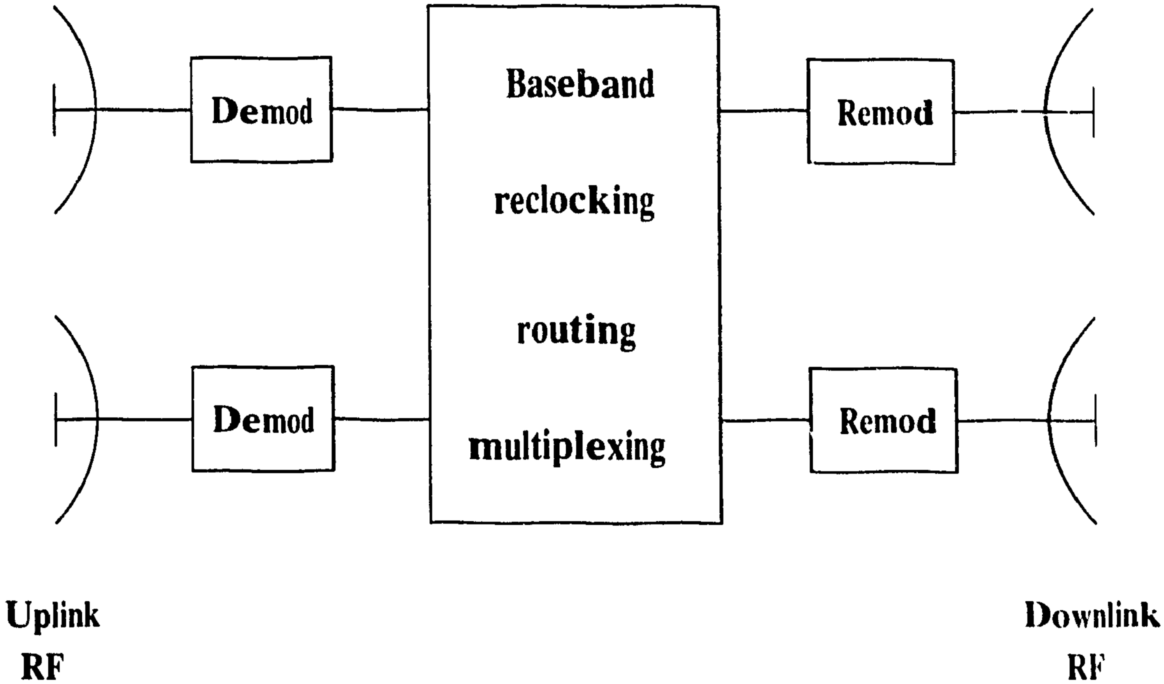


Figure 1.11: Generic satellite on-board processor.

and transmitters that amplify and retransmit the incoming signals.

At L-band, (1-2 GHz), where the maritime systems operate, transistor amplifiers can be used to generate 50 W of output power, and costs are not so high as at 14 or 30 GHz, where the traveling wave tube is the main high power amplifier (HPA) in use [6]. The receiver-transmitter units in satellites are known as transponders. A satellite transponder relays an uplink or forward link electromagnetic field to a downlink, or a return link. Signals (known as carriers) transmitted by an earth station are received at the satellite by either a zone beam or a spot beam antenna. Fig. 1.10 shows a typical single-conversion transponder. This transponder is of type bent-pipe and is currently used in most satellites. On-board processing (Fig. 1.11) has created the second generation of transponders where waveform demodulation or remodulation, decoding or recoding, interference reduction, and so on, are carried out at the satellite instead of on the ground. An on-board processing satellite seperates the uplink and downlink, hence reduces the link error rate P_b [8]:

$$P_b = P_{b,u} + P_{b,d} - 2P_{b,u}P_{b,d} \approx P_{b,u} + P_{b,d} \tag{1.8}$$

where $P_{b,u}$ and $P_{b,d}$ are uplink and downlink error rates, respectively.

1.2.2 Satellite Links

Satellite Uplink

Transmitter power for earth stations is generally provided by high-powered amplifiers, such as TWTs and Klystrons [7]. Since the amplifier and transmitting antenna are located on the ground, size and weight are not prime considerations, and fairly high transmitter EIRP levels can be achieved. These power levels, together with the transmitting antenna gains, determine the available effective isotropic radiated power (EIRP) for uplink communications.
In the design of satellite uplinks, the beam pattern may often be of more concern than the actual uplink EIRP. Whereas the latter determines the power to the desired satellite, the shape of the pattern determines the amount of sidelobe interference power impinging on nearby satellites. The beam pattern therefore establishes an acceptable satellite spacing, and thus the number of satellites that can simultaneously be placed in a given orbit with a specified amount of communication interference.

Satellite Downlink

A satellite downlink is constrained by the fact that the power amplifier and transmitting antenna must be spaceborn. This limits the output power capability that is dependent on the satellite carrier frequency. The spacecraft antenna, while similarly limited in size, must use beam patterns that provide the required coverage area on Earth. Knowing that the coverage area for a specified minimal viewing elevation angle depends only on the satellite altitude, hence the satellite downlink beamwidth for a given coverage area is automatically selected as soon as the satellite orbit altitude is selected.

1.3 Scope of the Thesis

This thesis describes a DS/CDMA mobile network employing orthogonal short concatenated codes for voice signals.

In chapter two, a number of definitions and basic crosscorrelation properties for pseudorandom and related sequences have been presented. Several results are derived and shown to be important for selecting signature code sequences for efficient CDMA network.

In chapter three, a DS/CDMA system employing orthogonal concatenated short codes is proposed and described. We provide a brief description of certain spread spectrum systems. We present different models of land mobile satellite channels and we discuss the need for power control and the representation of power control errors in mobile communications. We conclude the chapter by a statistical evaluation of data-bit block errors.

In chapter four, the results of simulating different cases of long and short concatenated codes are shown. The performance is evaluated in terms of bit error rate and burst error and error-free gap distributions.

Bibliography

[1] R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, *"Theory of spread-spectrum communications- A tutorial", IEEE Trans. Comm., vol. COM-30, pp. 855-884, May 1982.*

[2] William C. Y. Lee, *"Overview of cellular CDMA", IEEE Trans. Veh. Technol., vol. 40, no. 2, pp. 291-302, May 1991.*

[3] K. S. Gilhousen, I. M. Jacobs, R. Padovani, A. J. Viterbi, L. A. Weaver, and C. E. Wheatley, *"On the capacity of a cellular CDMA system", IEEE Trans. Veh. Technol., vol. 40, no. 2, pp. 303-312, May 1991.*

[4] K. S. Gilhousen, I. M. Jacobs, R. Padovani, and L. A. Weaver, *"Increased capacity using CDMA for mobile satellite communications", IEEE JSAC, vol. 8, pp. 503-514, May 1990.*

[5] J. G. Proakis, *"Digital Communications", McGRAW-HILL, 1989.*

[6] T. Pratt and C. W. Bostian, *"Satellite Communications", JOHN WILEY & SONS, 1986.*

[7] R. M. Gagliardi, *"Satellite Communications", Second Edition, VAN NOSTRAND REINHOLD, 1991.*

[8] Tri T. Ha, *"Digital Satellite Communications", Second Edition, McGRAW-HILL, 1990.*

Chapter 2

Signature Sequences Properties

In a direct sequence CDMA (DS/CDMA) system, a set of pseudonoise (PN) codes defines the signal channels, one code per channel. A DS/CDMA system suffers a receiver correlation loss resulting from crosscorrelation interference within the set of codes. This loss quantifies the difficulty the receiver has in distinguishing one code from another. In this chapter, a number of definitions and basic crosscorrelation properties for pseudorandom and related sequences have been presented. Several results are derived and shown to be important for selecting signature code sequences for efficient CDMA network.

2.1 Definitions and Basic Properties

Let C denote the set of complex numbers, and C^N the set of all vectors with N complex components. Elements of C^N are denoted by x, y, z, etc., where $x = (x_0, x_1, \ldots, x_{N-1})$ with $x_i \in C$ for $0 \leq i < N$. The inner product $< x, y >$ of the vectors x and y is defined by $< x, y >= x_0 y_0^* + x_1 y_1^* + \cdots + x_{N-1} y_{N-1}^*$ where a^* denotes the complex conjugate of a. Note that $< x, x >$ is a positive real number for all nonzero $x \in C^N$. The norm $\|x\|$ of x is the positive square root of $< x, x >$, and $\sum x$ denotes $x_0 + x_1 + \cdots + x_{N-1}$.

Let T denote the operator which shifts vectors cyclically to the left by one place , that is $Tx = (x_1, x_2, \ldots, x_{N-1}, x_0)$. If T is applied k times to x, we have $T^k x =$

$(x_k, x_{k+1}, \ldots, x_{N-1}, x_0, x_1, \ldots, x_{k-1})$ for $0 \leq k < N$, while $T^N x = x$. For larger values of k, $T^k x = T^{k'} x$ where $k' \equiv k \bmod N$. Similarly the operator T^{-1} shifts vectors cyclically to the right by one place, and it is easy to see that $T^{-k}x = T^{N-k}x$ for $0 \leq k < N$, and $T^{-N}x = x$. We also have that $\|T^k x\| = \|x\|$ and $\sum(T^k x) = \sum x$. Although $T^i x \neq T^j x$ for $0 \leq i < j < N$, the vectors $x, Tx, T^2x, \ldots, T^{N-1}x$, which are called phases of x, are cyclically equivalent; that is, they are cyclic shifts of each other.

If $x = (x_0, x_1, \ldots, x_{N-1})$ is a vector of length N, the reverse of x is the vector

$$w = (x_{N-1}, \ldots, x_1, x_0)$$

That is, $w_i = x_{N-1-i}$ for $0 \leq i \leq N-1$. The sequence w which is generated by the vector w is called the reverse sequence for x. For each k in the range $0 \leq k \leq N-1$, we have

$$(T^k w)_i = (T^{-k}x)_{N-1-i} = (T^{N-k}x)_{N-1-i}$$

so that each phase of w is the reverse of some phase of x.

Let $\mathcal{Z}$ denote the set of all integers. For vectors x and y of length N we define the periodic crosscorrelation function $\theta x, y(.)$ by

$$\theta_{x,y}(l) = < x, T^l y > \qquad , l \in \mathcal{Z} \tag{2.1}$$

If x and y are the sequences generated by x and y, respectively, then (2.1) is equivalent to

$$\theta_{x,y}(l) = \sum_{i=0}^{N-1} x_i y_{i+l}^* \qquad , l \in \mathcal{Z} \tag{2.2}$$

It is easy to verify that for each $l \in \mathcal{Z}$,

$$\theta_{x,y}(l+N) = \theta_{x,y}(l) \tag{2.3}$$

and

$$\theta_{x,y}(-l) = [\theta_{x,y}(l)]^* \tag{2.4}$$

Applying the Cauchy inequality $| < u, v > | \leq \|u\|.\|v\|$ to (2.1) we get

$$\begin{aligned} |\theta_{x,y}(l)| &= | < x, T^l y > | \\ &\leq \|x\|.\|T^l y\| \\ &= \|x\|.\|y\| \end{aligned}$$

Hence,

$$\|\theta_{x,y}(l)\| \leq \|x\|.\|y\| \tag{2.5}$$

The periodic autocorrelation function for the sequence x is just

$$\theta_{x,x}(l) = \theta_x(l) =< x, T^l x > \tag{2.6}$$

Notice that $\theta_x(0) =< x, x >$ is a positive real number, except when $x = 0$. From (2.1)-(2.5), we see that for each $l \in \mathcal{Z}$, $\theta_x(l) = \theta_x(l+N)$, $\theta_x(-l) = [\theta_x(l)]^*$, and

$$|\theta_x(l)| \leq \|x\|^2 =< x, x >= \theta_x(0) \tag{2.7}$$

Two sequences x and y are said to be uncorrelated if $\theta_{x,y}(l) = 0$ for all l, and a sequence x of period M is said to have a two-valued autocorrelation function if $\theta_x(l)$ equals some constant other than $\theta_x(0)$ for all $l \neq 0 \bmod M$. Since $\theta_x(l) = [\theta_x(-l)]^*$, this constant must be real.

2.1.1 Correlation Identities

Let $w, x, y, z \in \mathcal{C}^N$ and let w, x, y, z be the corresponding sequences. The four cross-correlation functions $\theta_{w,x}$, $\theta_{y,z}$, $\theta_{w,y}$, and $\theta_{x,z}$ are related through the following identity

$$\sum_{l=0}^{N-1} \theta_{w,y}(l)[\theta_{x,z}(l+n)]^* = \sum_{l=0}^{N-1} \theta_{w,x}(l)[\theta_{y,z}(l+n)]^* \tag{2.8}$$

Equation (2.8), which is a generalization of a result in [1], can be derived as follows. We first note that the right-hand side can be rewritten as

$$\begin{aligned}
&\sum_{l=0}^{N-1} \theta_{w,x}(l)[\theta_{y,z}(l+n)]^* \\
&= \sum_{l=0}^{N-1} [\sum_{i=0}^{N-1} w_i x_{i+l}^*][\sum_{j=0}^{N-1} y_j z_{j+l+n}^*]^* \\
&= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} w_i y_j^* [\theta_{x,z}(j+n-i)]^* \\
&= \sum_{i=0}^{N-1} \sum_{l=-i}^{-i+N-1} w_i y_{i+l}^* [\theta_{x,z}(l+n)]^* \\
&= \sum_{l=-i}^{-i+N-1} \sum_{i=0}^{N-1} w_i y_{i+l}^* [\theta_{x,z}(l+n)]^*
\end{aligned}$$

By again interchanging the order of summation, we see that the last expression is just the left-hand side of (2.8). For the special case $w = x$ and $y = z$, (2.8) reduces to

$$\sum_{l=0}^{N-1} \theta_{x,y}(l)[\theta_{x,y}(l+n)]^* = \sum_{l=0}^{N-1} \theta_x(l)[\theta_y(l+n)]^* \tag{2.9}$$

and setting $n = 0$ in (2.9) gives

$$\sum_{l=0}^{N-1} |\theta_{x,y}(l)|^2 = \sum_{l=0}^{N-1} \theta_x(l)[\theta_y(l)]^* \tag{2.10}$$

These identities provide a remarkable variety of bounds, computational techniques, and sequence construction techniques. As an example of sequence construction techniques, consider the following. For given x and y, (2.2) defines a mapping from $\mathcal{Z}$ into $\mathcal{C}$, that is, a complex-valued sequence (in most cases of interest, x and y sequences take on values ± 1). From (2.3), it is clear that this sequence (which we may denote by $\theta_{x,y}$) is periodic. We can thus interpret (2.8) to mean that the crosscorrelation function for the sequences $\theta_{w,y}$ and $\theta_{x,z}$ equals the crosscorrelation function for the sequences $\theta_{w,x}$ and $\theta_{y,z}$. Now, if w and x are uncorrelated sequences, (2.8) implies that $\theta_{w,y}$ and $\theta_{x,z}$ are also uncorrelated sequences for any other sequences y and z. Thus, starting from two uncorrelated sequences, we can produce two new uncorrelated sequences.

Turning to (2.9), we interpret this identity to mean that the autocorrelation function for the sequence $\theta_{x,y}$ equals the crosscorrelation function for the sequences θ_x and θ_y.

2.2 Binary Maximal-Length Sequences

2.2.1 Properties of Binary Shift-Register Sequences

Let $h(x) = h_0x^n + h_1x^{n-1} + \cdots + h_{n-1}x + h_n$ denote a binary primitive polynomial of degree n where $h_0 = h_n = 1$ and the other h_i's take on values 0 and 1. A binary sequence u is said to be a sequence generated by $h(x)$ if for all integers j

$$h_0u_j \oplus h_1u_{j-1} \oplus h_2u_{j-2} \oplus \cdots \oplus h_nu_{j-n} = 0 \tag{2.11}$$

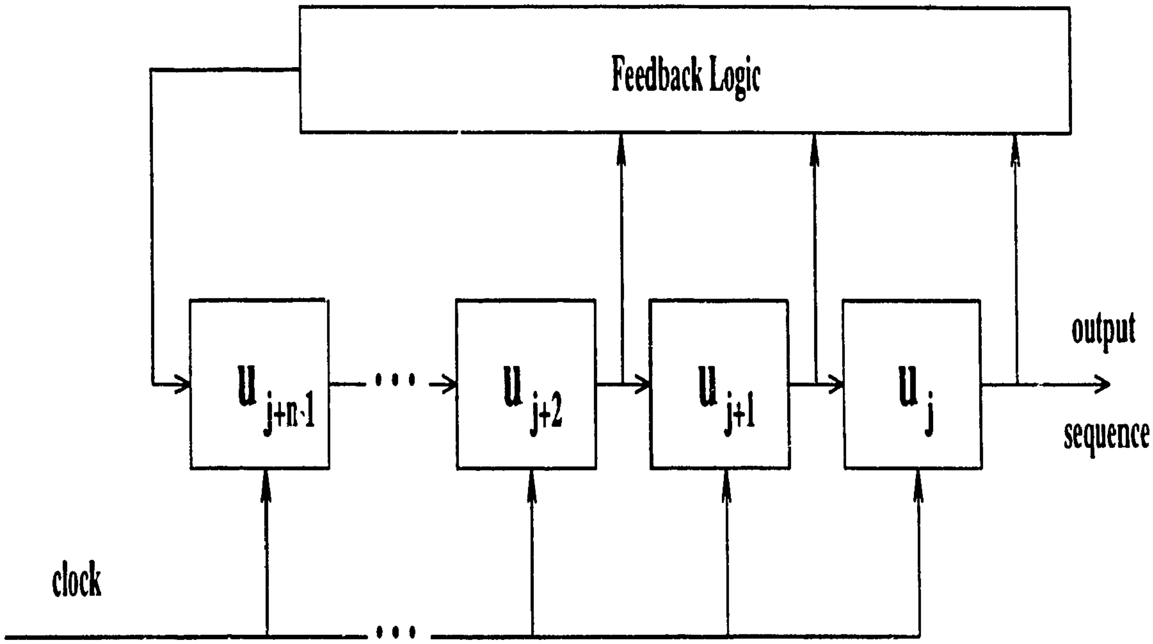


Figure 2.1: Generic maximal-length linear feedback shift register.

Here $\oplus$ denotes addition modulo 2. Replacing j by $j+n$ in (2.11), and using the fact that $h_0 = 1$, we obtain

$$u_{j+n} = h_n u_j \oplus h_{n-1} u_{j+1} \oplus \cdots \oplus h_1 u_{j+n-1} \tag{2.12}$$

From this it follows that the sequence u can be generated by an n-stage binary linear feedback shift register which has a feedback tap connected to the ith cell if $h_i = 1$, where $0 < i \leq n$. Since $h_n = 1$, there is always such a connection for the nth cell. Fig. 2.1 shows a generic maximal-length linear feedback shift register.

A shift register can generate several different sequences, one of which is the all-zeros sequence. Of course, only the nonzero sequences are of interest and henceforth we shall reserve u to denote a nonzero solution to (2.11).

The following properties of shift register sequences are well known. The period of u is $N = 2^n - 1$, where n is the number of cells in the shift register; there are exactly N nonzero sequences generated by $h(x)$, and they are just the N different phases of u; namely $u, Tu, T^2u, \ldots, T^{N-1}u$. Similarly, if u and v are generated by $h(x)$, then so is $u \oplus v$, where $u \oplus v$ denotes the sequence whose ith element is $u_i \oplus v_i$.

In most practical applications, a binary sequence is actually transmitted as a sequence of unit amplitude, positive and negative pulses. Let us introduce the function $X(.)$ defined by $X(0) = +1$ and $X(1) = -1$ or equivalently $X(\alpha) = (-1)^\alpha$ for $\alpha \in \{0, 1\}$. If u denotes an arbitrary $\{0, 1\}$-valued sequence, then $X(u)$ denotes the corrresponding $\{+1, -1\}$-valued sequence, where the ith element of $X(u)$ is just $X(u_i)$.

Clearly, if u is of period N, then so is $X(u)$. Notice that $T^i(X(u)) = X(T^i u)$ and

$$\begin{aligned} \sum X(u) &= X(u_0) + X(u_1) + \cdots + X(u_{N-1}) \\ &= N - 2wt(u) \end{aligned} \quad (2.13)$$

where $wt(u)$ denotes the hamming weight of u, that is, the number of ones in u. If we define the periodic crosscorrelation function $\theta_{u,v}(.)$ to be equal to $\theta_{X(u),X(v)}(.)$, then we have

$$\begin{aligned} \theta_{u,v}(l) \doteq \theta_{X(u),X(v)}(l) &= \sum_{i=0}^{N-1} X(u_i)X(v_{i+l}) \\ &= \sum_{i=0}^{N-1} (-1)^{u_i}(-1)^{v_{i+l}} \\ &= \sum_{i=0}^{N-1} (-1)^{u_i \oplus v_{i+l}} \\ &= \sum_{i=0}^{N-1} X(u_i \oplus v_{i+l}) \end{aligned} \quad (2.14)$$

Applying (2.13), we get

$$\theta_{u,v}(l) = N - 2wt(u \oplus T^l v) \quad (2.15)$$

The periodic autocorrelation function $\theta_u(.)$ is just $\theta_{u,u}(.)$ and we have

$$\theta_u(l) = N - 2wt(u \oplus T^l u) \quad (2.16)$$

Note that $\theta_{u,v}(l)$ is always an integer, and this integer is odd (even) if N is odd (even).

2.3 Sets of Binary Sequences with Small Cross-correlation

2.3.1 Gold Sequences

A preferred pair of m-sequences (maximal-length sequences) is a part of m-sequences of period $N = 2^n - 1$ which has the preferred three-valued crosscorrelation function. The values taken on by the preferred three-valued crosscorrelation function are -1, $-t(n)$, and $t(n) - 2$, where

$$\begin{aligned} t(n) &= 1 + 2^{(n+2)/2} \quad &, n \text{ even} \\ &= 1 + 2^{(n+1)/2} \quad &, n \text{ odd} \end{aligned} \tag{2.17}$$

The pair of primitive polynomials that generate a preferred pair of m-sequences is called a preferred pair of polynomials.

For a set Ξ of periodic sequences, we define θ_c, peak crosscorrelation magnitude, by

$$\theta_c = max\{|\theta_{x,y}(l)| : 0 \leq l \leq N-1, \quad x \in \Xi, \quad y \in \Xi, \quad x \neq y\} \tag{2.18}$$

and θ_a, peak out-of-phase autocorrelation magnitude, by

$$\theta_a = max\{|\theta_x(l)| : 1 \leq l \leq N-1, \quad x \in \Xi\} \tag{2.19}$$

One important class of periodic sequences which provides larger sets of sequences with good periodic crosscorrelation is the class of Gold sequences. A set of Gold sequences of period $N = 2^n - 1$, consists of $N+2$ sequences for which $\theta_c = \theta_a = t(n)$. A set of Gold sequences can be constructed from appropriately selected m-sequences as described below.

Suppose a shift register polynomial $f(x)$ factors into $h(x).\hat{h}(x)$ where $h(x)$ and $\hat{h}(x)$ have no factors in common. Then the set of all sequences generated by $f(x)$ is just the set of all sequences of the form $a \oplus b$ where a is some sequence generated by $h(x)$, b is some sequence generated by $\hat{h}(x)$, and we do not make the usual restriction that a and b are nonzero sequences [2], [3].

Now suppose that $h(x)$ and $\hat{h}(x)$ are two different primitive binary polynomials of degree n that generate the m-sequences u and v, respectively, of period $N = 2^n - 1$. If y denotes a nonzero sequence generated by $f(x) = h(x).\hat{h}(x)$, then, from the above and the property that we can generate different phases of an m-sequence by using the same primitive polynomial we get that either

$$y = T^i u \tag{2.20}$$

or

$$y = T^j v \tag{2.21}$$

or

$$y = T^i u \oplus T^j v \tag{2.22}$$

where $0 \leq i, j \leq N - 1$. From this it follows that y is some phase of some sequence in the set $G(u, v)$ defined by

$$G(u,v) = \{u, v, u \oplus v, u \oplus Tv, u \oplus T^2 v, \ldots, u \oplus T^{N-1} v\} \tag{2.23}$$

We note $G(u, v)$ contains $N + 2 = 2^n + 1$ sequences of period N. The peak cross-correlation parameters θ_c and θ_a for $G(u, v)$ satisfy

$$\theta_c = \theta_a = max\{|\theta_{u,v}(l)| : 0 \leq l \leq N - 1\} \tag{2.24}$$

In other words, given a pair of m-sequences u and v with peak periodic crosscorrelation magnitude M, we can construct a set of $N + 2$ sequences with peak periodic crosscorrelation magnitude and peak out-of-phase periodic autocorrelation magnitude equal to M. Let us compare now the parameter $\theta_{max} = max\{\theta_a, \theta_c\}$ for a set of Gold sequences to a bound due to Sidelnikov [4] which states that for any set of N or more binary sequences of period N

$$\theta_{max} > (2N - 2)^{1/2} \tag{2.25}$$

For $N = 2^n - 1$, equation (2.25) implies that

$$\theta_{max} > -1 + 2^{(n+1)/2} \tag{2.26}$$

when n is odd, the right-hand side of (2.26) equals $t(n) - 2$. Since θ_{max} must be an odd integer, we obtain that for any set of N or more binary sequences of period $N = 2^n - 1$, n odd,

$$\theta_{max} \geq t(n) \tag{2.27}$$

Since $\theta_{max} = t(n)$ for Gold sequences, we conclude that they form an optimal set with respect to the bounds (2.25) and (2.27) when n is odd. When n is even, the right-hand side of (2.26) is not an integer, and is smaller than $t(n)$ by a factor of approximately $\sqrt{2}$. Although the bound itself probably is weak, Gold sequences are not optimal in this case.

2.3.2 Kasami Sequences

Let q denote a positive integer, and consider the sequence v formed by taking every qth bit of u (i.e., $v_i = u_{qi}$ for all $i \in \mathcal{Z}$), u is a nonzero sequence generated by $h(x)$. The sequence v is said to be a decimation by q of u, and will be denoted by $u[q]$. Let $gcd(a, b)$ denote the greatest common divisor of the integers a and b. A property of m-sequence arises for $u[q]$, assuming $u[q]$ is not identically zero, then, $u[q]$ has period $N/gcd(N, q)$, and is generated by the polynomial $\hat{h}(x)$ whose roots are the qth powers of the roots of $h(x)$.

Now, let n be an even integer and let u denote an m-sequence of period $N = 2^n - 1$ generated by $h(x)$. Consider the sequence $w = u[s(n)] = u[2^{n/2} + 1]$. It follows from the property above that w is a sequence of period $2^{n/2} - 1$ which is generated by the polynomial $h'(x)$ whose roots are the $s(n)$th powers of the roots of $h(x)$. Furthermore, since $h'(x)$ can be shown to be a polynomial of degree $n/2$, w is an m-sequence of period $2^{n/2} - 1$.

2.3.3 TCH Codes

The new class of TCH [5] (Tomlinson, Cercas, Hughes) codes was found having in mind an easy implementation of the receiver for all types of communications. TCH codes are suitable not only for this kind of applications but also for use in CDMA, where its code words can be used as PN sequences given their good properties of auto and cross-correlation. TCH codes are binary nonlinear cyclic codes of length $n = 2^m$. The linear addition modulo 2 of two code words does not necessarily produce another valid code word, however the cyclic shift of any code word, as well as its inverse in the sense of a binary complement, is always a valid code word. A TCH code is then a binary nonlinear block code closed under cyclic shifting.

TCH codes can be defined in terms of h code polynomials, $P_i(x)_{i=1 to h}$, where $P_i(x) \neq P_j(x^r) \bmod n$, $i \neq j$, for all time shifts r. The number of information bits k of a TCH(n, k, t) code, able to correct t errors, or simply TCH(n, k), is $k = m + \log_2 h + 1$ where the term 1 accounts for including the inverses of all code words. Although the ratio k/n of TCH codes found so far is relatively small, it is shown [6, 7] that low-rate coding using TCH codes can have significant advantages.

The great number of TCH codes found of length 256 and 512 (namely 28 codes TCH(256, 16)) with good auto and cross-correlation properties turned our attention for its possible use in CDMA, which is sometimes limited by the fact that the number of known good sequences is relatively small. The autocorrelation function of TCH codes is periodic and shows a very clear peak when the sequences are superimposed and is either zero or negative for all other possible bit displacements, as in maximal-length sequences.

2.4 Aperiodic Correlation Functions for Complex-Valued Sequences

Consider a binary communications system in which the binary data sequence $b_n = \ldots, b_{-1}, b_0, \ldots$ is an arbitrary sequence of +1's and -1's. The binary signature sequence y is generated by a vector $y = (y_0, y_1, \ldots, y_{N-1})$ where each y_i is either +1 or -1. The sequence y can be written as

$$\begin{aligned} y &= \ldots; y_0, y_1, \ldots, y_{N-1}; y_N, y_{N+1}, \ldots, y_{2N-1}; \ldots \\ &= \ldots; y; y; \ldots \end{aligned} \tag{2.28}$$

The data and signature sequences are combined to give

$$\hat{y} = \ldots ; b_{-1}y; b_0y; b_1y; \ldots \tag{2.29}$$

In other words, $\hat{y}$ is the sequence which has as its ith element $\hat{y}_i = b_n y_k$ for all i such that $i = nN + k$ for k in the range $0 \leq k \leq N - 1$.

Consider for the moment an idealized binary system without channel noise and with only one transmitted signal. The data symbol b_n is transmitted as $\hat{y}_n = b_n y$. A synchronous correlation receiver forms the inner product $< \hat{y}_n, y >$, where $\hat{y}_n$ represents the received vector in this noiseless system and y is the signature sequence which is stored at the receiver. We have

$$< \hat{y}_n, y >= b_n < y, y >= b_n \theta_y(0) \tag{2.30}$$

Thus, in the absence of noise, the data sequence is recovered at the output of the correlation receiver.

Now suppose that there are two transmitters in this idealized system and that we are interested in receiving the data b_n in the presence of another signal. This other signal is a sequence $\hat{x}$ which is formed from the data sequence b'_n and the signature sequence x (generated by a binary vector $x = (x_0, x_1, \ldots, x_{N-1})$) in exactly the same manner as $\hat{y}$ was formed from b_n and y. The two transmitters are not required to be synchronized; moreover, there may be different transmission delays for the two signals. If a correlation receiver is synchronized to the sequence y, and if sequence x is delayed by an amount l $(1 \leq l \leq N - 1)$ relative to y, then for a noiseless additive channel the received sequence is $\hat{y} + T^{-l}\hat{x}$ where

$$\hat{y} = \ldots ; b_{-1}y; b_0y; b_1y; \ldots \tag{2.31}$$

and

$$\hat{x} = \ldots ; b'_{-1}x; b'_0x; b'_1x; \ldots \tag{2.32}$$

Thus the output of a correlation receiver which is in synchronism with y is given by

$$z_n =< \hat{y}_n, y > +[b'_{n-1} \sum_{i=0}^{l-1} x_{N-l+i}y_i + b'_n \sum_{i=l}^{N-1} x_{i-l}y_i] \tag{2.33}$$

The first term represents the desired signal and the term in brackets represents the interference due to the presence of another transmited signal in the channel. Equation (2.33) can be rewritten as

$$z_n = b_n\theta_y(0) + [b'_{n-1}C_{x,y}(l-N) + b'_nC_{x,y}(l)] \tag{2.34}$$

where, for complex-valued sequences x and y, the aperiodic crosscorrelation function C_{xy} is defined as [14]

$$C_{x,y}(l) = \begin{cases} \sum_{j=0}^{N-1-l} x_j y^*_{j+l} & 0 \le l \le N-1 \\ \sum_{j=0}^{N-1+l} x_{j-l} y^*_j & 1-N \le l < 0 \\ 0 & |l| \ge N \end{cases} \tag{2.35}$$

For $l \neq 0$, it is desirable that the magnitude of the term in brackets in (2.34) be small compared with N to permit reliable baud synchronization and also, in certain applications, to reject multipath delayed version of the transmitted signal (as will be shown later).

Thus we are motivated to consider the two correlation functions

$$\theta_{x,y}(l) = C_{x,y}(l) + C_{x,y}(l-N) \tag{2.36}$$

which is the usual periodic crosscorrelation function defined before for complex-valued sequences x and y, and

$$\hat{\theta}_{x,y}(l) = C_{x,y}(l) - C_{x,y}(l-N) \tag{2.37}$$

which is the odd crosscorrelation function for complex-valued sequences x and y. Notice that if $b'_{n-1} = b'_n$ in (2.34) then

$$z_n = b_n\theta_y(0) + b'_n\theta_{x,y}(l) \tag{2.38}$$

on the other hand, if $b'_{n-1} = -b'_n$ then

$$z_n = b_n\theta_y(0) + b'_n\hat{\theta}_{x,y}(l) \tag{2.39}$$

we note that if b'_n is a sequence of independent, identically distributed, binary random variables, then $P(b'_{n-1} \neq b'_n) = \frac{1}{2}$ so that the situation which results in

(2.39) occurs one half of the time on the average. The conclusion is that for spread spectrum multiple-access systems, the odd crosscorrelation function is as important as the periodic crosscorrelation function.

The name "odd crosscorrelation function" is appropriate because the function $\hat{\theta}_{x,y}$ is such that

$$\begin{aligned}\hat{\theta}_{x,y}(N-l) &= C_{x,y}(N-l) - C_{x,y}(-l)\\ &= [C_{y,x}(l-N) - C_{y,x}(l)]^{*}\\ &= -[\hat{\theta}_{y,x}(l)]^{*}\end{aligned} \quad (2.40)$$

This is in contrast to the periodic (or even) crosscorrelation function which satisfies

$$\begin{aligned}\theta_{x,y}(N-l) &= C_{x,y}(N-l) + C_{x,y}(-l)\\ &= [C_{y,x}(l-N) + C_{y,x}(l)]^{*}\\ &= [\theta_{y,x}(l)]^{*}\end{aligned} \quad (2.41)$$

for $0 \leq l < N$.

The importance of the odd autocorrelation function $\hat{\theta}_x$ can be illustrated by a simple example. Suppose that the sequences x and y in this example are the same and that $b_k = b'_k$ for all k. Then the quantity z_n in (2.33) is the output of a correlation receiver which is in synchronism with y when the input is $\hat{y} + T^{-l}\hat{y}$. More generally, in certain multipath channels, the received signal may consist of a desired signal component plus a delayed attenuated version of the desired signal component, such as $\hat{y} + \beta T^{-l}\hat{y}$ where $|\beta| \leq 1$.

If $x = y$, $b_k = b'_k$ for all k, and the received signal is $\hat{y} + \beta T^{-l}\hat{y}$, then (2.34) becomes

$$z_n = b_n\theta_y(0) + \beta[b_{n-1}C_y(l-N) + b_nC_y(l)] \quad (2.42)$$

if $b_n = b_{n-1}$, then

$$z_n = b_n[\theta_y(0) + \beta\theta_y(l)] \quad (2.43)$$

On the other hand, if $b_n = -b_{n-1}$, then (2.42) becomes

$$z_n = b_n[\theta_y(0) + \beta\hat{\theta}_y(l)] \quad (2.44)$$

It is clear that for certain applications the odd correlation function is as important as the periodic correlation function.

Another situation in which the odd autocorrelation function plays a major role is in the acquisition by the receiver of the epoch of the signature sequence from the received signal. Here the odd autocorrelation function comes into play even if there is no interfering signal present at the receiver. For such applications, the sequences should be selected such that $\theta_y(l)$ and $\hat{\theta}_y(l)$ are both small for $0 < l < N$.

Now, let us consider the maximum values of the magnitudes of the sidelobes of the periodic and odd autocorrelation functions (the sidelobes are the out-of-phase autocorrelations). For a given sequence x, let $M(x)$ and $\hat{M}(x)$ denote the maximum sidelobe magnitudes for the periodic and odd autocorrelation, respectively; that is,

$$M(x) = max\{|\theta_x(l)| : 1 \leq l < N\} \tag{2.45}$$

and

$$\hat{M}(x) = max\{|\hat{\theta}_x(l)| : 1 \leq l < N\} \tag{2.46}$$

It is important to notice that $l = 0$ is excluded in (2.45) and (2.46).

Notice also that $M(x) \leq \theta_x(0)$ and $\hat{M}(x) \leq \hat{\theta}_x(0)$. For a set Ξ of sequences, it is convenient to define parameters θ_a and $\hat{\theta}_a$ as measures of the maximum periodic and odd autocorrelation sidelobes and θ_c and $\hat{\theta}_c$ as measures of the maximum periodic and odd crosscorrelation magnitudes, where the maximum is taken over all sequences in the set

$$\begin{aligned}
\theta_a &= max\{M(x) : x \in \Xi\} \\
&= max\{|\theta_x(l)| : 1 \leq l \leq N-1, \;\; x \in \Xi\}, \\
\hat{\theta}_a &= max\{\hat{M}(x) : x \in \Xi\} \\
&= max\{|\hat{\theta}_x(l)| : 1 \leq l \leq N-1, \;\; x \in \Xi\}, \\
\theta_c &= max\{|\theta_{x,y}(l)| : 0 \leq l \leq N-1, \;\; x \in \Xi, \;\; y \in \Xi, \;\; x \neq y\}, \\
\hat{\theta}_c &= max\{|\hat{\theta}_{x,y}(l)| : 0 \leq l \leq N-1, \;\; x \in \Xi, \;\; y \in \Xi, \;\; x \neq y\}, \\
\theta_{max} &\doteq max\{\theta_a, \theta_c\}
\end{aligned} \tag{2.47}$$

and we define C_a and C_c as measures of the maximum aperiodic autocorrelation and crosscorrelation magnitudes, respectively, where the maximum is taken over all sequences in the set

$$\begin{aligned} C_a &= max\{|C_x(l)| : 1 \le l \le N-1,\ \ x \in \Xi\}, \\ C_c &= max\{|C_{x,y}(l)| : 0 \le l \le N-1,\ \ x \in \Xi,\ \ y \in \Xi,\ \ x \neq y\}, \\ C_{max} &\doteq max\{C_a, C_c\} \end{aligned} \tag{2.48}$$

For certain applications such as the simple multipath communication example that led to (2.44), it is desirable to make the odd autocorrelation sidelobes as small as possible.

One common method for selecting code sequences for phase-coded spread-spectrum multiple access (SSMA) systems is to search a family of sequences for which θ_c is relatively small (e.g., the Gold sequences [8], [9]) in hope of finding a sub-family for which $\hat{\theta}_c$ is also small. This is basically the approach of Massey and Uhran [11], who consider a family of sequences which consists of all distinct shifts of a set of K sequences of period N for which θ_c is small. The goal is to minimize the parameter $\hat{\theta}_c$ by finding the best shift for each of the K sequences in the set. Massey and Uhran discuss the role of the odd autocorrelation function $\hat{\theta}_x$ in the analysis and design of phase-coded SSMA systems for multipath channels and the minimization of the odd autocorrelation parameter $\hat{\theta}_a$ for a set of sequences for which θ_a is small. Although θ_a does not change if the sequences are shifted in time [11], Massey and Uhran found that $\hat{\theta}_a$ is very sensitive to such shifts. In particular, they discovered that there is a unique optimum shift for most of the 18 different binary maximal-length shift register sequences (m-sequences) [10] of period 127 (see also [12]).

By choosing an optimum shift for each of these sequences, one obtains a set of 18 sequences for which $\hat{\theta}_a = 19$ and, as is true for any set of m-sequences, $\theta_a = 1$. The crosscorrelation functions for these sequences satisfy $\theta_c = 41$. Of course, some of the parameters can be improved (perhaps at the expense of others) if a smaller set of sequences is acceptable. For instance, there is a subset consisting of 14 of these m-sequences with $\hat{\theta}_a = 17$ (see [11] or [12]). Also, there exist sets (known as maximal connected sets) of m-sequences of period 127 which contain six sequences and have periodic crosscorrelation $\theta_c = 17$.

The odd crosscorrelation functions were examined in [12] for some of those maximal connected sets. One maximal connected set is given for which $\hat{\theta}_a = 23$ and $\hat{\theta}_c = 29$, and hence $\hat{\theta}_{max} = 29$.

It is of interest to obtain lower bounds on these various correlation parameters in order to have a standard against which to compare the parameters for a set of sequences. Such bounds were established by Welch [13] for the parameters θ_{max} and C_{max}. Welch proved that for a set of K complex-valued sequences of period N,

$$\theta_{max} \geq N[(K-1)/(NK-1)]^{1/2} \tag{2.49}$$

and

$$C_{max} \geq N[(K-1)/(2NK-K-1)]^{1/2} \tag{2.50}$$

For large values of K and N the lower bound on θ_{max} is approximately $\sqrt{N}$ and the lower bound on C_{max} is approximately $\sqrt{N/2}$.

For the $K = 2^n + 1$ Gold sequences [8] of period $N = 2^n - 1$, the lower bound on θ_{max} from (2.48) is approximately $2^{n/2}$ for large n. The actual value of θ_{max} is $2^{\lfloor (n+2)/2 \rfloor} + 1$, where $\lfloor r \rfloor$ denotes the integer part of the real number r. Thus, for Gold codes θ_{max} is greater than the bound by a factor of $\sqrt{2}$ for odd values of n and by a factor of 2 for even values.

Fukumasa *et al.* [15] proposed and investigated a method for designing complex-valued polyphase pseudo-noise (PN) sequences having both good even and odd correlation properties, which are important for acquisition and demodulation in SSMA communications. In particular, they proposed a method of designing equivalent odd and even correlation (EOE) sequences of which the absolute values of the odd and even correlation functions at each shift are equal, that is,

$$\begin{aligned} |\hat{\theta}_x(l)| &= |\theta_x(l)|, \\ |\hat{\theta}_y(l)| &= |\theta_y(l)|, \end{aligned}$$

and

$$|\hat{\theta}_{x,y}(l)| = |\theta_{x,y}(l)| \tag{2.51}$$

for every $l \in \{0, 1, \ldots, N-1\}$.

The above proposed method can be stated as:

Let u and v be arbitrary real valued sequences of period N. Then the complex valued sequences, x and y, given by

$$x_n = u_n \exp j(\frac{\pi kn}{2N} + \beta), \qquad (n = 0, 1, \ldots, N-1)$$
$$y_n = v_n \exp j(\frac{\pi kn}{2N} + \beta), \qquad (n = 0, 1, \ldots, N-1)$$

are EOE sequences when k is an arbitrary odd integer, and β is an arbitrary real constant satisfying $(0 \leq \beta < 2\pi)$.

Cochannel interference in the system was expressed by using even, odd and generalized odd crosscorrelation functions. They defined the generalized odd correlation function for a polyphase spread spectrum system with orthogonal polyphase-coded carriers, as

$$\Theta_{x,y}^{(\gamma)}(l) = C_{x,y}(l) + \gamma C_{x,y}(l - N) \qquad (2.52)$$

where $\gamma \in \{1, \exp j2\pi/M, \ldots, \exp j2\pi(M-1)/M\}$, is the data M-ary phase signal. In fact, they found that the derived polyphase sequences have lower peaks of correlations and lower bit error rate than their original sequences.

Matsufiji and Imamura [16] derived from the exact even periodic correlation distribution an approximate equation of the odd periodic correlation distribution for the family of binary sequences. The distribution means the probabilities of correlation values which appear among all the phase-shifted sequences in the family. They have also shown that the odd periodic correlation distribution of the family with optimal periodic correlation properties is not the Gaussian distribution, but that of the family of the Gold sequences with short period seems to be similar to the Gaussian distribution.

2.5 Conclusion

In this chapter, we have discussed the properties of pseudorandom and related sequences, such as binary maximal-length sequences and binary sequences with small crosscorrelation. We have seen that for certain applications the odd correlation function is as important as the periodic correlation function. We have shown a method proposed by Fukumasa *et al.* for designing complex-valued polyphase PN sequences having both good even and odd correlation properties.

In the next chapter, we present the channel encountered in land mobile satellite communications. We discuss the channel effects on signal transmission (e.g. power control) and signal reception (e.g. error burstiness).

Bibliography

[1] Michael B. Pursley and Dilip V. Sarwate, *"Performance Evaluation for Phase-Coded Spread-Spectrum Multiple Access Communication- Part II: Code Sequence Analysis", IEEE Trans. Comm., vol. COM-25, No. 8, pp. 800-803, August 1977.*

[2] N. Zierler, *"Linear Recurring Sequences", J. Soc. Industrial and Applied Mathematics, vol. 7, pp.31-48, 1959.*

[3] N. Zierler and W. H. Mills, *"Products of Linear Recurring Sequences", J. Algebra, vol. 27, pp. 147-157, 1973.*

[4] V. M. Sidelnikov, *"On Mutual Correlation of Sequences", Soviet Math. Dokl.,vol. 12, pp. 197-201, 1971.*

[5] F. A. B. Cercas, M. Tomlinson, and A. A. Albuquerque, *"Theory of spread-spectrum communications- A tutorial", IEEE Trans. Comm., vol. COM-30, pp. 855-884, May 1982.*

[6] M. Tomlinson, F. A. B. Cercas, and E. C. Hughes, *"Aspects of Coding for Power Efficient Satellite VSAT Systems", ESA Journal, vol. 15, pp. 165-185, 1991.*

[7] F. A. B. Cercas, M. Tomlinson, and A. A. Albuquerque, *"TCH : A New Family of Cyclic Codes Length* 2^m*", 1993 IEEE International Symposium on Information Theory,* 17*th to* 22*nd January 1993, Hilton Palacio del Rio Hotel, San Antonio, Texas, USA, p. 198.*

[8] R. Gold, *"Optimal Binary Sequences for Spread Spectrum Multiplexing", IEEE Transactions on Information Theory, vol. IT-13, pp. 619-621, October 1967.*

[9] R. Gold, *"Maximal Recursive Sequences with 3-Valued Recursive Crosscorrelation Functions", IEEE Transactions on Information Theory, vol. IT-14, pp. 154-156, January 1968.*

[10] S. W. Golomb, *"Shift Register Sequences", San Francisco: Holden-Day, 1967.*

[11] J. L. Massey and J. J. Uhran, Jr., *"Sub-Baud Coding", Proceedings of the Thirteenth Annual Allerton Conference on Circuit and System Theory, pp. 539-547, October 1975.*

[12] H. F. A. Roefs and M. B. Pursley, *"Correlation Parameters of Random Sequences and Maximal Length Sequences for Spread-Spectrum Multiple-Access Communications", 1976 IEEE Canadian Communication and Power Conference, pp. 141-143, October 1976.*

[13] L. R. Welch, *"Lower Bounds on the Maximum Crosscorrelation of Signals". IEEE Transactions on Information Theory, vol. IT-20, pp. 397-399, May 1974.*

[14] D. V. Sarwate and M. B. Pursley, *"Correlation Properties of Pseudorandom and Related Sequences", Proc. IEEE, vol. 68, no. 5, May 1980.*

[15] H. Fukumasa, R. Kohno, and H. Imai, *"Design of Pseudonoise Sequences with Good Odd and Even Correlation Properties for DS/CDMA", IEEE Journal on Selected Areas in Communications, vol. 12, no. 5, June 1994.*

[16] S. Matsufuji and K. Imamura, *"Approximate Odd Periodic Correlation Distributions of Binary Sequences", IEICE Trans. Comm., Vol. E76-B, No. 8, August 1993.*

Chapter 3

Orthogonal Short Codes for CDMA Satellite Channels

3.1 Introduction

There has been a growing interest in the use of spread spectrum as a multiaccess technique [1], [2]. These spread spectrum systems defined as having a bandwidth much larger than the data bearing signal have come a long way from being devoted entirely to military communications systems to commercially viable techniques which recently found a wide range of applications. It is natural to expect that many of the properties of these signals are preferred in both military and commercial applications. For example, the low probability of intercept property (favored by the military) translates into the overlay possibility on existing domestic services. The selective identification property means automatic and instant identification of sender or destination, an important asset in mobile networking.

These systems also have greater immunity to multipath fading. The multipath of a certain slot transmission in a TDMA system can spill into neighboring bits and slots, causing massive destruction to data bits unless expensive equalizers are built and trained and/or large gaps left between slots, leading to waste of precious capacity.

In spread spectrum systems, recently called Code Division Multiple Access

(CDMA), these multipaths can actually be utilized to enhance the detection process. The same cannot be totally achieved in TDMA systems. CDMA systems also utilize the silence voice gaps automatically. Trying to do the same in TDMA systems costs a little bit of hardware and is cost effective only when data and voice modems are colocated. These systems also fill the entire available spectrum all the time in all cells in a cellular environment, unlike FDMA systems which can reuse a certain frequency after a certain number of cells, implying automatic frequency reuse in the CDMA case. In this chapter, a brief description of certain spread spectrum systems is provided. We propose and describe a DS/CDMA system employing orthogonal concatenated short codes. We present different models of land mobile satellite channels and we discuss the need for power control and the representation of power control errors in mobile communications. We conclude the chapter by a statistical evaluation of data-bit block errors.

3.2 Examples of Spread Spectrum Systems

The purpose of this section is to provide a very brief description of certain direct sequence spread spectrum systems that are either in operation or in the planning phase at the present time.

One example is the satellite navigation system known as Global Positioning System (GPS). This system employs spread spectrum for two purposes: one purpose is to accomodate the multiple-access capability, since each user of the system must receive signals from four different satellites in order to make proper measurements of the user's position. The second reason is to increase the resolution capability so that the accuracy of the system is improved. For this purpose there is a PN sequence running at a chip rate of 10.23 megachips per second and modulated with data at a rate of 50 b/s. This leads to a processing gain of about 56 dB and enables the system to operate with input signal-to-noise ratios of the order of -34 dB.

Another example is the Telecommunication Data Relay Satellite System (TDRSS), which relays data to and from the space shuttle. In this case spread spectrum is used primarily to reduce the energy density of the radiation at the surface of the earth in order to meet CCIR requirements. The S-band portion of this system utilizes a PN code running at 11.232 megachips per second with a code period of 2047 chips. This is modulated by data either at 32 Kb/s or 216 Kb/s. Thus the processing gain is not extremely high, but this was not the original objective of using spread spectrum.

In this last example, we describe in more detail the recent Qualcomm proposition [3] where signature coding takes two different formats in the forward link (from base station to user) and reverse link (from user to base station). In the forward link, all signals transmitted from a cell in a particular CDMA radio channel share a common long quadrature pairs of pseudorandom code phase. They are distinguished at the mobile station receiver by mixing these with a binary orthogonal code based on Walsh functions (also known as Hadamard matrices). The Walsh function is 64 PN code chips long and represents 64 different orthogonal codes corresponding to the 64 active voice cells of the Qualcomm system. Orthogonality provides nearly perfect isolation between the multiple signals transmitted by the base station.

The code length is 32768 chips while the processing gain equals $T_b/T_c \cong 64$ for a voice band of 19.2 KHz and total spread spectrum bandwidth of 1.23 MHz. For more security, baseband data is interleaved and scrambled by a user specific long PN code. So a channel in the forward link of the Qualcomm CDMA system consists of a signal centered on an assigned radio channel frequency, quadriphase modulated by a pair of PN codes with an assigned time offset, biphase modulated by an assigned orthogonal Walsh function and biphase modulated by the encoded interleaved and scrambled digital information signal.

The Qualcomm reverse link also employs PN modulation using the same 32768 length binary sequences that are used for the forward link. Here, however, a fixed code phase offset is used. Signals from different mobile stations are distinguished by the use of a very long $(2^{42} - 1)$ PN sequence with a user address determined time offset. Thus, leading to an extremely large address space which inherently provides a reasonably high level of privacy.

The interleaved information is grouped into six symbol groups (or code words). These code words are used to select one of 64 different orthogonal Walsh functions for transmission. The Walsh function chips are combined with the long $(2^{42} - 1)$ and short ($2^{15}-1$) PN chip codes. Note that this use of the Walsh function is different than on the forward link. On the forward link, the Walsh function is determined by the mobile station's assigned channel while on the reverse link the Walsh function is determined by the information being transmitted.

The use of a 32768 chips long PN code while the processing gain (PG) is only 64 does not yield good short term balance between number of 1s and 0s in the code, which may hinder the acquisition, synchronization, demodulation, and equalization processes.

The use of PN codes with a finite number of crosscorrelation levels, i.e. Gold codes [4], guarantees a certain upper bound on the bit error performance and a large number of users equals (2^n+1) where n is the shift register length.

Introducing short concatenated sequences in place of long code ones offer great advantage in reducing the size of the matched filter correlators (MFC) in spread spectrum receivers [5]. We will show later that using synchronous short Gold concatenated codes (start of the code and start of data bits are synchronous) can minimize the probability of error for moderate loads compared to the asynchronous short concatenated codes (start of the code and start of data bits are not synchronous) and it is close to the classical CDMA system in that range. Decreasing the probability of error means increasing the capacity of the system.

3.3 System Aspects of The New Concatenated Direct Sequence Codes

Fig. 3.1 shows block diagrams of the proposed spread spectrum system. Rather than using a long direct sequence code of length ($2^{n_L}-1$), where n_L is the length of the linear feedback shift register (LFSR) generating this code, a succession of C shorter linear maximal codes yield the same code length. The length of the shorter codes is $(2^{n_L}-1)/C$. Without loosing generality, we assume this to be equal to the processing gain of our system, i.e.

$$PG = 2^{n_s} - 1 = T_b/T_c = (2^{n_L} - 1)/C \qquad (3.1)$$

where n_s, T_b, T_c, are the length of the LFSR of the short code, the data bit and code chips durations respectively. In some of the cases discussed in this work care should be taken to select n_s large enough for C maximal codes of length $(2^{n_s} - 1)$ to exist.

Fig.3.1 shows that the code chips and the data bearing signal are multiplied, amplified, upconverted and transmitted. The data bearing signal could be any phase modulation technique such as MPSK. The processing gain in this case is given by:

$$PG = 2^{n_s} - 1 \cong W/W_d = W/(2R_b/\log_2 M) = (T_b/T_c)\log_2 M \qquad (3.2)$$

If one uses $M = 2$, we obtain the DS/PSK case where the PG is given by (3.1). At any of the base station receivers (Fig.3.2), each receiver corresponds to a certain active call. Downconversion, carrier recovery, code acquisiton, code tracking and despreading take place followed by data demodulation and possible data bits error correction as in conventional DS systems.

Each user cycles through the short codes in a random or deterministic manner. In this work we simulate only the random case also its performance error serves as a worst case of the deterministic one. In the course of simulation, each user selects randomly every $(2^{n_s} - 1)T_c$ sec one of his C short codes for transmission. Within this category we simulate different cases corresponding to fading environment, orthogonal function usage, etc. as in cases $(1 - 13)$ Table 3.1. While cases $(14 - 26)$ corresponds to the classic long code results.

It follows from the above that the probability of k out of the K active users using the same short code is given by,

$$P(k) = \begin{pmatrix} K \\ k \end{pmatrix} (\frac{1}{C})^k (1 - \frac{1}{C})^{K-k} \qquad (3.3)$$

The average of this binomial distribution i.e, (K/C) can be minimized by increasing C. An alternate realization (not shown in Fig.3.1) is to store all consecutive short codes constituting the long code transmitter and consider this code as pseudorandom code composed of C shorter codes.

The latter realization allows the minimization of the level of mutual user interference by properly designing the order of using the short codes in each user case. For example, user 1 uses the long code composed of the concatenation $C_1C_2C_3C_4C_5\ldots C_c$, user 2 uses $C_2C_3C_4C_5\ldots C_cC_1$, user 3 employs $C_3C_4C_5\ldots C_cC_1C_2$, and so on. This deterministic arrangements guarantee for example that during any short code period $(2^{n_s}-1)$ all users have different codes and the user interference is bounded. For example, for short Gold codes, this is given by:

- for n_s odd,

$$X_g = \begin{cases} -\frac{[2^{(n+1)/2}\pm 1]}{M} & \text{with probability of level } 0.25 \\ -\frac{1}{M} & \text{with probability of level } 0.50 \end{cases} \tag{3.4}$$

- for n_s even and $n_s \neq 0 \bmod 4$,

$$X_g = \begin{cases} -\frac{[2^{(n+2)/2}\pm 1]}{M} & \text{with probability of level } 0.125 \\ -\frac{1}{M} & \text{with probability of level } 0.75 \end{cases} \tag{3.5}$$

while for linear maximal codes, crosscorrelations are less uniform and do not obey certain bounds.

However, these crosscorrelations may be higher than those provided when all users have the same short code but with each user assigned a different code phase. This latter case will be investigated in the simulations. Also we look at cases where each user selects randomly in each short code period one of the C short codes as well as one out of the possible $(2^{n_s}-1)$ code phases for his transmission in this period (case 1). The probability of k out of K possible users **exactly** overlapping in the code as well as code phase is gven by:

$$P(k) = \binom{K}{k}\left(\frac{1}{C.(2^{n_s}-1)}\right)^k\left(1-\frac{1}{C.(2^{n_s}-1)}\right)^{K-k} \tag{3.6}$$

In the latter case, the average number of users overlapping in a certain short code period is given by:

$$K/(C.(2^{n_s}-1)) \tag{3.7}$$

If all K users overlap in the short code but not in the code phase the total average crosscorrelation is given by:

$$Y = K/(2^{n_s}-1) \tag{3.8}$$

which is higher than that of (3.7). Neglecting the plain additive white Gaussian noise (AWGN) of the receiver for example means that the signal-to-interference ratio at the receiver demodulator is given by (assuming all interfering user signals having the same energy at the base station receiver):

$$\frac{S}{I} = \frac{E_b}{E_b(K/(2^{n_s}-1))} \tag{3.9}$$

In general, one may notice that out of K active users the code of the intended receiver may overlap with K_1 users in the code and code phase, K_2 overlaps in the code but not in phase, K_3 overlaps in the code phase but not the code and K_4 does not overlap in both leading to a total (S/I) at a typical receiver, given by:

$$\frac{S}{I} = \frac{1}{K_1 + \frac{K_2}{2^{n_s}-1} + K_3X_L + K_4X_L} \tag{3.10}$$

where X_L is the crosscorrelation of two linear maximal codes generated by different LFSRs. The above equation is the average signal-to-interference ratio for fixed number of users. Cases $(1-13)$ of Table 3.1 deal with subcases of this scenario depending on fading, etc. Equation (3.10) assumes the C short concatenated codes are linear maximal. If one uses a certain family of Gold codes instead, one obtains

$$\frac{S}{I} = \frac{1}{K_1 + K_2X_g + K_3X_L + K_4X_L} \tag{3.11}$$

If each user selects the code and code phase randomly during each $(2^{n_s}-1)T_c$ period, K_1, K_2, K_3, and K_4 will be random variables. If we force all users to have the same linear maximal short code but different code phases, we obtain $K_2 = K$, $K_1 = K_3 = K_4 = 0$ and

$$\frac{S}{I} = \frac{(2^{n_s} - 1)}{K} \tag{3.12}$$

Further to this scenario, if we use Gold codes, $K_4 = K$, $K_1 = K_2 = K_3 = 0$ and

$$\frac{S}{I} = \frac{1}{K_4 X_L} \tag{3.13}$$

In the previous scenarios we have assumed that the bit and chip edges of each user are synchronized at the base station. However, various users bits edges can change anywhere within the short code length. To further minimize the user cross-interference, we employ the group of orthogonal functions used in [3]. Users are split into groups, each group identified by a certain binary periodic function $\phi(t)$. Each user in a certain group i, multiplies his DS/MPSK signal (Fig.3.1) by this periodic function (Figs.3.3-3.4) before transmission. Corresponding multiplication by the same $\phi(t)$ at the receiver, yields back the data bits. Crosscorrelations of these orthogonal functions, i.e. $E\{\phi_i(t)\phi_j(t+\tau)\}$, are minimal over a wide range of delays and multiplying DS/MPSK signal by these orthogonal functions further decreases the crosscorrelation of DS/MPSK signals as will be made clear through simulation results.

Fading Link	no. of Walsh functions groups	Walsh function usage	Fading environment	Codes and corresponding cases	
				Short codes	Long codes
0	0	0	0	(1), case 1	(2), case 14
0	4	1	0	(1), case 2	(2), case 15
0	8	1	0	(1), case 3	(2), case 16
0	12	1	0	(1), case 4	(2), case 17
0	16	1	0	(1), case 5	(2), case 18
1	0	0	1	(1), case 6	(2), case 19
2	0	0	1	(1), case 7	(2), case 20
1	0	0	2	(1), case 8	(2), case 21
2	0	0	2	(1), case 9	(2), case 22
1	0	0	3	(1), case 10	(2), case 23
2	0	0	3	(1), case 11	(2), case 24
1	0	0	4	(1), case 12	(2), case 25
2	0	0	4	(1), case 13	(2), case 26
0	0	0	0	(3), case 27	
0	0	0	0	(4), case 28	
0	0	0	0	(5), case 29	

key
Up link (1)
Down link (2)
No fading link (0)

key
Ray. flat fading (1)
Ray. freq. sel. fading (2)
Rice flat fading (3)
Rice freq. sel. fading (4)

key
Asynchronous short concatenated codes (1)
Long code (2)
Short concatenated codes (3)
Short concatenated codes with changing initial states (4)
Short Gold codes (5)

Table 3.1: Key for first part of the different cases simulated.

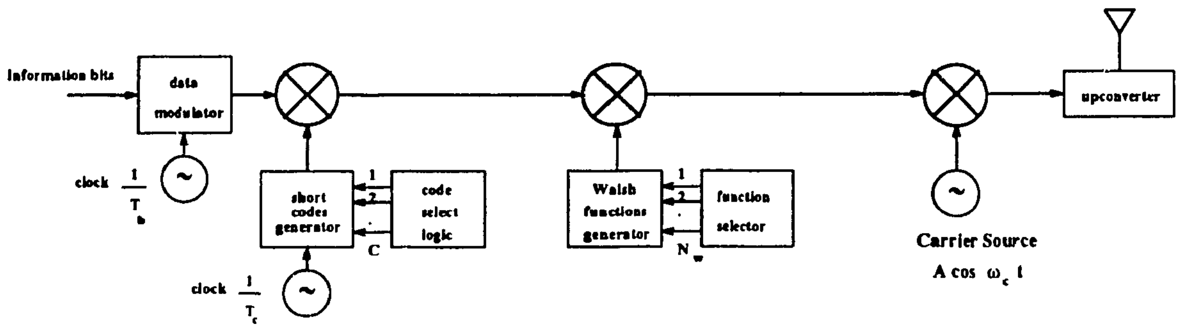


Figure 3.1: Generic Transmitter of the proposed DS orthogonal short code system.

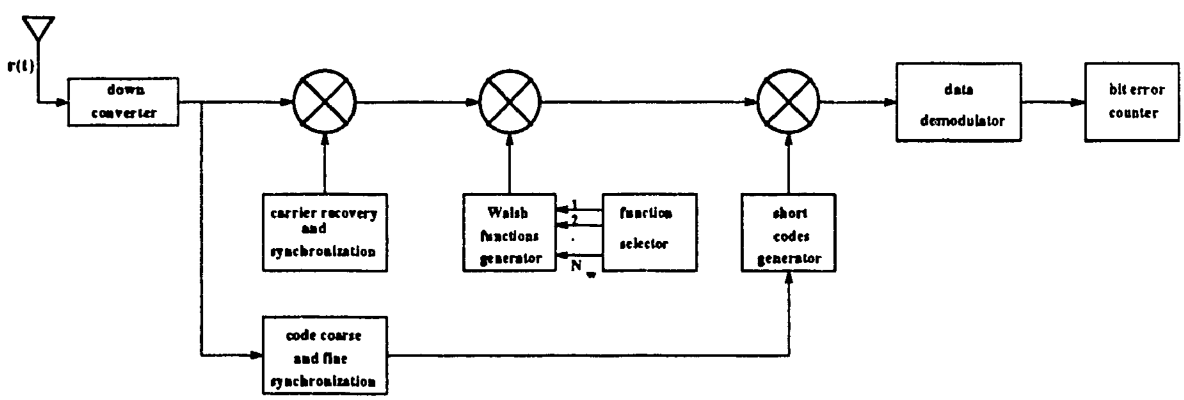


Figure 3.2: Generic receiver of the proposed DS orthogonal short code system.

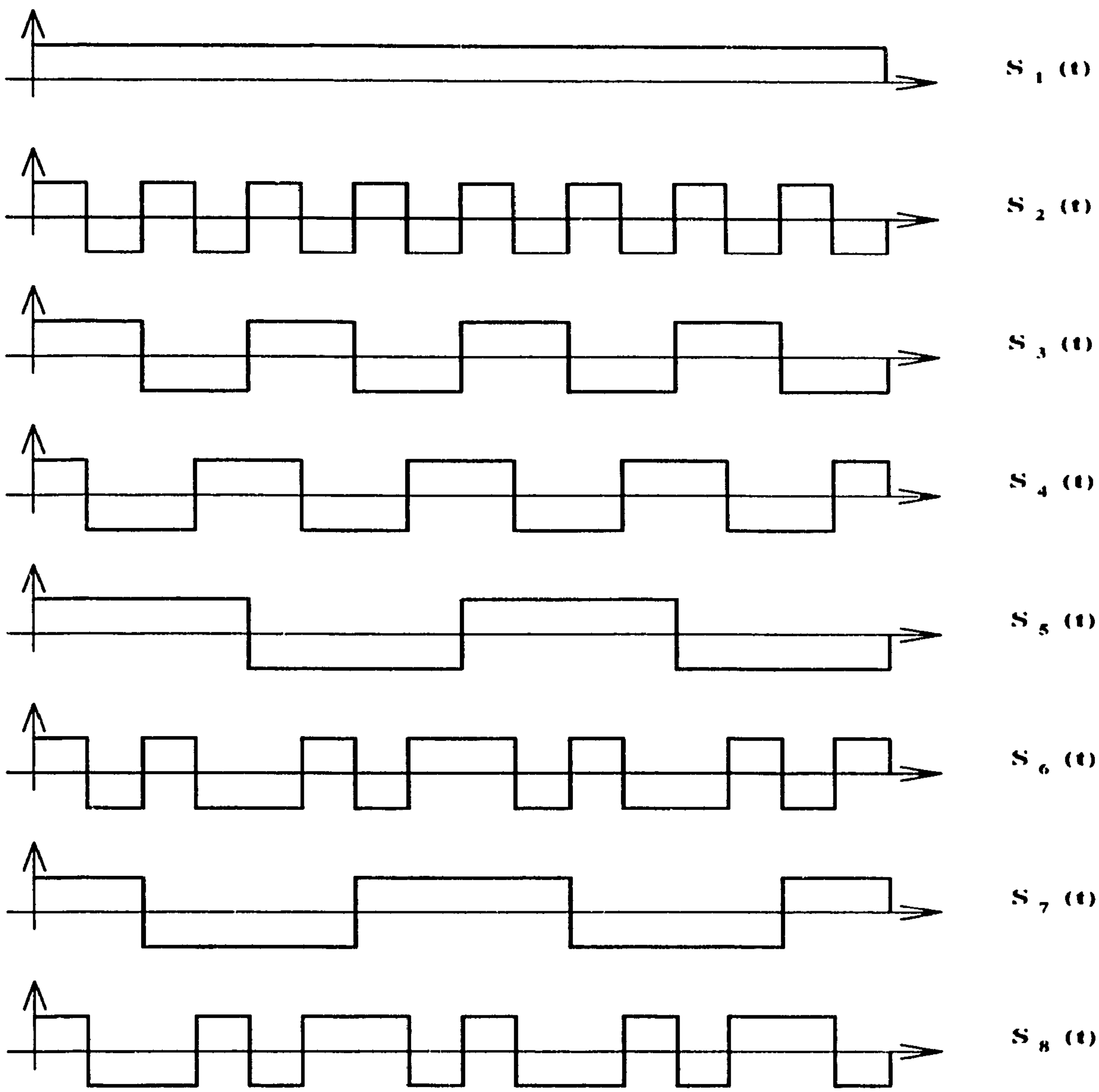


Figure 3.3: Walsh functions used in the simulation.

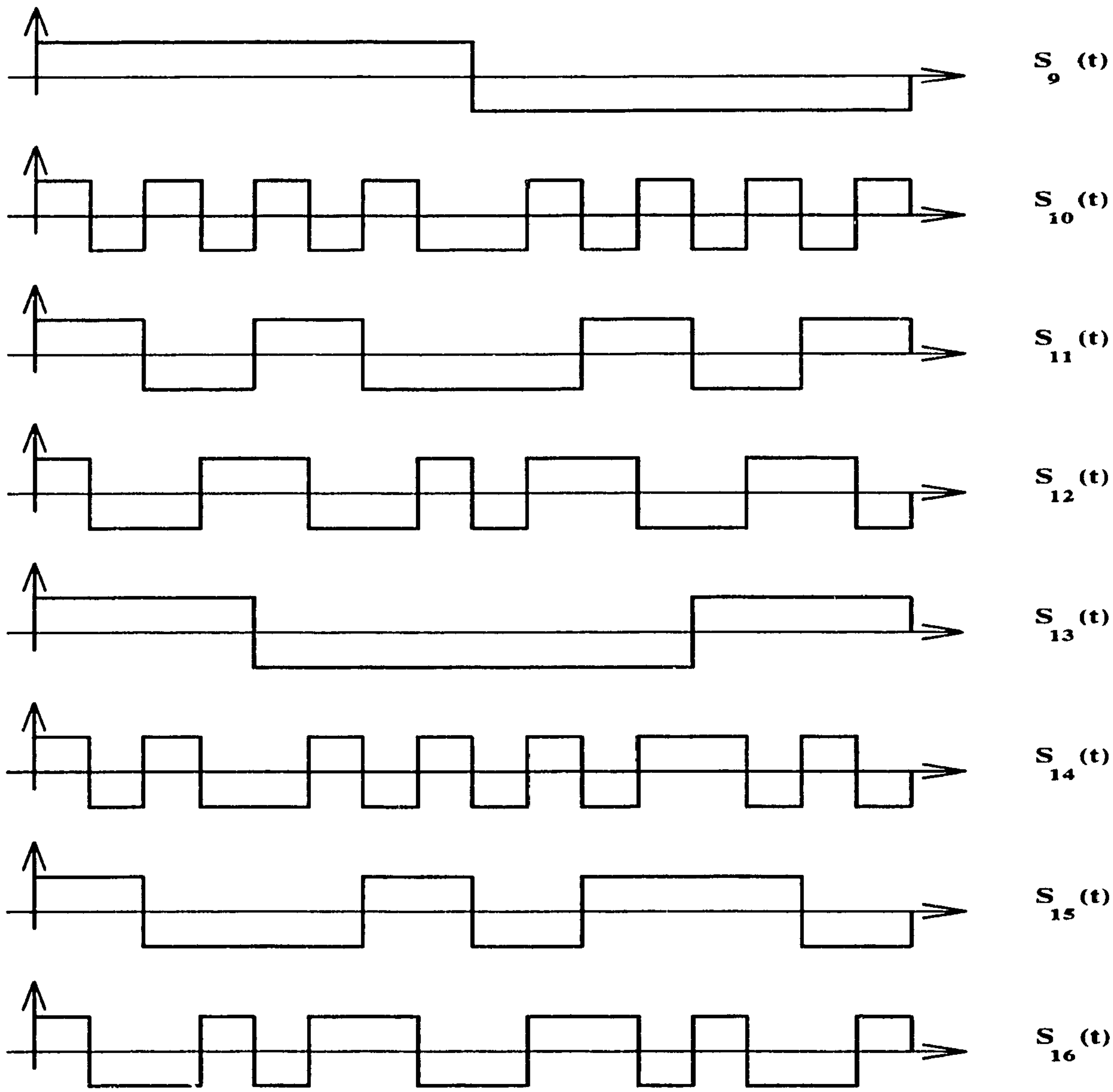


Figure 3.4: Walsh functions used in the simulation.

3.4 Land mobile satellite channels

Satellite communications with land mobile terminals are plagued by severe degradation due to signal shadowing and multipath fading. Shadowing is the attenuation of the direct path, i.e. line-of-sight (LOS), over the total signal bandwidth caused by trees , buildings, hills, and mountains. The process, which may be explained in terms of absorption, diffraction, and scattering, is strongly dependent on the signal path length through the obstacle, the type of obstacle, the elevation angle of the satellite with respect to the position of the mobile unit, the direction of travel, and carrier frequency [8, 9].

Multipath fading is due to the so-called signal diffuse component defined in [9]. The signal, transmitted from a satellite, illuminates obstacles in the vicinity of the mobile unit, which results in reflected energy emanating from multiple scatterers. Waves from these scatterers vary randomly in polarization, amplitude, and phase according to the nature of the scatterer and different propagation distances. All the contributions combine at the receiver to produce a distorted version of the transmitted signal. In addition, every wave shows a Doppler frequency shift proportional to the relative speed between any scatterer and the vehicle. Summing all the wave components, we observe a form of frequency spreading in addition to a frequency distortion. The maximum Doppler shift, termed fade rate [9], is closely related to the dynamics of fading. Typical values in L band and for the velocity of the mobile unit less than 100 Km/h range between 100 and 200 Hz [10, 11]. Fig. 3.5 shows the UHF mobile downlink channel.

Other effects on the link such as Faraday rotation, ionospheric scintillation, and the presence of a signal component reflected from the ground in the direction of the satellite, vanish in relation to particular carrier frequencies (L band) or antenna radiation patterns [9].

If we consider the receiver motion and the Doppler effect associated to each wave, the channel must necessarily be described as time variant. The effects observable on the received signal may be expressed in terms of superposition on two different phenomena [12]:

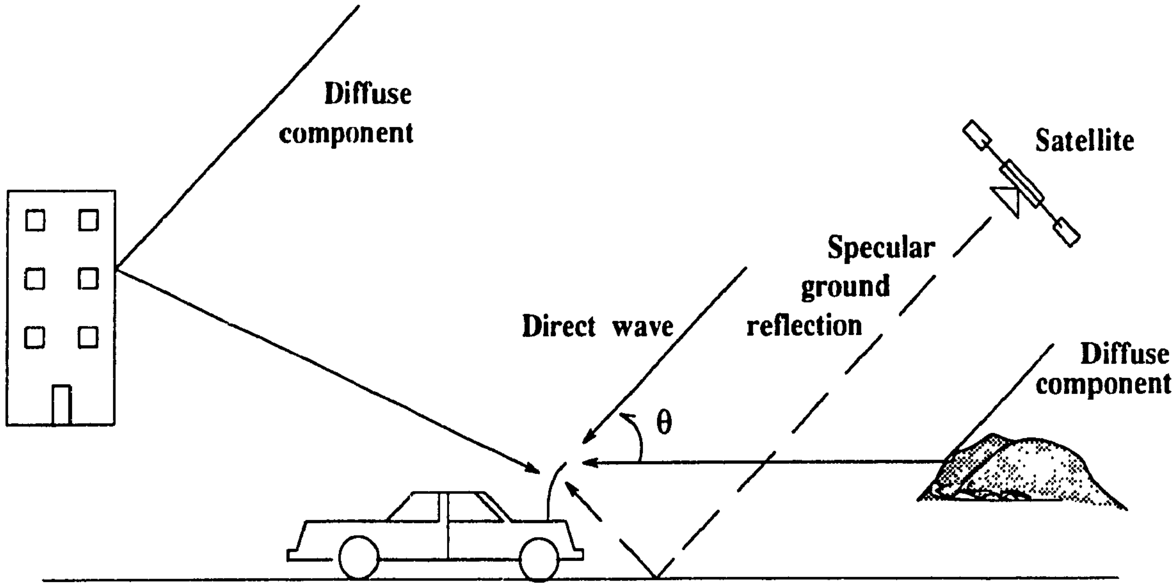


Figure 3.5: Mobile-satellite downlink model.

- A slow fading component, mainly due to local topographic conditions, antenna height, and other environmental conditions, whose statistics can be assumed to be of the lognormal type.
- A fast fading component (multipath fading) due to reflections from obstacles and mobile receiver movement.

While for narrow-band transmission, the multipath medium causes fluctuations in the received signal amplitude and phase (modeled by a multiplying complex Gaussian fading process [13]), if wide band signaling is used, not all the frequency components of the transmitted signal are influenced by the channel in the same manner and consequently a series of delayed and attenuated echoes, forming multipath delay spread, can be observed at channel output [14]. Being replicas of the same information signal over independently fading channels, the transmission paths produce a diversity effect because not all of them tend to fade together. For terrestrial suburban or rural channel, the multipath delay spread might be on the order of several microseconds. While, in low-altitude satellite channels, the delay spread is more typically less than 100 ns; thus, the coherence bandwidth (inverse of multipath delay spread) of the satellite multipath channel is at least 10 MHz [15]. This means that any CDMA system design that seriously intended to make constructive use of the spreading to combat multipath would have to be spreading by an amount greater than 10 MHz.

3.5 Channel Models

As the vehicle moves from one location to another, most of the propagation characteristics vary, resulting in different shadowing conditions and a nonstationary statistical character of the received signal. Several studies have been conducted to characterize the propagation effects for mobile satellite communications. We start by a study conducted by Vucetic *et al.* [9] and we conclude by a general study conducted by Lutz *et al.* [10] (see also [16], [17]).

The environment characteristic variations may be considered constant, due to their relative slowness, within a small area. Therefore, a land satellite mobile channel can be modeled as "stationary" over that small area. In order to derive a general model for any area, Vucetic *et al.* [9] have suggested dividing the whole area of interest into M areas with constant environmental properties. Each of the M areas is modeled by one of the three models: urban areas channel model, open areas channel model, or suburban and rural areas channel model.

A- Channel Models for Urban Areas

In urban areas, the direct line between the mobile unit and the satellite is almost completely obstructed by high buildings. Therefore electromagnetic energy propagation in urban areas is largely by way of scattering. In such a case, the vehicle picks up reflected signals from all directions in the horizontal plane. The received signal consists of many independent components with random phases. These signals collectively add to give a net signal at the receiver that varies randomly in amplitude and phase. We deal here with systems with a signal bandwidth small enough that a nonfrequency selective model is appropriate.

Let us consider a situation where a mobile receives N indirect signals. The ith indirect signal, coming in at an azimuthal angle θ_i with respect to the direction of travel, of the receiver will experience a Doppler shift

$$f_{di} = \frac{v f_c}{c} \cos\theta_i \tag{3.14}$$

where v is the velocity of the mobile unit, f_c is the frequency of the carrier and c is the speed of light in the medium.

The instantaneous value of the ith reflected signal is

$$d_i(t) = D_i(t)\cos(2\pi f_c t + \phi_{di}) \tag{3.15}$$

where $D_i(t)$ is the signal envelope. The ith signal phase ϕ_{di} is given by

$$\phi_{di} = 2\pi f_{di} t + \phi_i \tag{3.16}$$

where ϕ_i is the transmitted signal phase. Alternatively, the ith reflected signal can be expressed as

$$d_i(t) = d_{Ii}(t)\cos 2\pi f_c t - d_{Qi}(t)\sin 2\pi f_c t \tag{3.17}$$

where d_{Ii} is the ith signal in-phase and d_{Qi} is its quadrature component. The resulting diffuse signal is given by the sum of N independent waves

$$d(t) = \sum_{i=1}^{N} D_i(t)\cos(2\pi f_c t + \phi_{di}) \tag{3.18}$$

One can also express the resulting signal in terms of its in-phase and quadrature components as

$$d(t) = d_I(t)\cos 2\pi f_c t - d_Q(t)\sin 2\pi f_c t \tag{3.19}$$

The resulting I and Q components can be computed as the sum of the corresponding individual components as

$$d_I(t) = \sum_{i=1}^{N} d_{Ii}(t) \tag{3.20}$$

and

$$d_Q(t) = \sum_{i=1}^{N} d_{Qi}(t) \tag{3.21}$$

As a consequence of the Central Limit Theorem, when N gets very large and approaches infinity, as is the case in heavily built-up areas, $d_I(t)$ and $d_Q(t)$ each become uncorrelated Gaussian random process with zero mean and variance σ_d^2. The probability density function of the signal envelope $D(t) = \sqrt{d_I^2(t) + d_Q^2(t)}$ has a Rayleigh distribution of the form

$$p(D) \;=\; \frac{D}{\sigma_d^2} \exp \frac{-D^2}{2\sigma_d^2} \qquad \text{, for } D > 0 \tag{3.22}$$

where $2\sigma_d^2$ is the mean signal power which depends on the properties of the surrounding terrain. The phase of the received signal is uniformly distributed from 0 to 2π.

B- Channel Model for Open Areas

In open areas such as farm land or open fields, there are no obstacles on the line-of-sight path. The distortion-free received direct wave $s(t)$ interferes with the diffuse wave $d(t)$. The resulting wave is given by the sum

$$r(t) = s(t) + d(t) + n(t) \tag{3.23}$$

where $n(t)$ is additive white Gaussian noise (AWGN).

The diffuse component consists of many reflections from the nearby surface, each of them being independent and randomly phased. Hence, the probability density function of its envelope $D(t)$ is Rayleigh and given by (3.22). The diffuse multipath component interferes with the direct signal. We assume the envelope of the direct signal is constant in obstruction-free open areas. The sum of a constant envelope direct wave and a Rayleigh distributed diffuse wave results in a signal with Rician envelope statistics with probability density function [15]

$$p(R) = \frac{R}{\sigma^2} \exp(-\frac{R^2 + A_s^2}{2\sigma^2}) I_0(\frac{RA_s}{\sigma^2}) \tag{3.24}$$

where A_s is the amplitude of the specular (direct wave) component of the Rician part of the density, $2\sigma^2$ is the average scattered power due to multipath propagation and I_0 is the modified Bessel function of the first kind and zeroth order.

C- Channel Model for Suburban and Rural areas

In the suburban and rural areas, roads are surrounded by trees, houses, or small buildings. All of these obstacles near the mobile unit cause signal shadowing, manifested as an attenuation of the direct wave. As the vehicle moves along the road, the attenuation of the direct signal varies. It has been experimentally observed that the attenuation of the direct wave undergoes lognormal distribution [18], [19].

Since the variations of the direct signal envelope are slow, we can assume that its value over small areas (typically several tens of carrier wavelengths) is constant. The conditional pdf of the received signal envelope, given the envelope of the direct wave S, is Rician

$$p(R/S) = \frac{R}{\sigma_d^2} \exp(-\frac{R^2 + S^2}{2\sigma_d^2}) I_0(\frac{RS}{\sigma_d^2}) \tag{3.25}$$

The statistics of the direct signal envelope are described by the lognormal pdf

$$p(S) = \frac{1}{\sqrt{2\pi}\sigma_1 S} \exp(\frac{-(\ln S - \mu)^2}{2\sigma_1^2}) \quad ,0 \leq S \leq \infty \tag{3.26}$$

with the mean

$$m_s = \exp(\mu + \frac{\sigma_1^2}{2}) \tag{3.27}$$

and the variance

$$\sigma_s^2 = e^{(2\mu+\sigma_1^2)}(e^{\sigma_1^2} - 1) \tag{3.28}$$

where μ and σ_1 are parameters. The total probability is given by

$$p(R) = \int_0^\infty p(R/S)p(S)dS \tag{3.29}$$

A general statistical propagation channel model for land mobile satellite communications (Fig. 3.6), based on narrow band measurements at a single frequency, was developped by Lutz *et al.* [10]. The applications of the model are limited to the analysis of systems where the bandwidth of the transmitted signal is small compared to the coherence bandwidth of the channel. Under these conditions, the concept of multiplying fading [13] can be applied, so the received signal is the transmitted signal multiplied by a complex (Rice or Rayleigh distributed) fading process. Lutz *et al.* show that the model may be used only for signal bandwidths up to several tens of KHz. In that model two propagation link states are considered. Shadowing and no shadowing. The channel model is taken to be Rayleigh a fraction B of time , corresponding to when the signal is shadowed, and Rician the fraction $(1 - B)$ of the time. That is, the probability density function of the received amplitude is given by the mixture density

$$f_R(R) = B\frac{R}{\sigma^2} \exp(-\frac{R^2}{2\sigma^2}) + (1 - B)\frac{R}{\sigma^2} \exp(-\frac{R^2 + A_s^2}{2\sigma^2}) I_0(\frac{RA_s}{\sigma^2}) \tag{3.30}$$

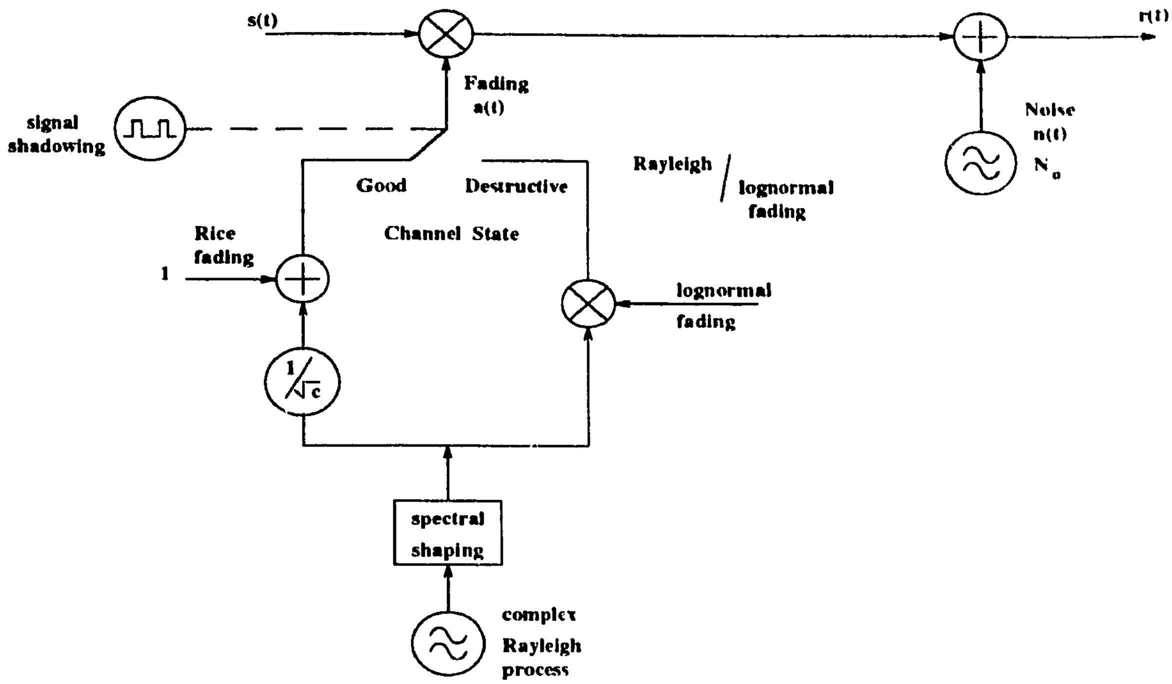


Figure 3.6: Dynamic model of the land mobile satellite channel. The signals s(t), n(t), r(t), and the fading process a(t) are complex valued.

In order to conform to the notation introduced in [10], we define

$$c = \frac{A_s^2}{2\sigma^2} \tag{3.31}$$

We notice that B, time-share of shadowing, is the most important parameter of the land mobile satellite channel. $(1 - B)$ is the time-share during which a transmission is possible and therefore represents a rough estimate for the achievable gross throughput of digital transmission systems.

Let us assume in the absence of either fading or shadowing, a signal arrives at the receiver with a nominal power denoted S_{nom}. Then, in the absence of shadowing, but in the presence of the Rician fade, the average received power, for $A_s = 1$, is

$$S_{av_R} = S_{nom}(A_s^2 + 2\sigma^2) = S_{nom}(1 + \frac{1}{c}) \tag{3.32}$$

If the user is shadowed, the received power is given by

$$S_{av_S} = S_{nom}(2\sigma^2) = S_{nom}(\frac{1}{c}) \tag{3.33}$$

Assuming that the power control algorithm is not fast enough to track power variations due to the pure multipath fading (i.e., fading not caused by shadowing) but is fast enough to track the variations due to shadowing, then during that fraction of the time when the signal is shadowed, its transmitted power is multiplied by

$$p = \frac{S_{av_R}}{S_{av_S}} = \frac{S_{nom}(1 + \frac{1}{c})}{S_{nom}(\frac{1}{c})} = 1 + c \qquad (3.34)$$

3.6 Power Control

One difficult problem in applying the DS/CDMA scheme to cellular land mobile radio is the near-far problem: signals received from transmitters close to the receiver are strong, while those from far transmitters are weak.

Insufficient signal-to-interference power ratio (SIR) increases the bit error rate (BER) which leads to degraded communication quality. On the other hand, excessive signal powers increase the interference to all other users sharing the DS/CDMA channel. Therefore, the traffic capacity is maximized when all transmitted signals are received at the minimum power needed to achieve the specified SIR, or in other words by ensuring that the average received signal power level of each user is equal. This implies that the transmitter power of each user must be accurately controled and that any error within the power control process will degrade system capacity.

For this reason accurate power control is an essential aspect of any practical CDMA system. Two mechanisms are usually used individually or simultaneously [26] to cope with power control problems in mobile communications networks. These two mechanisms are open loop and closed loop power control.

In the open loop mechanism each mobile unit measures the power level of the signal from the base station to which it is connected. Based on this measurement, the path loss is estimated. This estimate is then used by the mobile to adjust its own transmitter power. The stronger the received signal, the lower will be the mobile's transmitter power. In the closed loop power control mechanism, on the other hand, each base station demodulator measures the received signal strength from each mobile. The measured signal strength is compared to the desired signal strength for that mobile and a power adjustment command is sent to the mobile to increase or decrease its transmitter power by a predetermined amount.

However, from the implementation view point, transmitter power control (TPC) systems for DS/CDMA cellular mobile radios suffer from several imperfections which may significantly decrease traffic capacity. In a practical power control system, the average received power at the base station may not be the same for each user signal.

The performance of a power control system depends on the speed of the adaptive power control system, dynamic range of the transmitter, spatial distribution of users and propagation statistics (such as fading and shadowing). All these factors influence the *pdf* of the received power. The *pdf* of the received power due to the combined influence is assumed to be lognormal [29]. The imperfection in the power control system is determined by the logarithmic standard deviation σ of the lognormal power distribution of the received signal.

In the case of perfect power control the logarithmic standard deviation is 0 dB. Furthermore, even if the TPC system is perfect, the traffic capacity is largely affected by the propagation constant because the smaller it is, the larger the received interference power is.

A widely accepted model for the signal transmission environment encounterd in the mobile communications systems ([27], [28] and [20]) indicates that the power of a user, subjected to power control error and shadow, is given by,

$$W(r) = P_t r^{\alpha} 10^{(-\gamma-\delta)/10} \tag{3.35}$$

where P_t is the transmitted signal power, r is the distance between transmitter-receiver pair, α is the propagation constant, γ represents power control and is assumed to be a zero mean Gaussian random variable of standard deviation σ_γ, and δ represents shadowing and is assumed to be a zero mean Gaussian random variable of standard deviation σ_δ. For the different simulated cases in this thesis, the influence of the propagation constant on the performance of our proposed system is not investigated and it can be a subject for future study.

3.7 Block Error statistics

In packet data communications, the blocks of data can be protected against transmission errors either by forward error correction (FEC) or by automatic repeat request (ARQ). For a plain ARQ scheme with error detection, the probability of a data block containing no errors is directly related to the throughput of the ARQ scheme . If a t-error correcting block code is applied, the probability of correct block decoding equals the probability of at most t errors occuring in the data block. Investigations concerning the achievable throughput of ARQ schemes as well as the optimization of code rate, signaling rate and data block length are contained in [22] and [23].

Transmission channels with memory such as the land mobile satellite channel which has a strong statistical dependency among error gaps can be characterized by multi-gap distribution or by error correlation function [24], [2].

With error gap ν being the number of consecutive correctly received symbols between errors plus 1, the gap density $f_A(\nu)$ is defined as

$$f_A(\nu) = P(e_1 = 0 \bigcap e_2 = 0 \bigcap \cdots \bigcap e_{\nu-1} = 0 \bigcap e_\nu = 1 / e_0 = 1) \tag{3.36}$$

where e_i is 0 for an error-free symbol and 1 for an error.

The complementary gap distribution function $F_A(\nu)$ which is the probability of an error gap being larger than ν, is given by

$$F_A(\nu) = 1 - \sum_{j=1}^{\nu} f_A(j) \tag{3.37}$$

The gap density $f_A(j)$ is related to the block error density $P(m,n)$, probability of exactly m errors occuring in a block of n digits, by [25]

$$f_A(j) = \frac{1}{p}[P(0,j-1) - 2P(0,j) + P(0,j+1)] \tag{3.38}$$

where p is the mean bit error probability.

Inserting (3.38) into (3.37) and assuming that $P(0,0) = 1$ and $P(0,1) = 1 - p$, the complementary gap distribution function becomes

$$F_A(\nu) = \frac{P(0,\nu) - P(0,\nu+1)}{p} \tag{3.39}$$

The information of the complementary gap distribution $F_A(\nu)$ is completely contained in the block error probability densities $P(m,n)$, $n = \nu, \nu+1$ as indicated by (3.39) [10].

For random error channels, we have

$$f_A(n) \;=\; p.(1-p)^{n-1} \quad ,p \text{ is average probability of bit error} \tag{3.40}$$

If we define $\{G\}$ as a gap process, we have

$$\begin{aligned} f_A(G \le n) &= \sum_{i=1}^{n} p.(1-p)^{i-1} \\ &= p\sum_{i=1}^{n} q^{i-1} \qquad ,\text{for } q = 1-p \\ &= p\frac{1-q^n}{1-q} \\ &= (1-q^n) \end{aligned} \tag{3.41}$$

Let us denote the probability that a gap length is greater than n by

$$\begin{aligned} f_A(G > n) &= 1 - Pr\{G \le n\} \\ &= q^n \end{aligned} \tag{3.42}$$

However, for land mobile satellite channels, $Pr\{G > n\}$ might be written as

$$f_A(G > n) = 1 - \frac{1}{E}\sum_{i=1}^{n} EFG(i) \tag{3.43}$$

where E is the total number of errors and $EFG(i)$ is the number of error-free gaps of length i. For instance, the sequence 110001 has an error-free gap of length 3.

3.8 Conclusion

In this chapter, we have discussed briefly few examples that use long codes in DS/CDMA. Aspects of a DS/CDMA system, that employ orthogonal short concatenated codes, are presented. We have described different models of land mobile satellite channels and concluded the chapter by discussing the effects of these channels on the transmission and reception of the signal.

In the next chapter, the results of simulating different cases of long and short concatenated codes are shown. The performance is evaluated in terms of bit error rate and burst error and error-free gap distributions.

Bibliography

[1] A.K.Elhakeem, D.L.Schilling, W.P.Baier, M.Nakagawa, *"Code division multi-access networks I", IEEE JSAC, vol.12, no.4, May 1994.*

[2] A.K.Elhakeem, D.L.Schilling, W.P.Baier, M.Nakagawa, *"Code division multi-access networks II", IEEE JSAC, vol.13, no.4, June 1994.*

[3] *"An overview of the application of CDMA to digital cellular systems and personal cellular networks", TIA TR45.5, May 21, 1992.*

[4] D.V.Sarwate and M.B.Pursley, *"Crosscorrelation properties of pseudorandom and related sequences", Proc. IEEE, May 1980.*

[5] S.L.Maskara and J.Das *"Concatenated sequences for spread spectrum systems", IEEE Transactions on Aerospace and Electronic Systems, vol. AES-17, no.3, May 1981.*

[6] S.Lin and D.J.Costello *"Error control coding", Prentice-Hall,1983.*

[7] Elhakeem, A.K. et al., *"Modified SUGAR/DS: a new CDMA scheme", IEEE JSAC, vol.10, no.4, May 1992.*

[8] J. Goldhirsh and W. J. Vogel, *"Propagation effects for land mobile satellite systems: overview of experimental and modeling results", NASA, Reference Publication 1274, February 1992.*

[9] B. Vucetic and J. Du, *"Channel modeling and simulation in satellite mobile communication systems", IEEE Journal on Selected Areas in Communications, vol. 10, no. 8, October 1992.*

[10] E. Lutz, D. Cygan, M. Dippold, F. Dolainsky, and W. Papke, *"The land mobile satellite communication channel - recordings, statistics and channel model", IEEE transactions on Vehicular Technology, vol. 40, no. 2, May 1991.*

[11] F. Davarian, *"Channel simulation to facilitate mobile-satellite communications research", IEEE Trans. on Communications, vol. COM-35, no. 1, Jan. 1987.*

[12] E. Del Re, G. Benelli, G. Gastellini, R. Fantacci, L. Pierucci, and L. Pogliani, *"Design of a digital MLSE receiver for mobile radio communications", GLOBECOM '91.*

[13] J. G. Proakis, *"Digital Communications", Second Edition, McGRAW-HILL, 1989.*

[14] H. Hashemi, *"Simulation of the urban radio propagation channel", IEEE Trans. on Vehicular Technology, vol. VT. 28, no. 3, Aug. 1979.*

[15] B. R. Vojcic, R. L. Pickholtz, and L. B. Milstein, *"Performance of DS-CDMA with imperfect power control operating over a low earth orbiting satellite link", IEEE Journal on Selected Areas in Communications, vol. 12, no. 4, May 1994.*

[16] J. S. Butterworth, *"Propagation measurements for land mobile satellite systems at 1452 MHz", Commun. Res. Cent., Tech. Note no. 723, Dep. Commun., Canada, Aug. 1984.*

[17] W. L. Stutzman *et al.*, *"Mobile Satellite Propagation measurements and modeling: A review of results for systems engineers", Proc. Mobile Satellite Conf., May 3-5 1988, Pasadena, CA, pp. 107-117.*

[18] W. J. Vogel and E. K. Smith, *"Theory and measurements of propagation for satellite to land mobile communication at UHF", Proc. IEEE 35th Vehic. Technol. Conf., Boulder, CO, 1985, pp.218-223.*

[19] C. Loo, *"Measurements and models of a land mobile satellite channel and their applications to MSK signals", IEEE Trans. Vehic. Technol., vol. 35, no. 3, pp. 114-121, Aug. 1987.*

[20] P. Newson and M. R. Heath, *"The capacity of a spread spectrum CDMA system for cellular mobile radio with consdcration of system imperfections", IEEE Journal on Selected Areas in Communications, vol. 12, no. 4, May 1994.*

[21] E. Kudoh, *"On the capacity of DS/CDMA cellular mobile radios under imperfect transmitter power control", IEICE Trans. Commun., vol. E76-B, no. 8, Aug. 1993.*

[22] E. Lutz, W. Papke, and E. Plöchinger, *"Land mobile satellite communications - channel model, modulation and error control", Proc. 7th Int. Conf. on Digital Satellite Commun. 1986, Munich, Germany, pp. 537-543.*

[23] E. Lutz, *"Simulation of FEC/ARQ data transmission using stored land mobile satellite channels", Proc. IEEE 36th Veh. Technol. Conf. 1986, Dallas, TX, pp. 109-115.*

[24] L. N. Kanal and A. R. K. Sastry, *"Models for channels with memory and their applications to error control", Proc. IEEE, vol. 66, pp. 724-744, July 1978.*

[25] J.-P.A. Adoul, *"Error intervals and cluster density in channel modeling", IEEE Trans. Inform. Theory, vol. IT-20, pp. 125-129, Jan. 1974.*

[26] A. Salmasi and K. S. Gilhousen, *"On the system design aspects of code division multiple access (CDMA) applied to digital cellular and personal communications networks", Conf. Rec. of IEEE VTC '91, (ST. Louis), pp. 57-62, 1991.*

[27] E. Kudoh and T. Matsumoto, *"Effect of transmitter power control imperfections on capacity in DS/CDMA cellular mobile radios", Conf. Rec. of IEEE ICC'92 (Chicago), pp. 237-242, 1992.*

[28] E. Kudoh and T. Matsumoto, *"Effects of power control error on the system user capacity of DS/CDMA cellular mobile radios", IEICE Trans. Comm., vol. E75-B, no. 6, pp. 524-529, June 1992.*

[29] R. Prasad, M. G. Jansen, and A. Kegel, *"Capacity analysis of a cellular direct sequence code division multiple access system with imperfect power control", IEICE Trans. Comm., vol. E76-B, no. 8, pp. 894-905, Aug. 1993.*

Chapter 4

Simulation Results

It is known that short codes have less synchronization time and have better balance and spectral characteristics than their long counterparts. Concatenating short codes having the same period, in a predictable manner, to form a new long code makes it less vulnerable to acquisition problems than would be the case with a very long code.

We have simulated direct sequence CDMA for satellite channels, using Long codes and Short concatenated ones, in the L-band channel environment. We have incorporated thermal noise, Rayleigh flat fading (worst case channel), Rayleigh frequency selective fading, Rician flat fading, Rician frequency selective fading, and shadowing. In all these cases we have applied perfect power control or power control error. We have used also orthogonal coding by incorporating Walsh fuctions in the modulation and demodulation, in order to reduce the interference among different users. We have not dealt with Ka-band since its statistical characteristics are not yet clearly known. Performance evaluation is done for a data bit rate of 9.6 Kb/sec, but it could be extended to the vicinity of 64 Kb/sec.

It is known that there are fundamental differences among the characteristics of satellite and terrestrial mobile channels, such as larger coherence bandwidth and larger time delays from the mobile to the base station on the satellite channels. Larger coherence bandwidth means that by appropriate design of the code rate, Rake receiver is no longer needed to compensate for multipath fading. Open loop power control in satellite channels case is suggested due to the large propagation delay.

4.1 Signals and Simulation Procedure for the Underlying Proposed Systems

Fig.3.1 shows block diagrams of the proposed spread spectrum system. Rather than using a long direct sequence code, a succession of C shorter linear maximal codes yield the same code length. The data bearing signal could be any phase modulation technique such as MPSK. Each user selects randomly (worst case) at the begining of each data bit one of his C short codes for transmission. At the base station data bits and chip edges of each user can be synchronized or can change anywhere within the short code length.

To further minimize the user cross-interference, users are split into groups where each group is identified by a certain binary periodic orthogonal function $\phi(t)$. Each user in a certain group i, multiplies his DS/MPSK signal (Fig.3.1) by this periodic function (Figs.3.3-3.4) before transmission. Multiplying DS/MPSK signal by these orthogonal functions not only it further decreases the crosscorrelation of DS/MPSK signals but also decreases the effects of shadowing on short codes as will be made clear through simulation results.

Based upon the transmitter shown in Fig. 3.1 and assuming that several images (L_u) of the user signals are received, due to multipath fading, the received signal r(t) is given by

$$\begin{aligned} r(t) &= \sum_{u=1}^{U}\sum_{l=0}^{L_u} A_u\sqrt{P_L}\sqrt{P_S}d_u(t-\tau_{l,u})C_u(t-\tau_{l,u})W_u(t-\tau_{l,u})\Upsilon_{l,u}(t-\tau_{l,u}) \\ &\quad \times\cos(2\pi f_c(t-\tau_{l,u})+\theta_{l,u})+n(t) \end{aligned} \tag{4.1}$$

where U is the number of active users, L_u is the number of multipath signals of User u, A_u is the peak amplitude of User u signal, $P_L = 10^{-\gamma/10}$, is the power loss factor due to power control error, $P_S = 10^{-\delta/10}$, is the power loss factor due to shadowing, σ_γ, power control error standard deviation, and σ_δ, shadowing standard deviation, are assumed to be 0.5 dB and 5 dB respectively throughout the second part of the simulation, d_u is the data bit of uth User, C_u is the code chip of short or long user code depending on the case, f_c is the carrier frequency, $\theta_{l,u}$ is the phase of lth multipath component of uth User, $\Upsilon_{l,u}$ is the fading gain of lth multipath component of uth User, $\tau_{l,u}$ is the delay of lth multipath component of uth User, W_u is the orthogonal (Walsh or Hadamard) signal used by uth User, and $n(t)$ is the added white Gaussian noise.

Without lose of generality we can take the first user as the intended one ($u=1$) and assume that $\tau_{0,1} = \theta_{0,1} = 0$ meaning that the specular line of sight component of the intended component of the received signal is (both coarse and fine) synchronized to the local DS code. Also setting $\Upsilon_{0,u} = 1$ implies that the specular component of each of the users signals has power equal to $(A_u^2/2)$ while the remaining components have powers equal to $\Upsilon_{l,u}^2.(A_u^2/2)$,$l = 1, 2, \ldots, L_u$. The multipath delays lie in the range $0 < \tau_{l,u} < T_b$, where T_b is the duration of a data bit.

We have divided the simulation into two parts. In the course of simulation of the first part which corresponds to Figs. 4.2-4.13, values of the random variables $\theta_{l,u}$ and $\tau_{l,u}$ were supplied by uniform random variates in the range $(0, 2\pi)$ and $(0, T_b)$ respectively. Thermal noise $SNR = \frac{A_u^2}{\sigma_n^2} = 30dB$, where σ_n^2 is white Gaussian noise variance, was assumed to be fixed. The random variables $\Upsilon_{l,u}$ were assumed to have either a Rayleigh or Rician distribution depending on the case. Also, in the case of having a flat Rayleigh or Rician fading channel, we set $L_u = 0$, $\tau_{0,1} = 0$, $\theta_{0,1}$ a uniform random variable in the range $(0, 2\pi)$, and $\Upsilon_{0,1}$ is either Rayleigh or Rician distributed. Slow fading was assumed such that the $\theta_{l,u}$, $\tau_{l,u}$ and $\Upsilon_{l,u}$ random variates were called once every twenty bits. The two ray multipath model was also used means ($L_u = 1$).

By calling a uniform random number generator $U(0, 2^{n_s} - 1)$, a certain code phase shift is selected. All users keep their code phases throughout their calls.

For the case of synchronous short Gold codes, we choose two preferred polynomials of degree 7. Fig.4.1 shows the block diagram for generating short Gold code

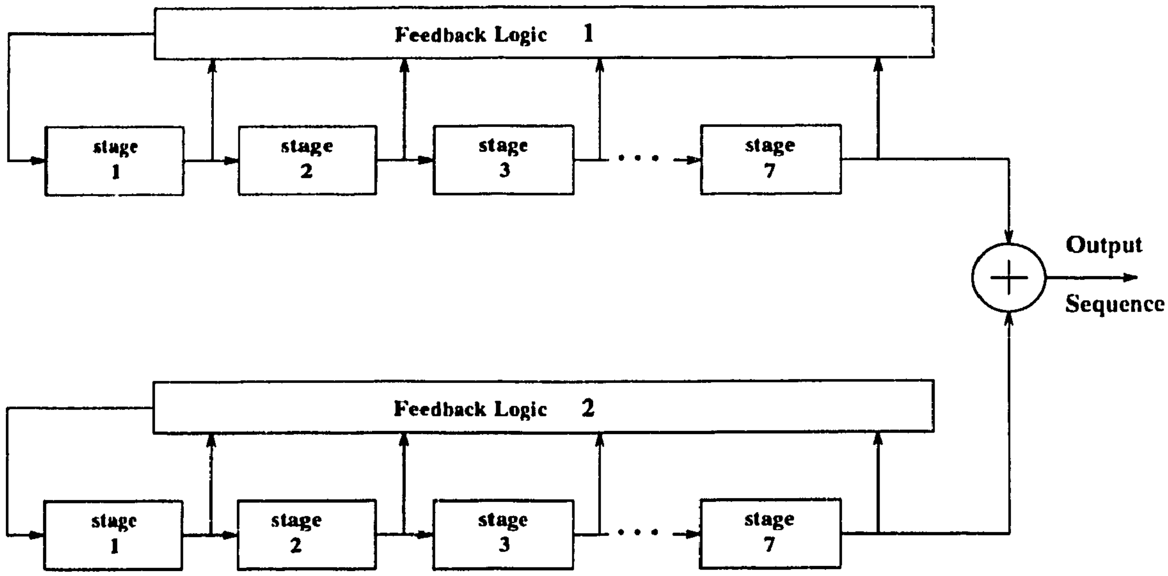


Figure 4.1: Block diagram for short Gold code sequence generator.

sequence. The spreading code is obtained by adding modulo 2 the outputs of the two LFSRs. In order to change randomly the spreading code at each data bit, we change randomly the initial state of one of the two LFSRs.

In the second part of the simulation corresponding to Figs. 4.14-4.23, we have added white Gaussian noise, shadowing, and/or power control error assuming now that thermal $SNR = 13dB$. The receiver correlates r(t) in (4.1) with $c_1(t)\cos(\omega_c t)W_1(t)$ (assuming intended signal is that of User 1) over the range $[0, T_b]$,the output of this correlator is given by:

$$y = \int_0^{T_b} c_1(t)\cos(\omega_c t)W_1(t)r(t)dt \tag{4.2}$$

This output is filtered and passed to a demodulator, where demodulation decisions are made, and the error counter is incremented and averaged so as to yield the average probability of bit errors.

Next section presents simulation results for different cases pertaining to different fading environments, different long, short codes (including short Gold codes), use of Walsh functions, and different code phases. The different cases are numbered and are given in Tables 4.1-4.2 and captions of the simulation results show the details of each case.

4.2 Simulation Results

Due to the huge simulation times encountered we had to reduce our total number of bits per run to 10^4. In general, we measure the performance of different cases in two ways:

4.2.1 Probability of error versus number of users

In total, fourty-nine cases are considered in this work (see Tables 4.1-4.2). We have used FORTRAN 77 in order to simulate these different cases. Source codes for some cases are shown in the Appendix. Since we are dealing with voice signals, we assume that the quality of signals received with probability of error less than 10^{-3} is good otherwise it is poor. In terms of the performance of these different cases Figs. 4.2-4.23 show some of the results that have been obtained. Figs. 4.2-4.13 belonging to Table 4.1 are divided into two groups (a) and (b). Group (a) of figures is for number of users less than twenty and group (b) is for number of users between twenty and two hundred. The reader may also notice the overall high bit error rate obtained which is not uncommon due to the fact that error correcting codes are not used here and large number of users are used. Voice activity is not taken into consideration, but since it is shown from simulation results that probability of error increases with number of users we can assume that introducing voice activity improves the performance of the network by reducing probability of error for less number of users or by allowing more users to communicate for the same probability of error. Processing gain, $PG = \frac{T_b}{T_c} = 127$ is used.

We define several types of channel models:

- Only multiaccess interference channel:

  Comparision of cases 1, 14, 27, 28 and 29 (Figs.4.2a-b). Fig.4.2a reveals that cases 1 and 29 have probability of error less than 10^{-4} in all the specified loads; case 27 has probability of error less than 10^{-3} for loads less than 5; case 14 has probability of error less than 10^{-3} for loads less than 7. Case 28 yields better results than case 27 for all the specified loads, and better than cases 14 and 27 for loads between 7 and 10. Fig.4.2b reveals that: case 28 yields better results up to a load of 60 users. In the range of 60-140 simultaneous users case 14 {Case 14 means long code with data bits and code chips synchronous with respect to each other.} yields better results. For number of users larger than 140 case 1 {Case 1 means short code with data bits and code chips not synchronous with respect to each other.} is preferred. Case 29 {Case 29 means short gold code with data bits and code chips synchronous with respect to each other.} has approximately the same performance as case 14 but it is always better than case 1 within our operating range (until 140 users). The increase in the capacity of case 29 over case 1 is approximately 20% in the operating range. The probability of error of case 28 {Case 28 means short code with data bits and code chips synchronous with respect to each other.initial state is changed randomly at the start of each data bit.} is always better than that of case 27 {Case 27 means short code with data bits and code chips synchronous with respect to each other.}. The reason may be due to the fact that as the number of users increase (in case 27), we will have to assign the same initial state to many users leading to occasional complete overlap in the code and phase. Secondly, most of the interference comes from users with different codes (not the same code), in which case randomization of the initial state of the code may yield better results. For example, 180 interferers and 9 different short codes, on average 160 users have different code than the intended user and 20 have the same code of the intended user.

- Long codes, multiaccess interference with/or without dividing number of users into many groups:

  Comparision of cases 14, 15, 16, 17 and 18 (Figs.4.3a-b). Fig.4.3a reveals that: For number of users less than 5 all cases have probability of error less than 10^{-3}. Case 16 yields better results for the specified loads. Outage probability (probability of error greater than 10^{-3}) of case 14 happens around 7 users, for case 15 it is around 11 users, and for the rest of the cases it is around 15 users. Fig.4.3b reveals the following:

the differences among these cases are small. Case 15 {Case 15 means long code with data bits and code chips synchronous with respect to each other and 4 Walsh function groups.} is the best for low loads up to 40 users. Case 14 is preferred for greater number of users.

- Asynchronous short codes, multiaccess interference with/or without dividing number of users into many groups:

  Comparision of cases 1, 2, 3, 4 and 5 (Figs.4.4a-b). Fig.4.4a reveals that for number of users less than 9 all cases have probability of error less than 10^{-3}. Outage probability for case 1 happens around 9 users, and for the rest of cases it happens around 13 users. Fig.4.4b reveals that case 4 {Case 4 means short code with data bits and code chips not synchronous with respect to each other and 12 Walsh function groups.} yields better results up to a load of 100 users. In the range of 100-150 simultaneous users case 2 {Case 2 means short code with data bits and code chips not synchronous with respect to each other and 4 Walsh function groups.} yields better results. For number of users larger than 150 case 1 is preferred.

- Long codes or asynchronous short codes, multiaccess interference with/or without dividing number of users into many groups:

  Comparision of cases 1, 2, 3, 4, 5 and 14 (Figs.4.5a-b):

  Fig.4.5a reveals that for number of users less than 7 all cases have probability of error less than 10^{-3}. Outage probability for case 1 happens around 9 users, for case 14 it happens around 7 users, and for the rest of cases it happens around 13 users. Probability of error of case 14 is the worst for number of users greater than 10. Fig.4.5b reveals that: case 4 yields better results up to a load of 60 users. In the range of 60-120 simultaneous users case 14 yields better results. Case 2 yields better results for number of users between 120 and 150. Case 1 is the best for number of users greater than 150.

- Long or asynchronous short codes, Rayleigh flat fading and up link (from users to satellite):

  Comparision of cases 6 and 19 (Figs.4.6a-b):

  {Case 6 means short code with data bits and code chips not synchronous with respect to each other, Rayleigh flat fading, up link and no usage of Walsh functions.}

  {Case 19 means long code with data bits and code chips synchronous with respect to each other, Rayleigh flat fading, up link and no usage of Walsh functions.}

  Fig.4.6a reveals that performance of case 6 is better than that of case 19 for number of users less than 6, while it is the opposite for greater number of users. Fig.4.6b shows that case 19 is better than case 6 in our operating range (less than 140 users). However, case 6 is better outside that range.

- Long or asynchronous short codes, Rayleigh flat fading and down link (from satellite to users):

  Comparision of cases 7 and 20 (Figs.4.7a-b):

  {Case 7 means short code with data bits and code chips not synchronous with respect to each other, Rayleigh flat fading, down link and no usage of Walsh functions.}

  {Case 20 means long code with data bits and code chips synchronous with respect to each other, Rayleigh flat fading, down link and no usage of Walsh functions.}

  Fig.4.7a reveals that outage probability of case 7 happens for number of users greater than 7 and for case 20 it happens around 8 users. Case 20 outperforms case 7 for all the specified loads. Fig.4.7b reveals the following:

  Case 7 is better than case 20 for number of users less than 30. Case 20 yields better results in the range of 30-140 simultaneous users. Case 7 is better for number of users greater than 140.

- Long or asynchronous short codes, Rayleigh frequency selective fading and up link (from users to satellite):

  Comparision of cases 8 and 21 (Figs.4.8a-b):

  {Case 8 means short code with data bits and code chips not synchronous with respect to each other, Rayleigh frequency selective fading, up link and no usage of Walsh functions.}

  {Case 21 means long code with data bits and code chips synchronous with respect to each other, Rayleigh frequency selective fading, up link and no usage of Walsh functions.}

  Fig.4.8a reveals that case 21 yields better results for number of users less than 5, while it is the other way around for greater number of users. Fig.4.8b reveals that case 21 is slightly better than case 8 for number of users less than 60 or greater than 140. Cases 8 and 21 have approximately the same performance for number of users between 60 and 140.

- Long or asynchronous short codes, Rayleigh frequency selective fading and down link (from satellite to users):

  Comparision of cases 9 and 22 (Figs.4.9a-b):

  {Case 9 means short code with data bits and code chips not synchronous with respect to each other, Rayleigh frequency selective fading, down link and no usage of Walsh functions.}

  {Case 22 means long code with data bits and code chips synchronous with respect to each other, Rayleigh frequency selective fading, down link and no usage of Walsh functions.}

  Fig.4.9a reveals that case 9 yields better results for loads less than 4 and greater than 9, while it is the opposite for number of users between 4 and 9. Fig.4.9b shows that case 22 is slightly better than case 9 for number of users less than 100. However, case 9 is better for number of users greater than 100.

- Long or asynchronous short codes, Rician flat fading and up link (from users to satellite):

  Comparision of cases 10 and 23 (Figs.4.10a-b):

  {Case 10 means short code with data bits and code chips not synchronous with respect to each other, Rician flat fading, up link and no usage of Walsh functions.}

  {Case 23 means long code with data bits and code chips synchronous with respect to each other, Rician flat fading, up link and no usage of Walsh functions.}

  Fig.4.10a reveals that outage probability of cases 10 and 23 happens around 6 users. Case 23 yields better results for number of users less than 7, while it is the opposite for greater number of users. Fig.4.10b reveals that case 23 is better than case 10 within our operating range. However, case 10 is better outside that range.

- Long or asynchronous short codes, Rician flat fading and down link (from satellite to users):

  Comparision of cases 11 and 24 (Figs.4.11a-b):

  {Case 11 means short code with data bits and code chips not synchronous with respect to each other, Rician flat fading, down link and no usage of Walsh functions.}

  {Case 24 means long code with data bits and code chips synchronous with respect to each other, Rician flat fading, down link and no usage of Walsh functions.}

  Fig.4.11a reveals that probability of error of case 11 is approximately 10^{-4} for all the specified loads. Outage probability of case 24 is around 7 users. Fig.4.11b reveals that case 24 is better than case 11 within our operating range. However, case 11 is better outside that range.

- Long or asynchronous short codes, Rician frequency selective fading and up link (from users to satellite):

  Comparision of cases 12 and 25 (Figs.4.12a-b):

  {Case 12 means short code with data bits and code chips not synchronous with respect to each other, Rician frequency selective fading, up link and no usage of Walsh functions.}

  {Case 25 means long code with data bits and code chips synchronous with respect to each other, Rician frequency selective fading, up link and no usage of Walsh functions.}

  Fig.4.12a reveals that outage probability of cases 12 and 25 happens around 6 users. Case 12 yields better results for number of users less than 6, while it is the opposite for greater number of users. Fig.4.12b reveals that case 25 is slightly better than case 12 for number of users less than 120. Cases 12 and 25 have approximately the same performance for number of users greater than 120.

- Long or asynchronous short codes, Rician frequency selective fading and down link (from satellite to users):

  Comparision of cases 13 and 26 (Figs.4.13a-b):

  {Case 13 means short code with data bits and code chips not synchronous with respect to each other, Rician frequency selective fading, down link and no usage of Walsh functions.}

  {Case 26 means long code with data bits and code chips synchronous with respect to each other, Rician frequency selective fading, down link and no usage of Walsh functions.}

  Fig.4.13a reveals that the outage probability of cases 13 and 26 happens around 6 users. Case 26 yields better results for number of users between 5 and 9. Fig.4.13b reveals that case 26 is slightly better than case 13 for number of users less than 130. However, case 13 yields better results for greater number of users.

In the second part of the simulation (see Table 4.2) we will notice that the degradation of performance of the above cases, when number of users increase, becomes more severe due to shadow, power control error, and AWGN acting on the system. Performance evaluation is done in terms of probability of error versus number of users. Number of users is reduced to be in the range from 0 to 20 users. Processing gains for long codes and asynchronous short codes are kept the same at $2^{15} - 1$ and $2^7 - 1$, respectively.

Examining the performance of cases 30-49 (Figs.4.14-4.23) we notice that it does not have a monotonic character due to the big influence of shadow on the system.

Comparision of cases 40 and 30 (Fig.4.14):

{Case 40} means long code with data bits and code chips synchronous with respect to each other, Shadow, Gaussian noise, 4 Walsh functions groups and no fading link.

{Case 30} means short code with data bits and code chips not synchronous with respect to each other, Shadow, Gaussian noise, 4 Walsh functions groups and no fading link.

Fig.4.14 shows that case 30 has somewhat better performance than that of case 40.

Comparision of cases 31 and 41 (Fig.4.15):

{Case 41} means long code with data bits and code chips synchronous with respect to each other, Shadow, Gaussian noise, Power control error, 4 Walsh functions groups and no fading link.

{Case 31} means short code with data bits and code chips not synchronous with respect to each other, Shadow, Gaussian noise, Power control error, 4 Walsh functions groups and no fading link.

Fig.4.15 reveals that the performance of long code and short codes is kept the same as in cases 40 and 30 except that power control error has affected short codes more than long code when number of users increase.

Comparision of cases 32-49 (Figs.4.16-4.23):

We notice that long code and short codes have approximately the same performance, and introducing power control error does not change the character of the performance of thes cases too much, since standard deviation of shadow is 5 dB and that of power control error is 0.5 dB.

4.2.2 Burst length and error-free gaps distributions

The length of a burst in a block of n digits may be defined as the length starting from the first error to the last error in the block (both errors included), irrespective of the nature of errors in between [1]. For burst length distribution versus burst error length and constant number of users, we define P_i, probability of having a burst length of i errors in a block of K bits, and K, maximum burst error length possible in the system.

We have $P_0 + P_1 + P_2 + \ldots + P_K = 1$, which implies that:

$$P_1 + P_2 + \ldots + P_K = 1 - P_0$$

where P_0 is the probability of having no error in the system.

Comparing the burst length distribution of cases 1 and 14 for 60 users in 17-bit data blocks (Fig. 4.24), we notice that cases 1 and 14 have the same burst error length but case 14 has slightly better burst error distribution.

Comparing the burst length distribution of cases 1 and 14 for 60 users in 34-bit data blocks (Fig. 4.25), we notice that cases 1 and 14 have the same burst error length but case 1 has slightly better burst error distribution.

Comparing cases 1 and 14 against random error channels (Figs. 4.26 and 4.27) we notice that cases 1 and 14 have almost the same character of random channels in terms of error-free gaps distribution which is related to $P(m, n)$, probability of exactly m errors occuring in a block of n digits [2].

Figs. 4.28 and 4.29 show that the effects of shadow and Gaussian noise on cases 1 and 14 in terms of error-free gaps distribution have been weakened by introducing orthogonal Walsh functions which have made cases 30 and 40 to have almost the same error-free gaps distribution as random error channels.

Fading Link	no. of Walsh functions groups	Walsh function usage	Fading environment	Codes and corresponding cases	
				Short codes	Long codes
0	0	0	0	(1), case 1	(2), case 14
0	4	1	0	(1), case 2	(2), case 15
0	8	1	0	(1), case 3	(2), case 16
0	12	1	0	(1), case 4	(2), case 17
0	16	1	0	(1), case 5	(2), case 18
1	0	0	1	(1), case 6	(2), case 19
2	0	0	1	(1), case 7	(2), case 20
1	0	0	2	(1), case 8	(2), case 21
2	0	0	2	(1), case 9	(2), case 22
1	0	0	3	(1), case 10	(2), case 23
2	0	0	3	(1), case 11	(2), case 24
1	0	0	4	(1), case 12	(2), case 25
2	0	0	4	(1), case 13	(2), case 26
0	0	0	0	(3), case 27	
0	0	0	0	(4), case 28	
0	0	0	0	(5), case 29	

key

Up link (1)
Down link (2)
No fading link (0)

key

Ray. flat fading (1)
Ray. freq. sel. fading (2)
Rice flat fading (3)
Rice freq. sel. fading (4)

key

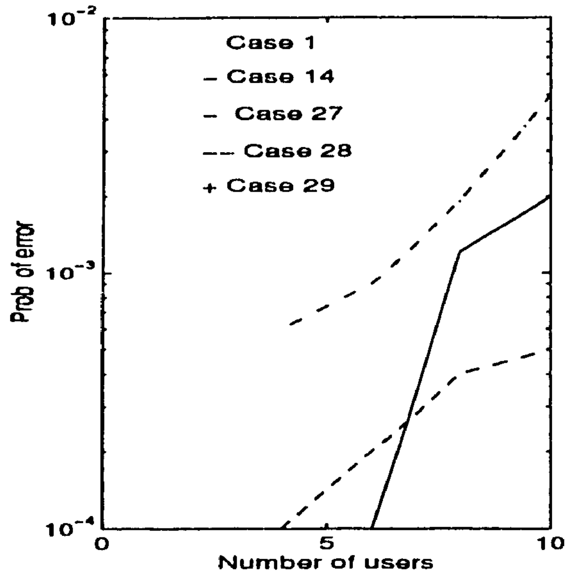
Asynchronous short concatenated codes (1)
Long code (2)
Short concatenated codes (3)
Short concatenated codes with changing initial states (4)
Short Gold codes (5)

Table 4.1: Key for first part of the different cases simulated.

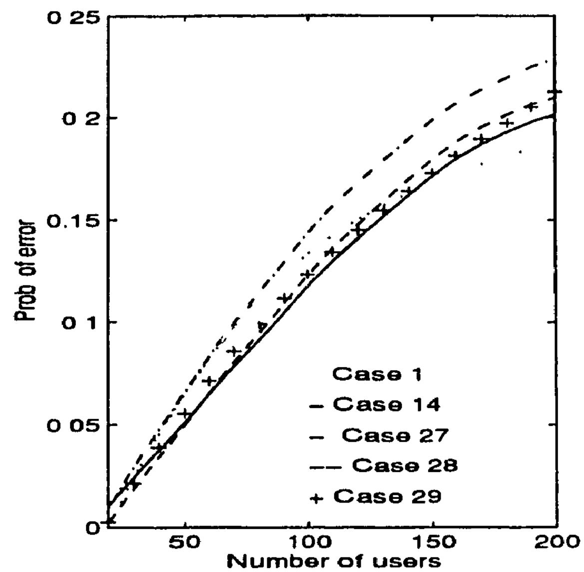
Fading link	Power control	Walsh functions groups	Channel environment	Codes and corresponding cases	
				Short codes	Long codes
0	0	4	5 , *	(1) , case 30	(2) , case 40
0	1	4	5 , *	(1) , case 31	(2) , case 41
1	0	4	1 , 5 , *	(1) , case 32	(2) , case 42
1	1	4	1 , 5 , *	(1) , case 33	(2) , case 43
1	0	4	2 , 5 , *	(1) , case 34	(2) , case 44
1	1	4	2 , 5 , *	(1) , case 35	(2) , case 45
1	0	4	3 , 5 , *	(1) , case 36	(2) , case 46
1	1	4	3 , 5 , *	(1) , case 37	(2) , case 47
1	0	4	4 , 5 , *	(1) , case 38	(2) , case 48
1	1	4	4 , 5 , *	(1) , case 39	(2) , case 49

key	key	key	key
Up link (1)	Perfect power control (0)	Gaussian noise (*)	Asynchronous short concatenated codes (1)
No fading link (0)	Power control error (1)	Ray. flat fading (1)	Long code (2)
		Ray. freq. sel. fading (2)	
		Rice flat fading (3)	
		Rice freq. sel. fading (4)	
		Shadow (5)	

Table 4.2: Key for second part of the different cases simulated.



(a)



(b)

Figure 4.2: Probability of error versus number of users.

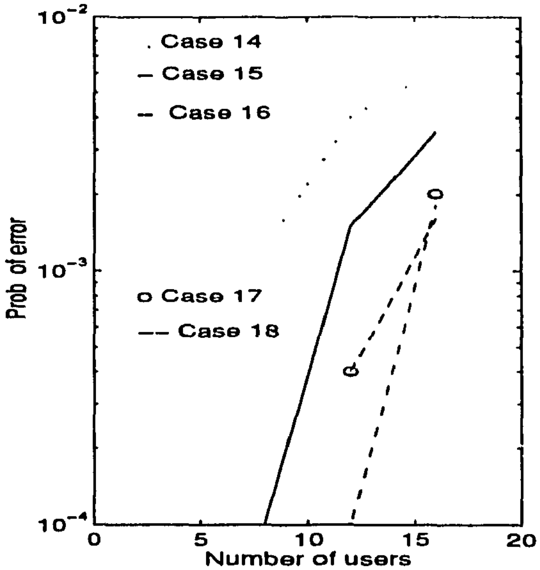
{Case 1} means short code with data bits and code chips not synchronous with respect to each other, no fading environment, no usage of Walsh functions and no fading link.

{Case 14} means long code with data bits and code chips synchronous with respect to each other, no fading environment, no usage of Walsh functions and no fading link.

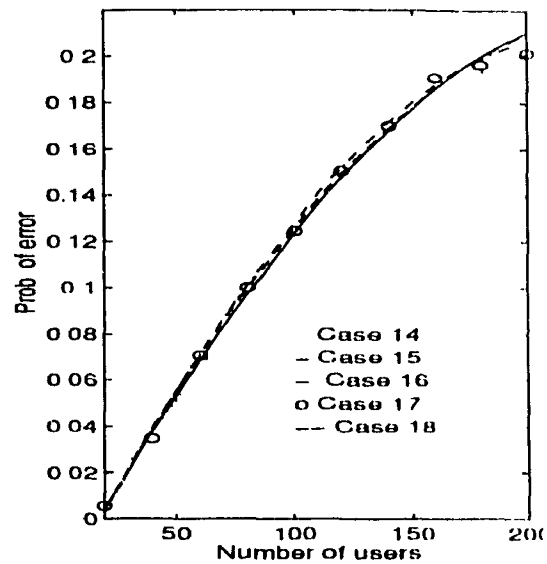
{Case 27} means short code with data bits and code chips synchronous with respect to each other, no fading environment, no usage of Walsh functions and no fading link.

{Case 28} is the same as case 27 except that initial state is changed randomly at the start of each data bit.

{Case 29} means short gold code with data bits and code chips synchronous with respect to each other, no fading environment, no usage of Walsh functions and no fading link.

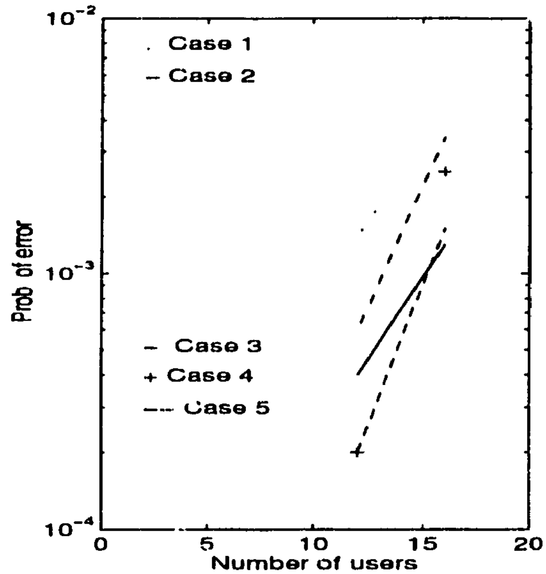


(a)

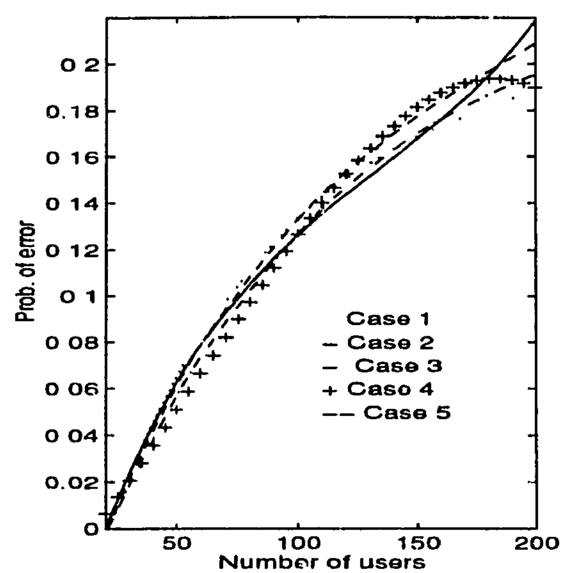


(b)

Figure 4.3: Probability of error versus number of users.
{Case 14} means long code with data bits and code chips synchronous with respect to each other, no fading environment, no usage of Walsh functions and no fading link.
{Case 15} means long code with data bits and code chips synchronous with respect to each other, no fading environment, 4 Walsh function groups and no fading link.
{Case 16} means long code with data bits and code chips synchronous with respect to each other, no fading environment, 8 Walsh function groups and no fading link.
{Case 17} means long code with data bits and code chips synchronous with respect to each other, no fading environment, 12 Walsh function groups and no fading link.
{Case 18} means long code with data bits and code chips synchronous with respect to each other, no fading environment, 16 Walsh function groups and no fading link.



(a)



(b)

Figure 4.4: Probability of error versus number of users.

{Case 1} means short code with data bits and code chips not synchronous with respect to each other, no fading environment, no usage of Walsh functions and no fading link.

{Case 2} means short code with data bits and code chips not synchronous with respect to each other, no fading environment, 4 Walsh function groups and no fading link.

{Case 3} means short code with data bits and code chips not synchronous with respect to each other, no fading environment, 8 Walsh function groups and no fading link.

{Case 4} means short code with data bits and code chips not synchronous with respect to each other, no fading environment, 12 Walsh function groups and no fading link.

{Case 5} means short code with data bits and code chips not synchronous with respect to each other, no fading environment, 16 Walsh function groups and no fading link.

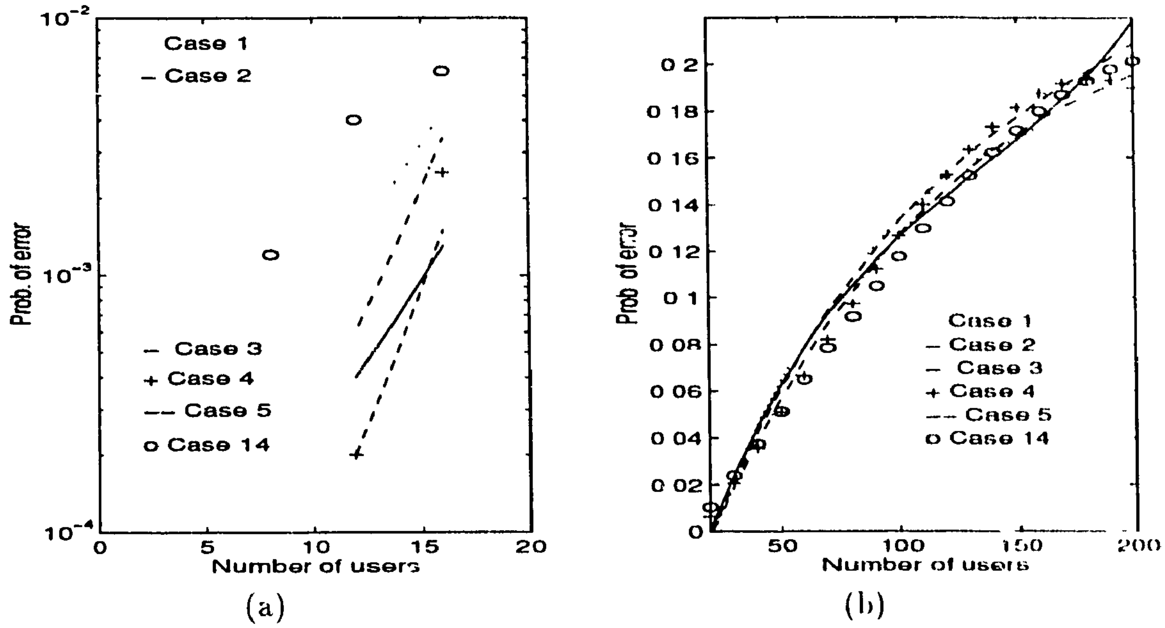


Figure 4.5: Probability of error versus number of users.

{Case 1} means short code with data bits and code chips not synchronous with respect to each other, no fading environment, no usage of Walsh functions and no fading link.

{Case 2} means short code with data bits and code chips not synchronous with respect to each other, no fading environment, 4 Walsh function groups and no fading link.

{Case 3} means short code with data bits and code chips not synchronous with respect to each other, no fading environment, 8 Walsh function groups and no fading link.

{Case 4} means short code with data bits and code chips not synchronous with respect to each other, no fading environment, 12 Walsh function groups and no fading link.

{Case 5} means short code with data bits and code chips not synchronous with respect to each other, no fading environment, 16 Walsh function groups and no fading link.

{Case 14} means long code with data bits and code chips synchronous with respect to each other, no fading environment, no usage of Walsh functions and no fading link.

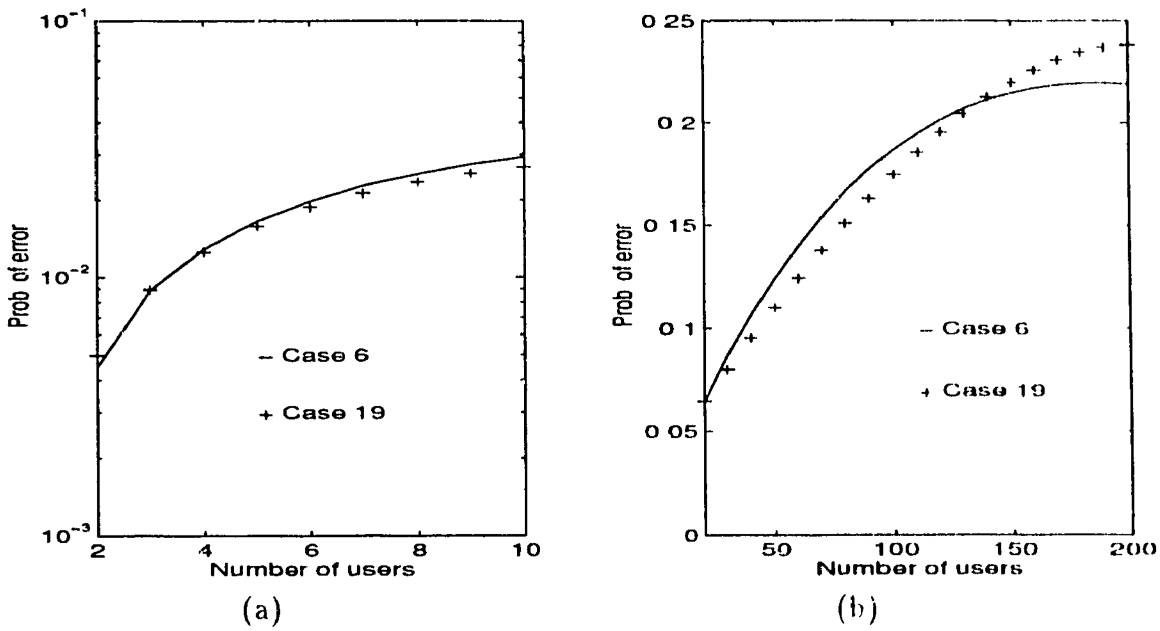


Figure 4.6: Probability of error versus number of users.
{Case 6} means short code with data bits and code chips not synchronous with respect to each other, Rayleigh flat fading, up link and no usage of Walsh functions. {Case 19} means long code with data bits and code chips synchronous with respect to each other, Rayleigh flat fading, up link and no usage of Walsh functions.

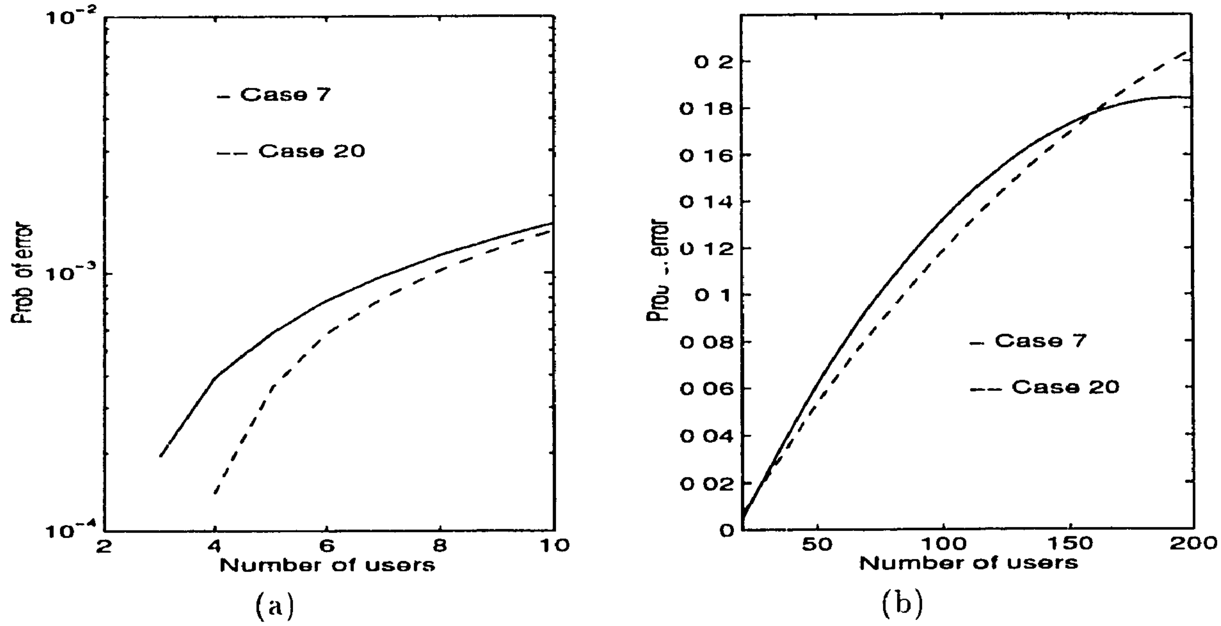
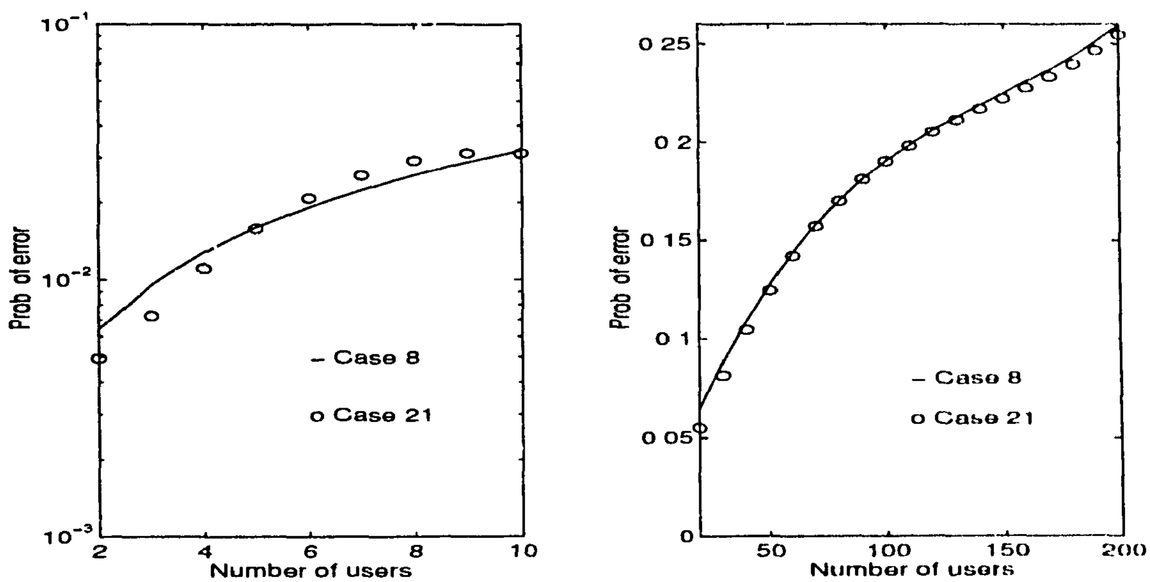


Figure 4.7: Probability of error versus number of users.
{Case 7} means short code with data bits and code chips not synchronous with respect to each other, Rayleigh flat fading, down link and no usage of Walsh functions.
{Case 20} means long code with data bits and code chips synchronous with respect to each other, Rayleigh flat fading, down link and no usage of Walsh functions.



(a) (b)

Figure 4.8: Probability of error versus number of users.
{Case 8} means short code with data bits and code chips not synchronous with respect to each other, Rayleigh frequency selective fading, up link and no usage of Walsh functions.
{Case 21} means long code with data bits and code chips synchronous with respect to each other, Rayleigh frequency selective fading, up link and no usage of Walsh functions.

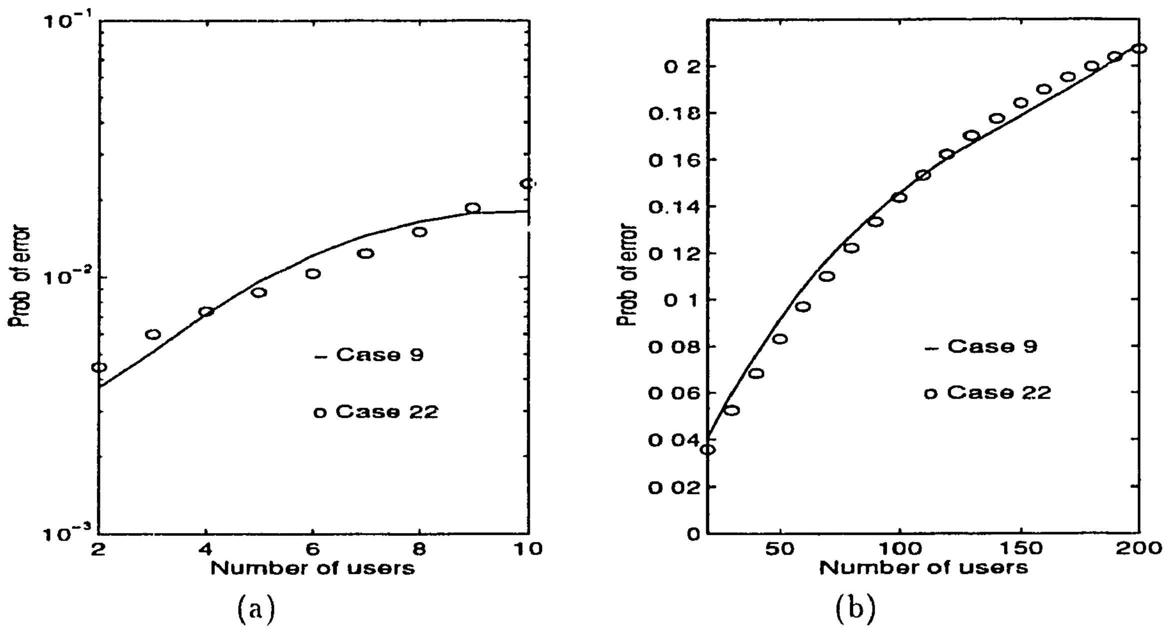


Figure 4.9: Probability of error versus number of users.

{Case 9} means short code with data bits and code chips not synchronous with respect to each other, Rayleigh frequency selective fading, down link and no usage of Walsh functions.

{Case 22} means long code with data bits and code chips synchronous with respect to each other, Rayleigh frequency selective fading, down link and no usage of Walsh functions.

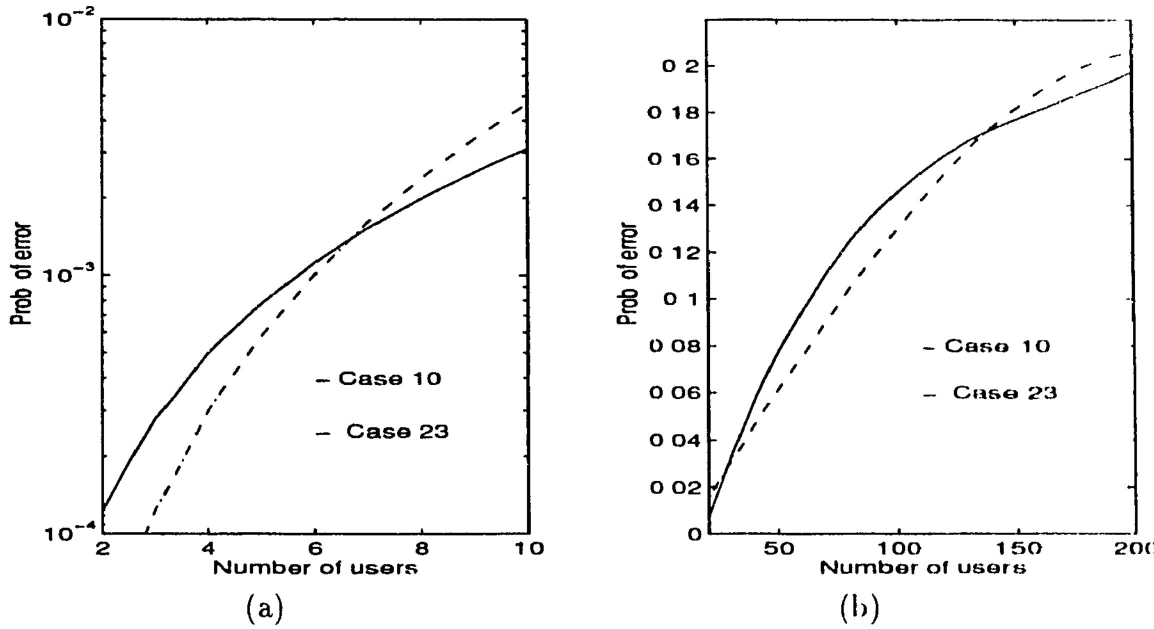
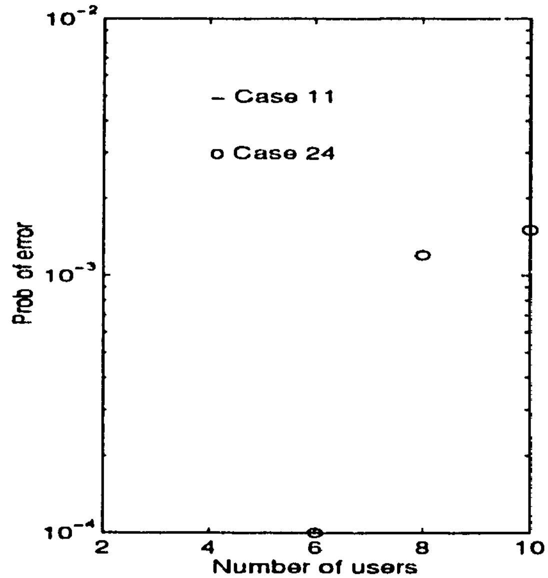
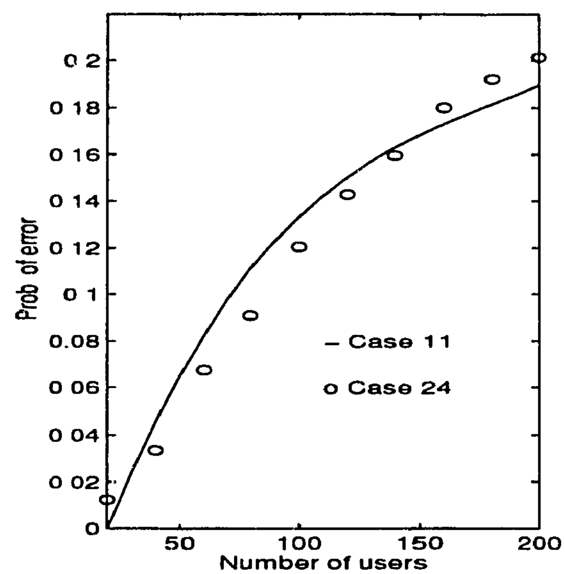


Figure 4.10: Probability of error versus number of users.
{Case 10} means short code with data bits and code chips not synchronous with respect to each other, Rician flat fading, up link and no usage of Walsh functions.
{Case 23} means long code with data bits and code chips synchronous with respect to each other, Rician flat fading, up link and no usage of Walsh functions.

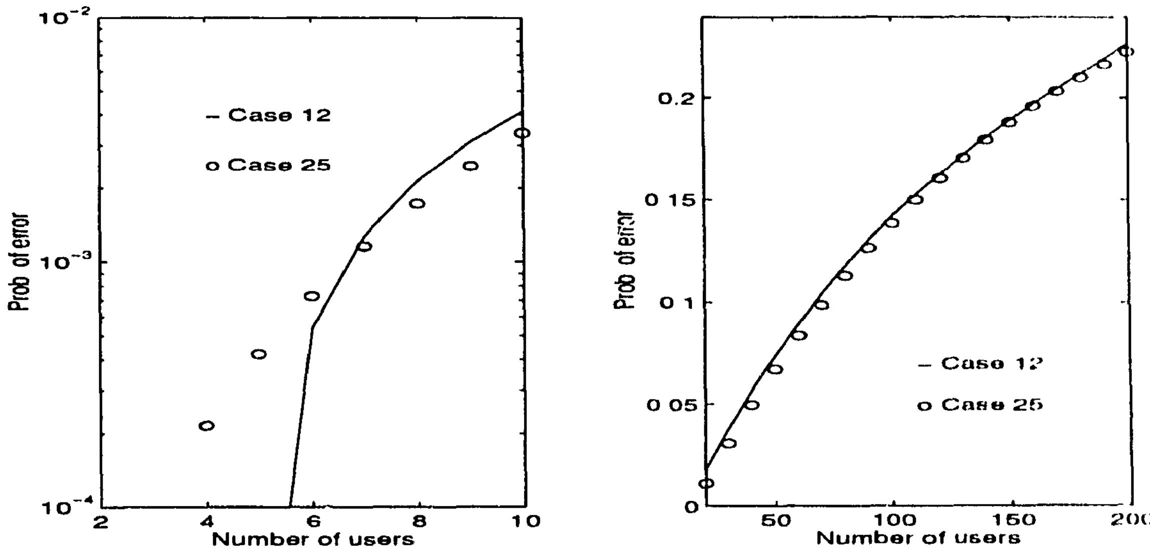


(a)



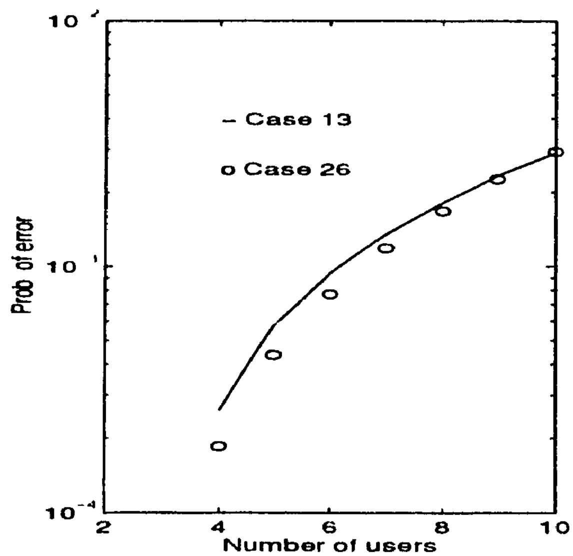
(b)

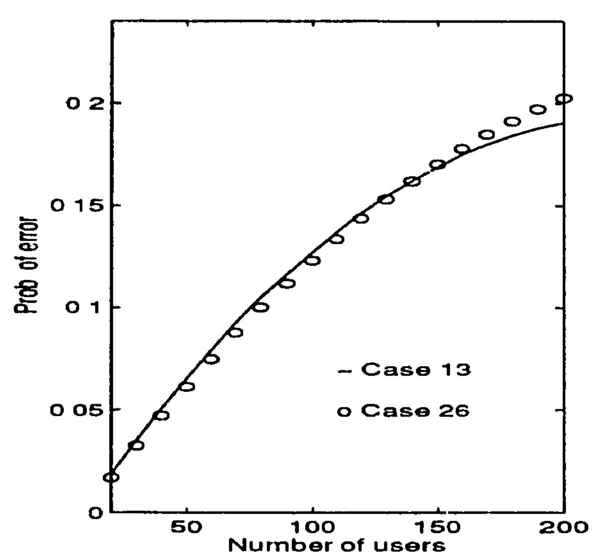
Figure 4.11: Probability of error versus number of users.
{Case 11} means short code with data bits and code chips not synchronous with respect to each other, Rician flat fading, down link and no usage of Walsh functions.
{Case 24} means long code with data bits and code chips synchronous with respect to each other, Rician flat fading, down link and no usage of Walsh functions.



(a) (b)

Figure 4.12: Probability of error versus number of users.
{Case 12} means short code with data bits and code chips not synchronous with respect to each other, Rician frequency selective fading, up link and no usage of Walsh functions.
{Case 25} means long code with data bits and code chips synchronous with respect to each other, Rician frequency selective fading, up link and no usage of Walsh functions.





(a) (b)

Figure 4.13: Probability of error versus number of users.
{Case 13} means short code with data bits and code chips not synchronous with respect to each other, Rician frequency selective fading, down link and no usage of Walsh functions.
{Case 26} means long code with data bits and code chips synchronous with respect to each other, Rician frequency selective fading, down link and no usage of Walsh functions.

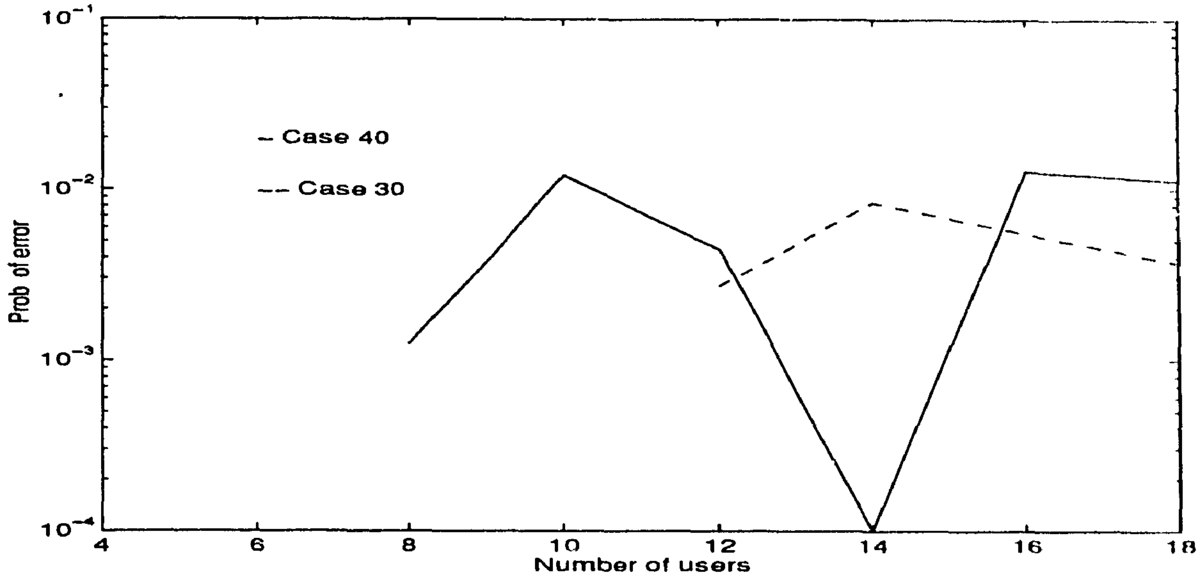


Figure 4.14: Probability of error versus number of users.
{Case 40} means long code with data bits and code chips synchronous with respect to each other, Shadow, Gaussian noise, 4 Walsh functions groups and no fading link. {Case 30} means short code with data bits and code chips not synchronous with respect to each other, Shadow, Gaussian noise, 4 Walsh functions groups and no fading link.

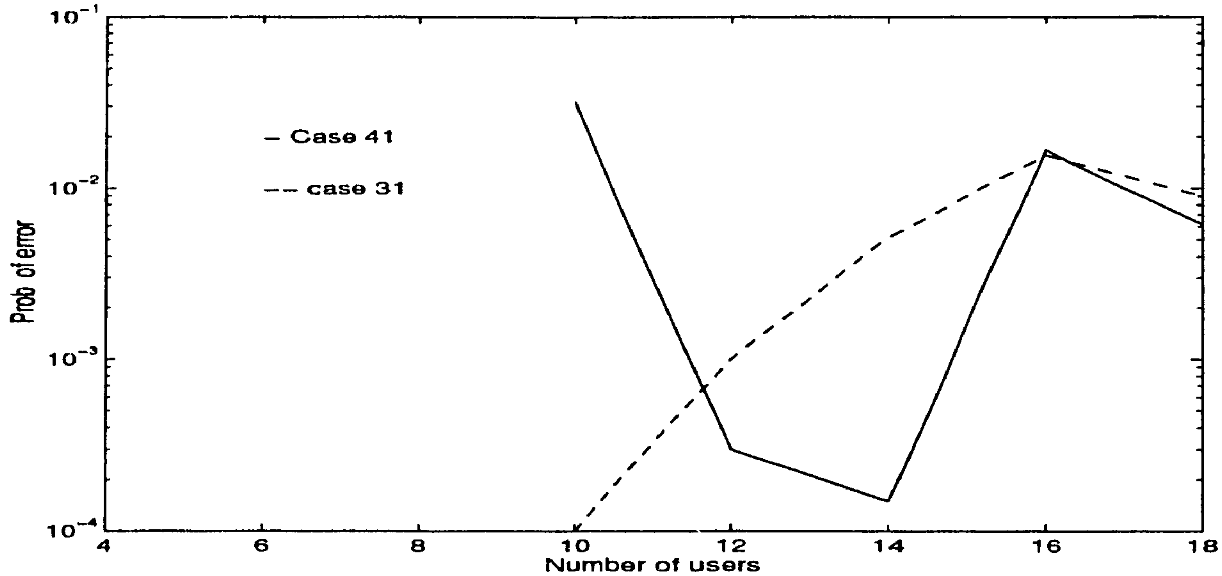


Figure 4.15: Probability of error versus number of users.
{Case 41} means long code with data bits and code chips synchronous with respect to each other, Shadow, Gaussian noise, Power control error, 4 Walsh functions groups and no fading link.
{Case 31} means short code with data bits and code chips not synchronous with respect to each other, Shadow, Gaussian noise, Power control error, 4 Walsh functions groups and no fading link.

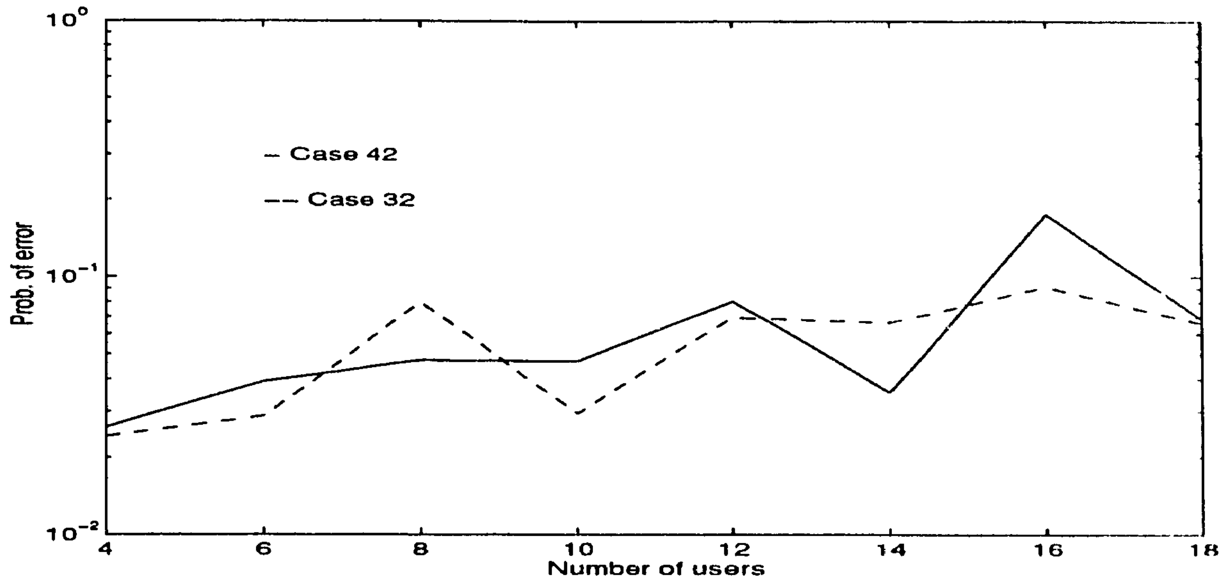


Figure 4.16: Probability of error versus number of users.
{Case 42} means long code with data bits and code chips synchronous with respect to each other, Shadow, Gaussian noise, Rayleigh flat fading, 4 Walsh functions groups and uplink.
{Case 32} means short code with data bits and code chips not synchronous with respect to each other, Shadow, Gaussian noise, Rayleigh flat fading, 4 Walsh functions groups and uplink.

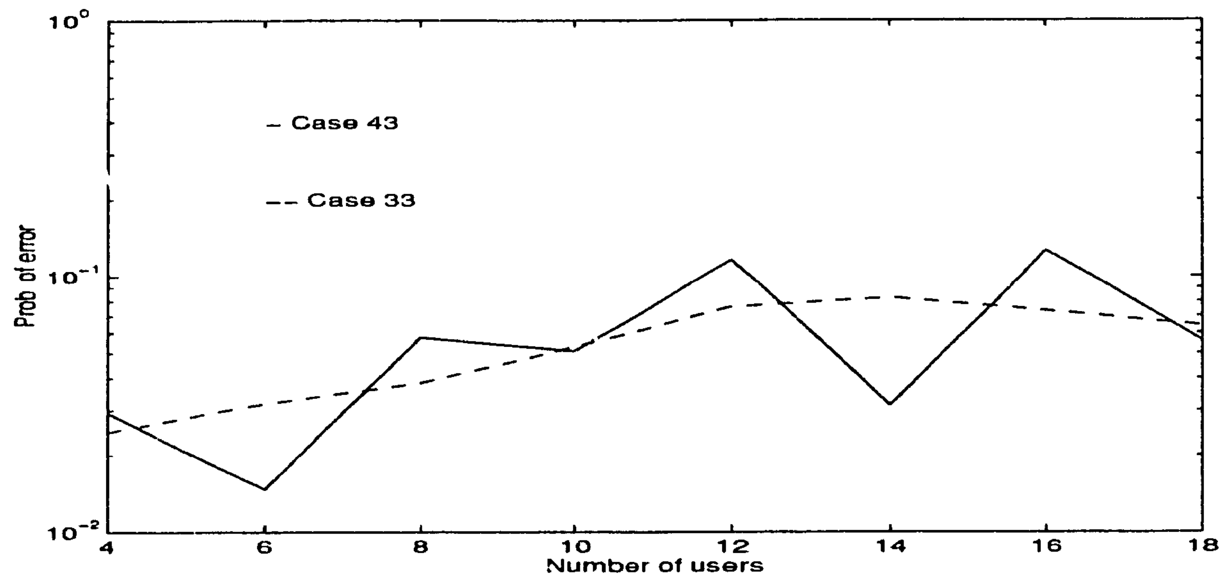


Figure 4.17: Probability of error versus number of users.
{Case 43} means long code with data bits and code chips synchronous with respect to each other, Shadow, Gaussian noise, Power control error, Rayleigh flat fading, 4 Walsh functions groups and uplink.
{Case 33} means short code with data bits and code chips not synchronous with respect to each other, Shadow, Gaussian noise, Power control error, Rayleigh flat fading, 4 Walsh functions groups an l uplink.

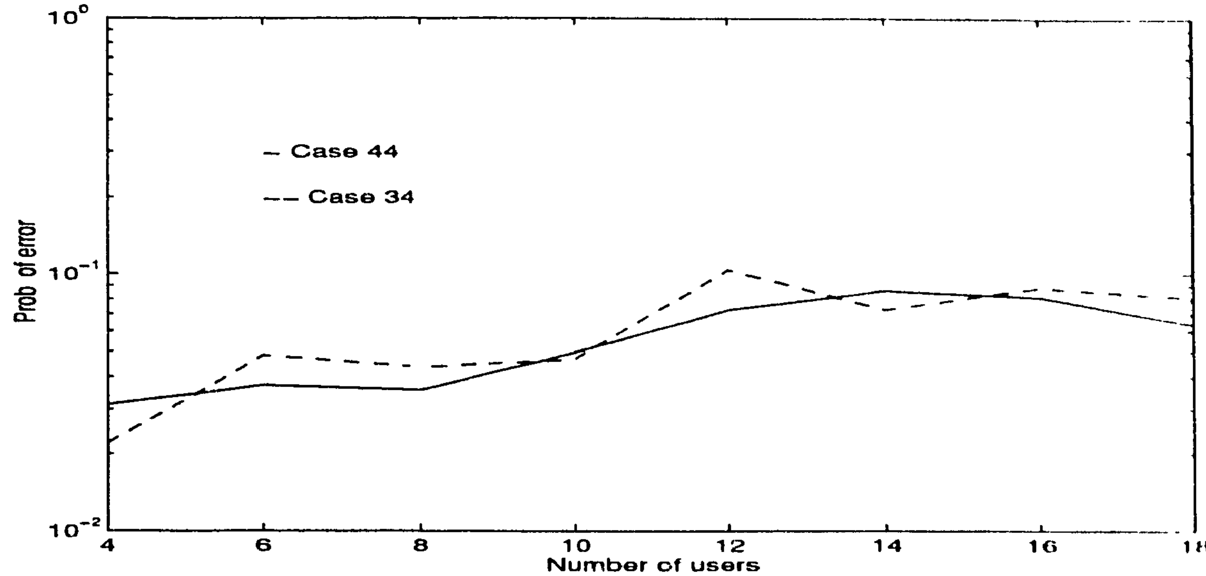


Figure 4.18: Probability of error versus number of users.
{Case 44} means long code with data bits and code chips synchronous with respect to each other, Shadow, Gaussian noise, Rayleigh frequency selective fading, 4 Walsh functions groups and uplink.
{Case 34} means short code with data bits and code chips not synchronous with respect to each other, Shadow, Gaussian noise, Rayleigh frequency selective fading, 4 Walsh functions groups and uplink.

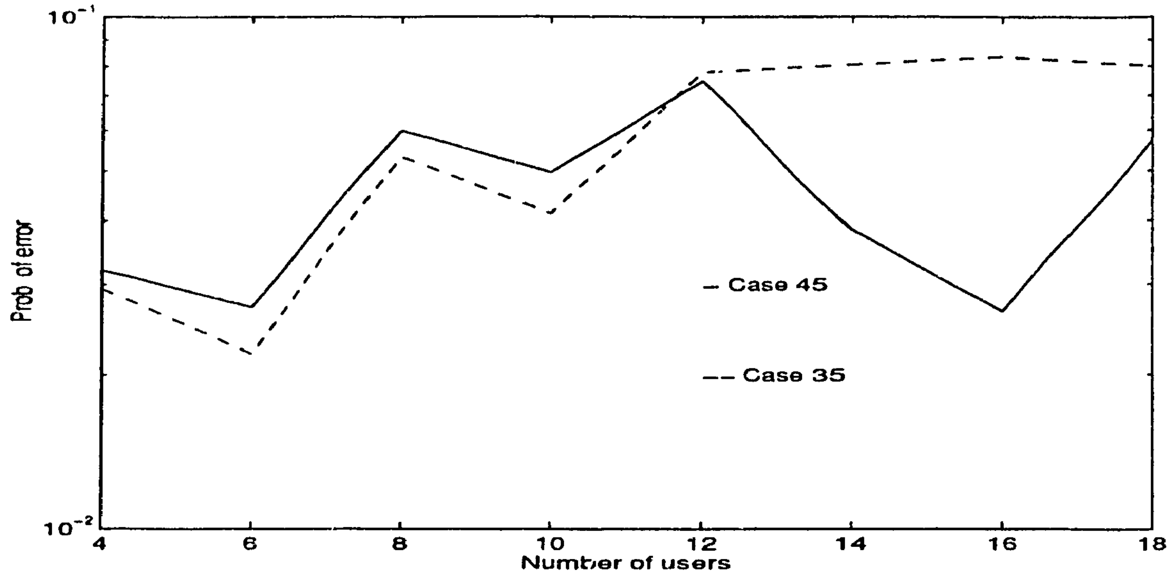


Figure 4.19: Probability of error versus number of users.
{Case 45} means long code with data bits and code chips synchronous with respect to each other, Shadow, Gaussian noise, Power control error, Rayleigh frequency selective fading, 4 Walsh functions groups and uplink.
{Case 35} means short code with data bits and code chips not synchronous with respect to each other, Shadow, Gaussian noise, Power control error. Rayleigh frequency selective fading, 4 Walsh functions groups and uplink.

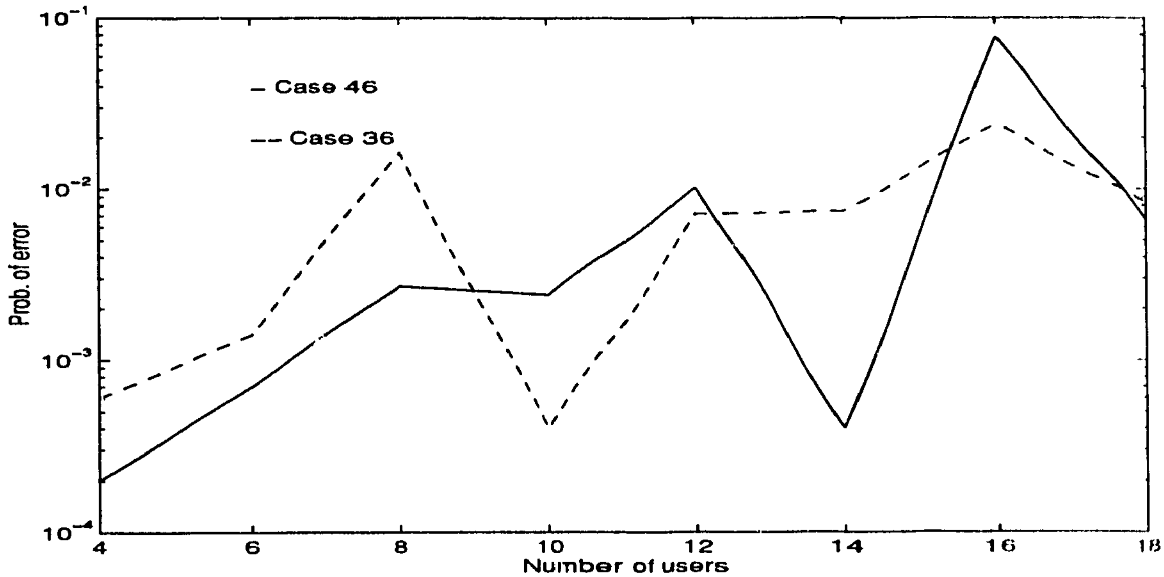


Figure 4.20: Probability of error versus number of users.
{Case 46} means long code with data bits and code chips synchronous with respect to each other, Shadow, Gaussian noise, Rician flat fading, 4 Walsh functions groups and uplink.
{Case 36} means short code with data bits and code chips not synchronous with respect to each other, Shadow, Gaussian noise, Rician flat fading, 4 Walsh functions groups and uplink.

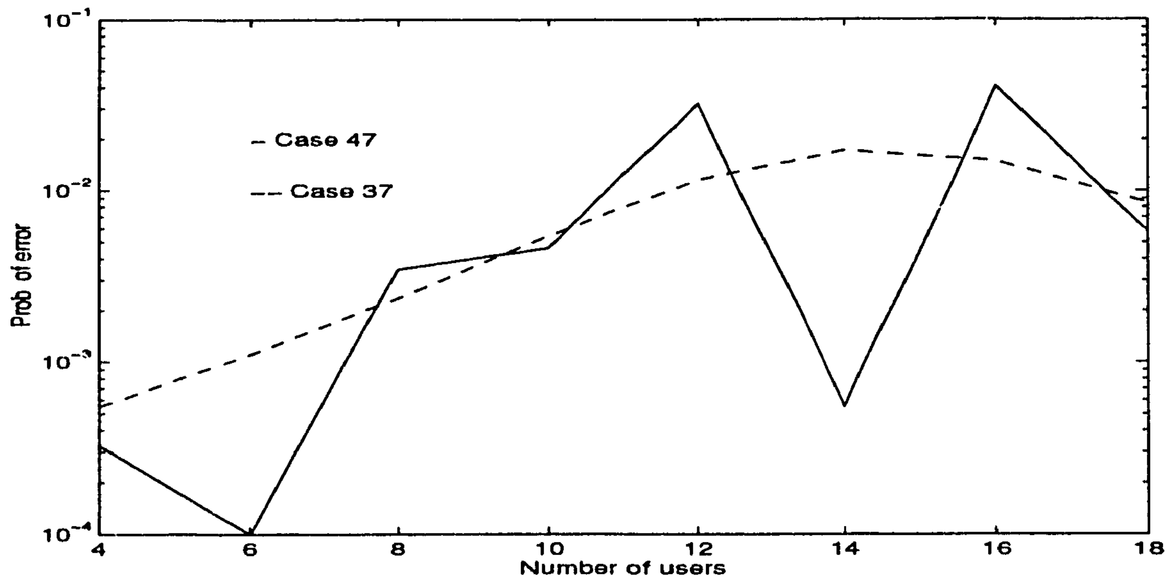


Figure 4.21: Probability of error versus number of users.
{Case 47} means long code with data bits and code chips synchronous with respect to each other, Shadow, Gaussian noise, Power control error, Rician flat fading,4 Walsh functions groups and uplink.
{Case 37} means short code with data bits and code chips not synchronous with respect to each other, Shadow, Gaussian noise, Power control error, Rician flat fading,4 Walsh functions groups and uplink.

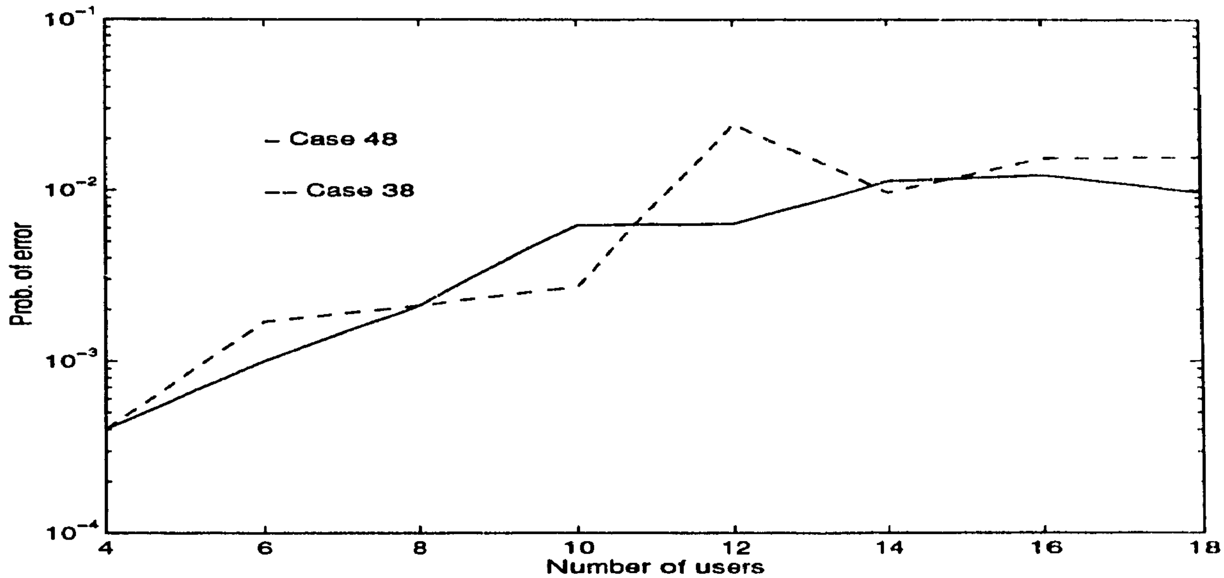


Figure 4.22: Probability of error versus number of users.
{Case 48} means long code with data bits and code chips synchronous with respect to each other, Shadow, Gaussian noise, Rician frequency selective fading, 4 Walsh functions groups and uplink.
{Case 38} means short code with data bits and code chips not synchronous with respect to each other, Shadow, Gaussian noise, Rician frequency selective fading, 4 Walsh functions groups and uplink.

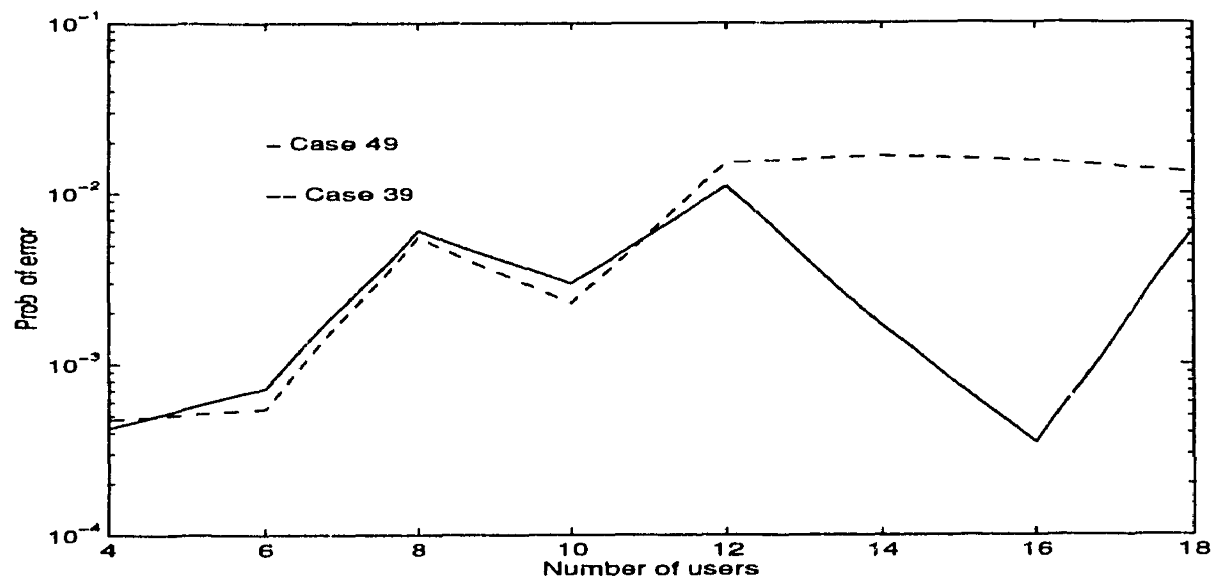


Figure 4.23: Probability of error versus number of users.
{Case 49} means long code with data bits and code chips synchronous with respect to each other, Shadow, Gaussian noise, Power control error, Rician frequency selective fading, 4 Walsh functions groups and uplink.
{Case 39} means short code with data bits and code chips not synchronous with respect to each other, Shadow, Gaussian noise, Power control error, Rician frequency selective fading, 4 Walsh functions groups and uplink.

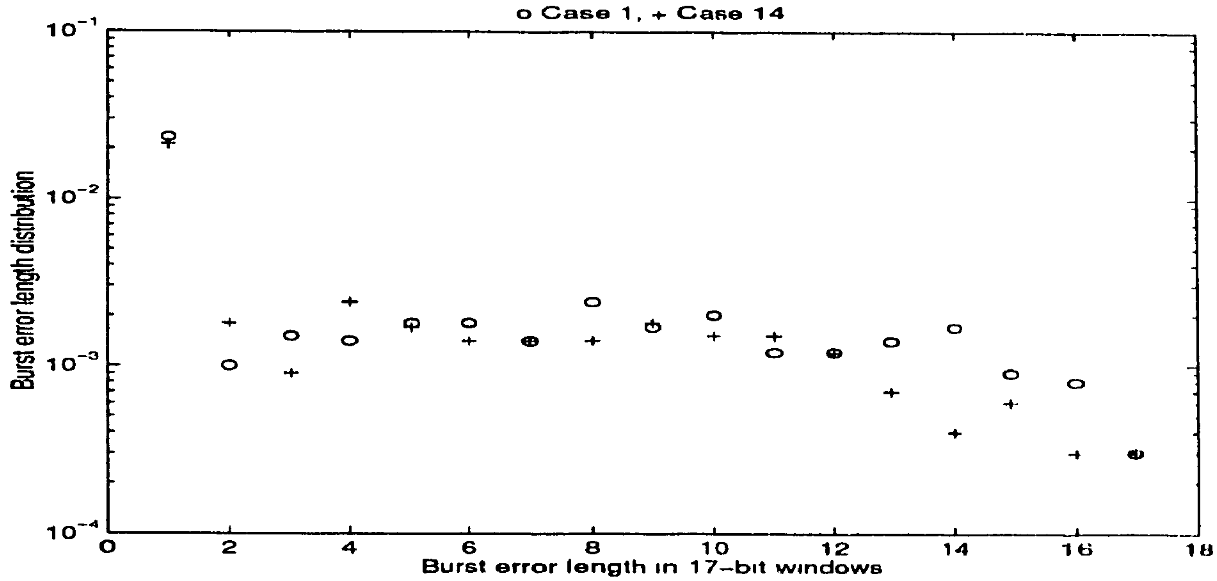


Figure 4.24: Burst error length distribution versus burst error length for 60 users in 17-bit data blocks.
{Case 1} means short code with data bits and code chips not synchronous with respect to each other, no fading environment, no usage of Walsh functions and no fading link.
{Case 14} means long code with data bits and code chips synchronous with respect to each other, no fading environment, no usage of Walsh functions and no fading link.

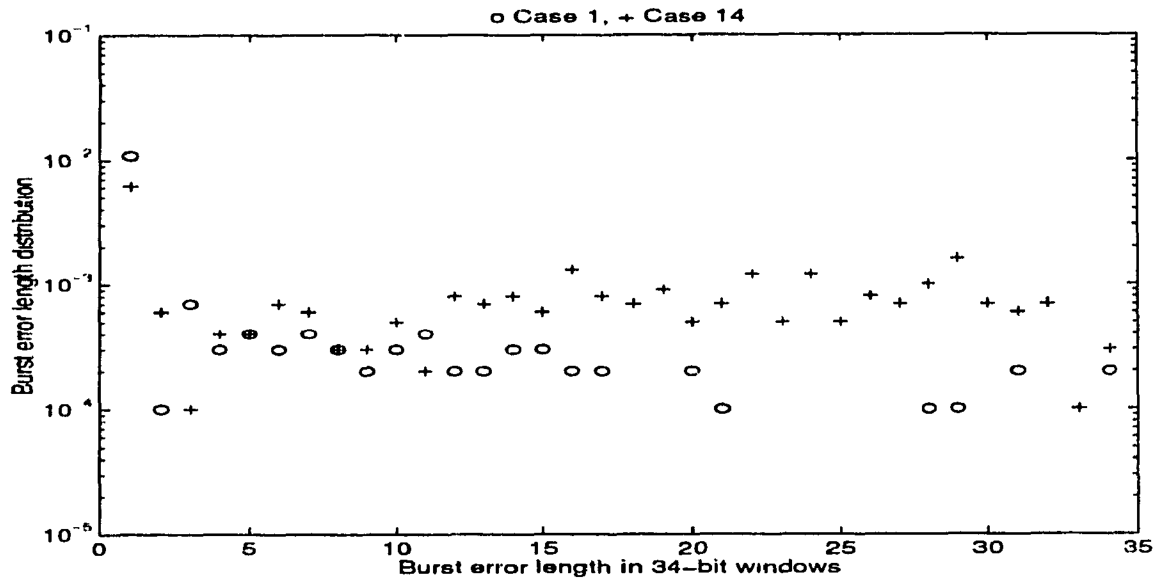


Figure 4.25: Burst error length distribution versus burst error length for 60 users in 34-bit data blocks.
{Case 1} means short code with data bits and code chips not synchronous with respect to each other, no fading environment, no usage of Walsh functions and no fading link.
{Case 14} means long code with data bits and code chips synchronous with respect to each other, no fading environment, no usage of Walsh functions and no fading link.

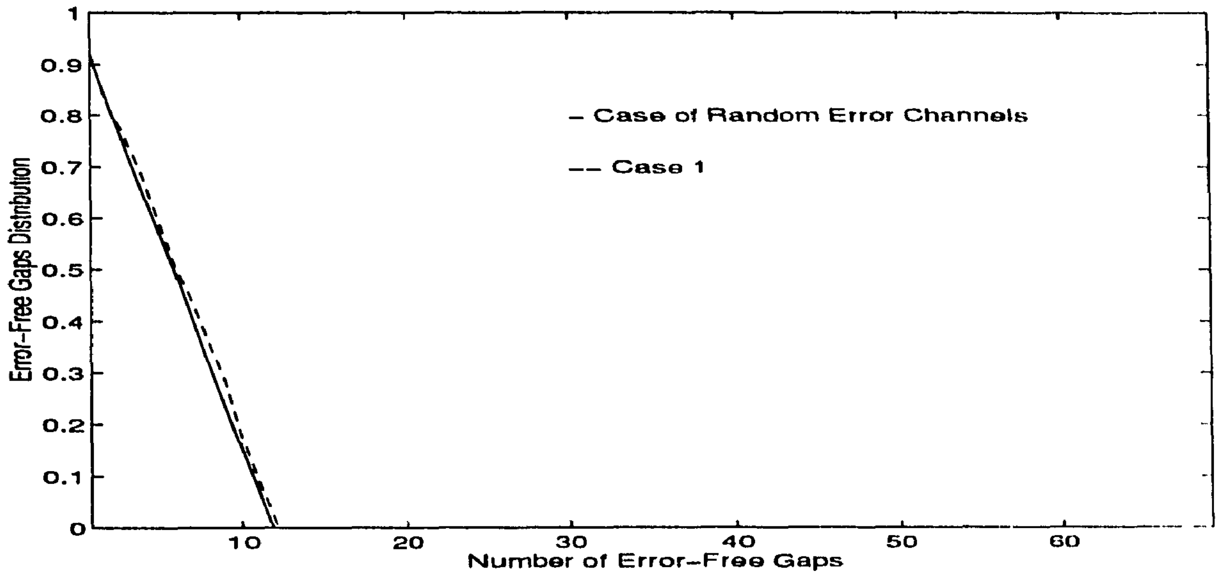


Figure 4.26: Comparing Case 1 to Random error channels for 60 users. {Case 1} means short code with data bits and code chips not synchronous with respect to each other, no fading environment, no usage of Walsh functions and no fading link.

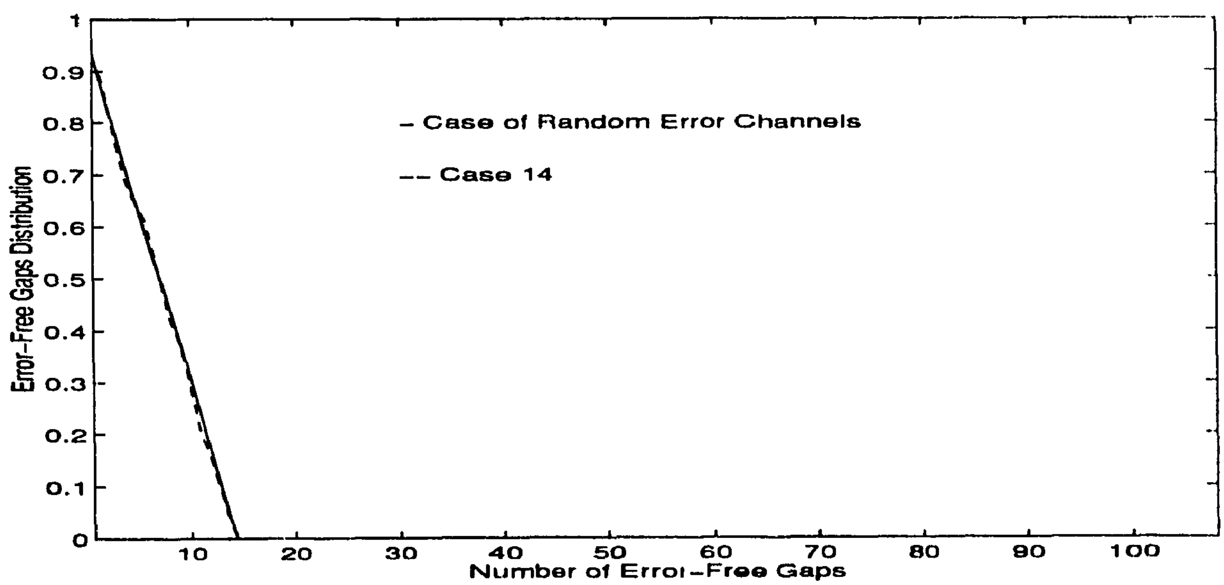


Figure 4.27: Comparing Case 14 to Random error channels for 60 users. {Case 14} means long code with data bits and code chips synchronous with respect to each other, no fading environment, no usage of Walsh functions and no fading link.

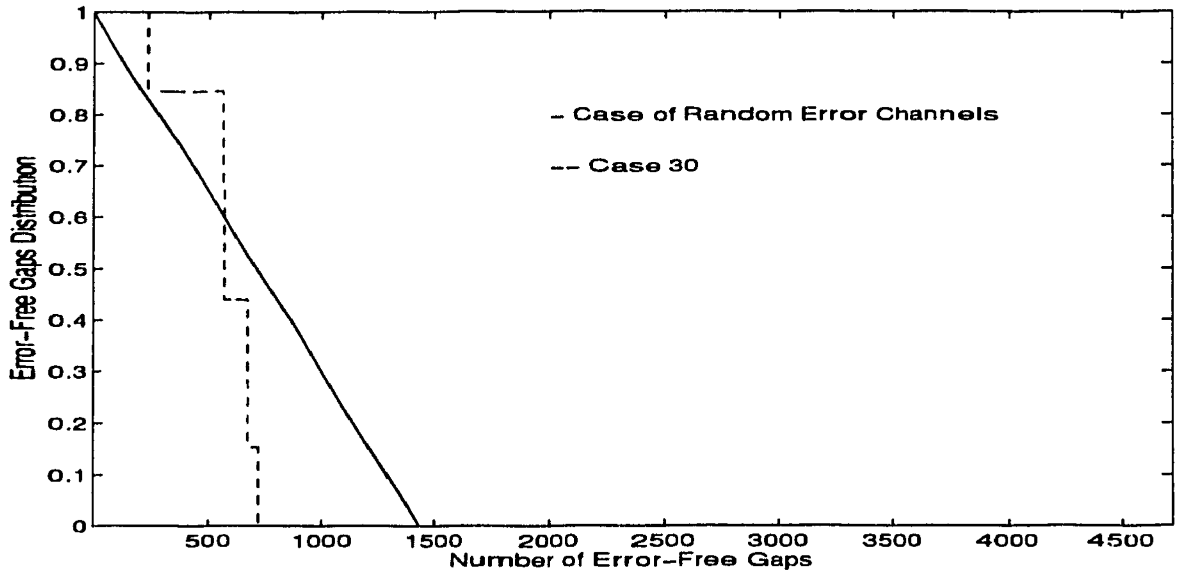


Figure 4.28: Comparing Case 30 to Random error channels for 16 users. {Case 30} means short code with data bits and code chips not synchronous with respect to each other, Shadow, Gaussian noise, 4 Walsh functions groups and no fading link.

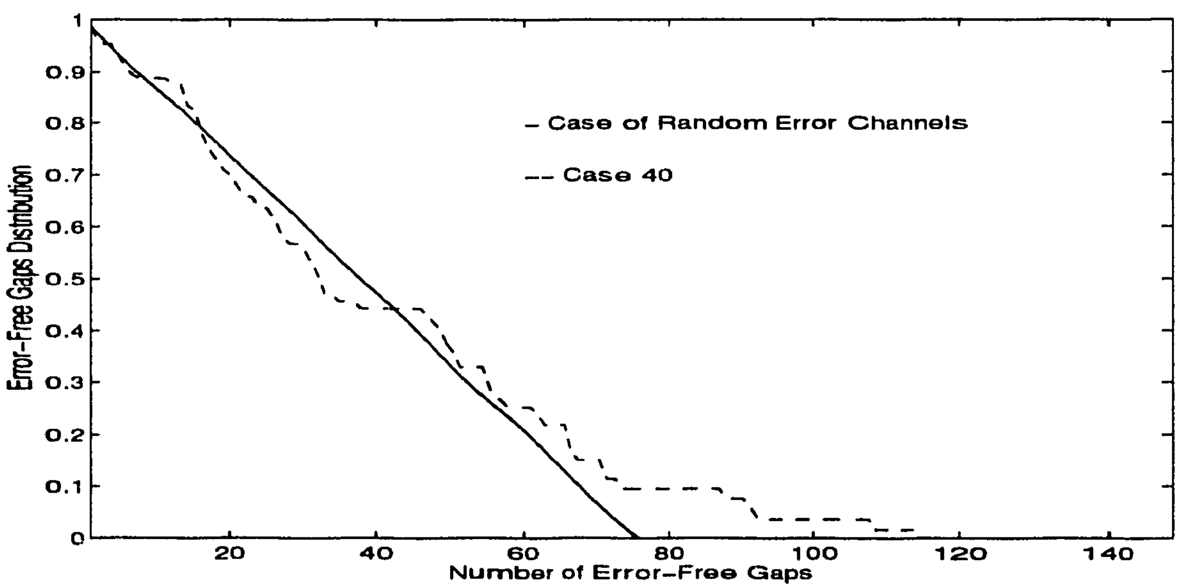


Figure 4.29: Comparing Case 40 to Random error channels for 16 users. {Case 40} means long code with data bits and code chips synchronous with respect to each other, Shadow, Gaussian noise, 4 Walsh functions groups and no fading link.

4.3 Numerical Results

- Multiaccsess interference channels (Figs.4.2 - 4.5):

  For number of users less than twenty, we have seen that short codes outperform long codes due to the fact that short codes are correlated but long codes are partially correlated. however, for number of users greater than twenty long codes yield better results since the effect of many users sharing the same code (short codes case) dominates.

  It is observed that short Gold codes (case 29) and short codes with changing initial state at each data bit (case 28) have shown good performance comparing to long codes due to minimal crosscorrelation of Gold codes, and the fact that changing initial state at each data bit for short codes weaken strongly the effect of many users sharing the same code.

  It is shown that using Walsh functions can reduce the interference among users and improve bit error rate for short codes.

  In terms of eror-free gap distribution (Figs.4.26 - 4.27), we have observed that long codes and short codes behave the same like random error channels and have the same number of error-free gaps.

- Multipath fading channels (Figs.4.6 - 4.13):

  In the case of Raileigh and Rician flat fading, long codes have shown to outperform short codes. However, in the case of Rayleigh and Rician frequency selective fading they have shown to have approximately the same performance due to the fact that selective fading gives the system the ability to resist the effects of multipath fading.

- Satellite channels (Figs.4.14 - 4.23):

  It is observed that long codes and short codes have almost the same performance and introducing power control error has not changed the performance of long and short codes too much, since the standard deviation of shadow is 5 dB and that of power error control is 0.5 dB.

  Comparing long and short codes in terms of error-free gap distribution (Figs.4.28

- 4.29), we have seen that short codes are more deviated from random error channel and have larger number of error-free gaps than long codes which indicates that short codes have less burst error lengths than long codes.

Bibliography

[1] L. N. Kanal and A. R. K. Sastry, "*Models for channels with memory and their applications to error control*", *Proc. IEEE, vol. 66, no. 7, July 1978.*

[2] J.-P.A. Adoul, "*Error intervals and cluster density in channel modeling*", *IEEE Trans. Inform. Theory, vol. IT-20, pp. 125-129, Jan. 1974.*

Chapter 5

Conclusions

Short codes are known to have easier acquisition and less synchronization time and implementation complexity than long codes.

Different cases of orthogonal short and long codes in CDMA networks have been simulated and studied.

In chapter two, a communication system for orthogonal short codes is proposed and taken as a model to simulate synchronous short codes, asynchronous short codes, synchronous short codes with changing initial state at each data bit, and finally synchronous short Gold codes.

In chapter three, land mobile satellite channels are described and the need for power control is discussed.

In chapter four, results obtained from simulation have shown that,

- For channels with only multiaccess interference effects, long codes yield better results for number of users less than 140 (our operating range). However, short codes outperform long codes for number of users less than 20. It has been observed that changing initial state at each data bit for synchronous short codes can improve the performance since it makes the effects of users sharing the same code very small. Short Gold codes show that they have approximately the same performance as long codes within our operating range. Short codes have better burst error distribution than long codes in 34-bit windows. However, long codes yield better results in 17-bit windows.

- For channels with multipath fading, it is shown that in the cases of Rayleigh and Rician flat fading long codes outperform asynchronous short concatenated codes within our operating range. However, in the cases of Rayleigh and Rician frequency selective fading they offer the same performance.

- In satellite channels and power control errors, it is shown that intoducing Walsh functions into the system has made the performance of short codes and long codes almost the same.

- In the case of shadow short codes have shown to have less number of error bursts than long codes.

Further works could be carried out to simulate the above stated cases for larger number of short codes and data bits. Care should be taken in selecting different short codes for spreading. Short codes should have minimal crosscorrelation among themselves, otherwise probability of error will increase. It is expected that, by increasing number of short codes while keeping minimal crosscorrelation among them, the probability of error would decrease.

Tests could be performed to evaluate the performance of the above cases when we include the effects of propagation constant on the received signal path loss. It is expected that using channel coding would decrease the probability of error of the different simulated cases.

Appendix

A.1 Source code for synchronous short codes with changing initial states

```
* Program 'dspread3.f'(Short Concatenated Codes)
* with Changing initial state at each databit for each user.
REAL SUMSIG(0:126),XDAT(200),SINIT(20)
INTEGER DS,JX,IC(200),EFG(1000),vase(10000)
COMMON SINIT(20),XINIT(200,20),NSTAG,CON(20),FEED(200,20,9)
COMMON INT,XN1(200,20)
NSTAG=7
NCODE=9
NCHIP=127
NBT=10000
NSIM=NCHIP*NBT
INT=1
open (unit=7,file='es3',status='new')
write(7,*) 'RESULTS of SHORT CONC.CODES program'
write(7,*) 'with Changing initial state at each databit'
* Calculating Prob.of error for different no.of users
DO 123 NUSER=20,200,20
write(7,*)
write(7,341) NUSER
```

```
 341 format(1x,'No.of users=',i3)
        ISEED=156
DO 232 I1=1,1000
  EFG(I1)=0
 232 CONTINUE
KK=0
ERR=0
NBIT=0
* Assigning different phase for each user
DO 111 I4=1,NUSER
  IF (I4.GT.1) THEN
    DO 222 I6=1,NSTAG
      XN1(I4,I6)=XN1(I4-1,I6)
      XINIT(I4,I6)=XINIT(I4-1,I6)
 222     CONTINUE
  ENDIF
  I6=1
 2   XN1(I4,I6)=XN1(I4,I6)+1
  XINIT(I4,I6)=XINIT(I4,I6)+1
  IF (MOD(XN1(I4,I6),2.).EQ.0) THEN
    XN1(I4,I6)=0.
    XINIT(I4,I6)=0.
    I6=I6+1
    GO TO 2
  ENDIF
 111 CONTINUE
* Installing feedback connections of the polynomial
DO 216 I8=1,NUSER
  DO 218 I9=1,NSTAG
    DO 219 I10=1,NCODE
      FEED(I8,I9,I10)=0.
```

```
         IF(I9.EQ.1) FEED(I8,I9,I10)=1.
IF((I9.EQ.2).AND.(I10.EQ.4)) FEED(I8,I9,I10)=1.
         IF((I9.EQ.2).AND.(I10.EQ.6)) FEED(I8,I9,I10)=1.
IF((I9.EQ.2).AND.((I10.EQ.8).OR.(I10.EQ.9))) FEED(I8,I9,I10)=1.
IF((I9.EQ.3).AND.(I10.EQ.4)) FEED(I8,I9,I10)=1.
IF((I9.EQ.3).AND.((I10.EQ.5).OR.(I10.EQ.9))) FEED(I8,I9,I10)=1.
IF((I9.EQ.4).AND.((I10.EQ.3).OR.(I10.EQ.4))) FEED(I8,I9,I10)=1.
IF((I9.EQ.4).AND.((I10.EQ.5).OR.(I10.EQ.6))) FEED(I8,I9,I10)=1.
  IF((I9.EQ.5).AND.((I10.EQ.1).OR.(I10.EQ.2).OR.(I10.EQ.3))) THEN
          FEED(I8,I9,I10)=1.
ENDIF
IF((I9.EQ.5).AND.((I10.EQ.5).OR.(I10.EQ.8))) FEED(I8,I9,I10)=1.
IF((I9.EQ.6).AND.((I10.EQ.2).OR.(I10.EQ.3).OR.(I10.EQ.4))) THEN
          FEED(I8,I9,I10)=1.
ENDIF
         IF((I9.EQ.6).AND.(I10.EQ.5))  FEED(I8,I9,I10)=1.
         IF((I9.EQ.6).AND.((I10.EQ.6).OR.(I10.EQ.9))) THEN
          FEED(I8,I9,I10)=1.
ENDIF
IF((I9.EQ.7).AND.((I10.EQ.2).OR.(I10.EQ.4).OR.(I10.EQ.5))) THEN
          FEED(I8,I9,I10)=1.
ENDIF
IF((I9.EQ.7).AND.((I10.EQ.7).OR.(I10.EQ.8))) FEED(I8,I9,I10)=1.
219         CONTINUE
218    CONTINUE
216 CONTINUE

I=1
116 JX=MOD(I,NCHIP)
SUMSIG(JX)=0.
* Generation of a random data bit and different phase for NUSERs every 127
```

```
* chips
IF(JX.NE.1) GO TO 40
DO 70 J=1,NUSER
    CALL UNIFORM (ISEED,XX)
  IF (XX.LE.0.5) THEN
    XDAT(J)=-1.
  ELSE
    XDAT(J)=1.
  ENDIF
  IF(XX.LE.0.11) IC(J)=1
  IF(XX.GT..11.AND.XX.LE.0.22) IC(J)=2
  IF(XX.GT..22.AND.XX.LE.0.33) IC(J)=3
  IF(XX.GT..33.AND.XX.LE.0.44) IC(J)=4
  IF(XX.GT..44.AND.XX.LE.0.55) IC(J)=5
  IF(XX.GT..55.AND.XX.LE.0.66) IC(J)=6
  IF(XX.GT..66.AND.XX.LE.0.77) IC(J)=7
  IF(XX.GT..77.AND.XX.LE.0.88) IC(J)=8
  IF(XX.GT..88.AND.XX.LE.1.0) IC(J)=9
  IF(XX.LE.0.14) I6=1
  IF(XX.GT..14.AND.XX.LE.0.28) I6=2
  IF(XX.GT..28.AND.XX.LE.0.42) I6=3
  IF(XX.GT..42.AND.XX.LE.0.56) I6=4
  IF(XX.GT..56.AND.XX.LE.0.7) I6=5
  IF(XX.GT..7.AND.XX.LE.0.84) I6=6
  IF(XX.GT..84.AND.XX.LE.1.0) I6=7
 43   XN1(J,I6)=XN1(J,I6)+1
  XINIT(J,I6)=XINIT(J,I6)+1
  IF(MOD(XN1(J,I6),2.).EQ.0) THEN
    XN1(J,I6)=0.
    XINIT(J,I6)=0.
    IF(I6.NE.7) THEN
```

```
        I6=I6+1
        GO TO 43
      ENDIF
    ENDIF
   70 CONTINUE
* Spreading with summation of multiusers
   40 DO 60 K=1,NUSER
    DO 34 IZ=1,NSTAG
      CON(IZ)=FEED(K,IZ,IC(K))
      SINIT(IZ)=XINIT(K,IZ)
   34   CONTINUE
    CALL PNCOD(K,DS)
    SUMSIG(JX)=SUMSIG(JX)+DS*XDAT(K)
   60 CONTINUE
IF (JX.NE.0) THEN
    I=I+1
    GO TO 116
ENDIF
* Calculating no. of data bits (for intended user)
   NBIT=NBIT+1
* Despread with 127 chip summation (for intended user)
    DO 68 IY=1,NSTAG
      CON(IY)=FEED(INT,IY,IC(INT))
      SINIT(IY)=XN1(INT,IY)
   68   CONTINUE
    SUMBIT=0.
    DO 140 J9=1,NCHIP
              l=mod(J9,NCHIP)
      CALL PNCOD(INT,DS)
      SUMBIT=SUMBIT+SUMSIG(l)*DS
   140   CONTINUE
```

```
* Phase of shift register after 127 chips (used for demodulation)
  DO 484 JR=1,NSTAG
    XN1(INT,JR)=XINIT(INT,JR)
 484   CONTINUE
* Quantization of recovered data bits
  IF (SUMBIT.GT.0.) THEN
    DET=1.
  ELSE
    DET=-1.
  ENDIF
* Calculation of prob. of error
  IF (DET.NE.XDAT(INT)) THEN
    ERR=ERR+1
    NBE=NBE+1
  ENDIF
* Could be used to determine #of consecutive errors
IF(DET.EQ.XDAT(INT)) THEN
  NBE=0
ENDIF
* Determining error free gaps
IF(DET.eq.XDAT(INT)) THEN
  KK=KK+1
  IF(NBIT.EQ.NBT) EFG(KK)=EFG(KK)+1
ELSE
  IF(KK.EQ.0) GO TO 3
  EFG(KK)=EFG(KK)+1
  KK=0
ENDIF
 3 vase(NBIT)=KK
  I=I+1
IF (I.LE.NSIM) GO TO 116
```

```
 PROBER=ERR/NBT
 write(7,342) PROBER
  342 format(1x,'*Prob.of error=',f10.8)
 max1=vase(1)
 do 211 NBITT=2,NBT
   if (vase(NBITT).gt.max1) max1=vase(NBITT)
  211 continue
 DO 4 KK=1,max1
   write(7,*) ('ERROR FREE GAPS:')
   write(7,343) KK,EFG(KK)
  343   format(1x,'EFG(',i4,')=',i5)
  4 CONTINUE
  123 CONTINUE
 END
*----
 subroutine PNCOD(K,DS)
 integer K,DS,INT
 real SINIT(20),XINIT(200,20)
 common SINIT(20),XINIT(200,20),NSTAG,CON(20),FEED(200,20,9)
 common INT,XN1(200,20)
 sum=0.
 do 30 i=1,NSTAG
   sum=sum+CON(i)*SINIT(i)
  30 continue
 xsum=amod(sum,2.)
 if (SINIT(1).EQ.1.) then
   DS=1
 else
   DS=-1
 endif
 do 40 j=2,NSTAG
```

```
  SINIT(j-1)=SINIT(j)
 40 continue
SINIT(NSTAG)=xsum
* Storing the phase of shift register for calculating PNCOD chips after
* the first  chip
do 424 IL=1,NSTAG
  XINIT(K,IL)=SINIT(IL)
 424 continue
return
end
*----
subroutine UNIFORM(ISEED,RANDX)
integer ISEED
real RANDX
ISEED=2045*ISEED+1
ISEED=ISEED-(ISEED/1048576)*1048576
RANDX=real(ISEED+1)/1048577.0
return
end
```

A.2 Source code for synchronous short Gold codes

```
* Program 'gold2.f'(Short Gold Codes)
REAL SUMSIG(0:126),XDAT(200),SINIT1(20),SINIT2(20)
INTEGER DS,nDS,JX,EFG(1000),vase(10000)
INTEGER K,nDS1,nDS2,INT,NSTAG
real SINIT1(20),XN1(200,20),XINIT1(200,20)
real CON1(20),FEED1(200,20)
real SINIT2(20),XN2(200,20),XINIT2(200,20)
real CON2(20),FEED2(200,20)
common INT,NSTAG
```

```
 common SINIT1(20),XINIT1(200,20),CON1(20),FEED1(200,20)
 common XN1(200,20)
 common SINIT2(20),XINIT2(200,20),CON2(20),FEED2(200,20)
 common XN2(200,20)
 NSTAG=7
 NCODE=1
 NCHIP=127
 NBT=10000
 NSIM=NCHIP*NBT
 INT=1
        open (unit=117,file='eg2',status='new')
write(117,*) 'RESULTS of SHORT GOLD CODES program'
* Calculating Prob.of error for different no.of users
DO 123 NUSER=20,200,20
 write(117,*)
 write(117,341) NUSER
 341  format(1x,'No.of users=',i3)
        ISEED=156
DO 232 I1=1,1000
  EFG(I1)=0
 232 CONTINUE
KK=0
ERR=0
NBIT=0
* Assigning different phase for each user
DO 111 I4=1,NUSER
  IF (I4.GT.1) THEN
    DO 222 I6=1,NSTAG
      XN1(I4,I6)=XN1(I4-1,I6)
      XINIT1(I4,I6)=XINIT1(I4-1,I6)
      XN2(I4,I6)=XN2(I4-1,I6)
```

```
      XINIT2(I4,I6)=XINIT2(I4-1,I6)
 222     CONTINUE
  ENDIF
  I6=1
 2   XN1(I4,I6)=XN1(I4,I6)+1
  XINIT1(I4,I6)=XINIT1(I4,I6)+1
  XN2(I4,I6)=XN2(I4,I6)+1
  XINIT2(I4,I6)=XINIT2(I4,I6)+1
  IF (MOD(XN1(I4,I6),2.).EQ.0) THEN
    XN1(I4,I6)=0.
    XINIT1(I4,I6)=0.
    XN2(I4,I6)=0.
    XINIT2(I4,I6)=0.
    I6=I6+1
    GO TO 2
  ENDIF
 111 CONTINUE
* Installing feedback connections of the PREFERRED two polynomials
DO 216 I8=1,NUSER
  DO 218 I9=1,NSTAG
      FEED1(I8,I9)=0.
      FEED2(I8,I9)=0.
              IF(I9.EQ.1) THEN
        FEED1(I8,I9)=1.
        FEED2(I8,I9)=1.
      ENDIF
        IF(I9.EQ.5) THEN
                FEED1(I8,I9)=1.
                FEED2(I8,I9)=1.
      ENDIF
      IF(I9.EQ.6) FEED2(I8,I9)=1.
```

```
      IF(I9.EQ.7) FEED2(I8,I9)=1.
 218   CONTINUE
 216 CONTINUE

I=1
 116 JX=MOD(I,NCHIP)
SUMSIG(JX)=0.
* Generation of a random data bit for each user every 127 chips
IF(JX.NE.1) GO TO 40
DO 70 J=1,NUSER
    CALL UNIFORM (ISEED,XX)
  IF (XX.LE.0.5) THEN
    XDAT(J)=-1.
  ELSE
    XDAT(J)=1.
  ENDIF
* Changing the code randomly for each user at each data bit
  IF(XX.LE.0.14) I6=1
  IF(XX.GT..14.AND.XX.LE.0.28) I6=2
  IF(XX.GT..28.AND.XX.LE.0.42) I6=3
  IF(XX.GT..42.AND.XX.LE.0.56) I6=4
  IF(XX.GT..56.AND.XX.LE.0.7) I6=5
  IF(XX.GT..7.AND.XX.LE.0.84) I6=6
  IF(XX.GT..84.AND.XX.LE.1.0) I6=7
 43   XN2(J,I6)=XN2(J,I6)+1
  XINIT2(J,I6)=XINIT2(J,I6)+1
  IF(MOD(XN2(J,I6),2.).EQ.0) THEN
    XN2(J,I6)=0.
    XINIT2(J,I6)=0.
    IF(I6.NE.7) THEN
      I6=I6+1
```

```
      GO TO 43
    ENDIF
  ENDIF
 70 CONTINUE
* Spreading with summation of multiusers
 40 DO 60 K=1,NUSER
  DO 34 IZ=1,NSTAG
    CON1(IZ)=FEED1(K,IZ)
    SINIT1(IZ)=XINIT1(K,IZ)
    CON2(IZ)=FEED2(K,IZ)
    SINIT2(IZ)=XINIT2(K,IZ)
 34   CONTINUE
  CALL PNCOD1(K,nDS1)
  CALL PNCOD2(K,nDS2)
  nDS=(mod((nDS1+1),2)*nDS2)+(nDS1*mod((nDS2+1),2))
if(nDS.eq.1) then
  DS=1
else
  DS=-1
endif
  SUMSIG(JX)=SUMSIG(JX)+DS*XDAT(K)
 60 CONTINUE
IF (JX.NE.0) THEN
  I=I+1
  GO TO 116
ENDIF
* Calculating no. of data bits for intended user
 NBIT=NBIT+1
* Despread with 127 chip summation for the intended user
  DO 68 IY=1,NSTAG
    CON1(IY)=FEED1(INT,IY)
```

```
      SINIT1(IY)=XN1(INT,IY)
      CON2(IY)=FEED2(INT,IY)
      SINIT2(IY)=XN2(INT,IY)
   68   CONTINUE
    SUMBIT=0.
    DO 140 J9=1,NCHIP
  l=mod(J9,NCHIP)
      CALL PNCOD1(INT,nDS1)
      CALL PNCOD2(INT,nDS2)
    nDS=(mod((nDS1+1),2)*nDS2)+(nDS1*mod((nDS2+1),2))
  if(nDS.eq.1) then
    DS=1
  else
    DS=-1
  endif
      SUMBIT=SUMBIT+SUMSIG(l)*DS
   140   CONTINUE
* Phase of shift register after 127 chips(used for demodulation)
    DO 484 JR=1,NSTAG
      XN1(INT,JR)=XINIT1(INT,JR)
      XN2(INT,JR)=XINIT2(INT,JR)
   484   CONTINUE
* Quantization of recovered data bits
    IF (SUMBIT.GT.0.) THEN
      DET=1.
    ELSE
      DET=-1.
    ENDIF
* Calculation of prob. of error
    IF (DET.NE.XDAT(INT)) THEN
      ERR=ERR+1
```

```
    NBE=NBE+1
  ENDIF
* Could be used to determine #of consecutive errors
IF(DET.EQ.XDAT(INT)) THEN
  NBE=0
ENDIF
* Determining error free gaps
IF(DET.eq.XDAT(INT)) THEN
  KK=KK+1
  IF(NBIT.EQ.NBT) EFG(KK)=EFG(KK)+1
ELSE
  IF(KK.EQ.0) GO TO 3
  EFG(KK)=EFG(KK)+1
  KK=0
ENDIF
 3 vase(NBIT)=KK
  I=I+1
IF (I.LE.NSIM) GO TO 116
PROBER=ERR/NBT
write(117,342) PROBER
 342 format(1x,'*Prob.of error=',f10.8)
max1=vase(1)
do 211 NBITT=2,NBT
  if (vase(NBITT).gt.max1) max1=vase(NBITT)
 211 continue
DO 4 KK=1,max1
  write(117,*) ('ERROR FREE GAPS:')
  write(117,343) KK,EFG(KK)
 343   format(1x,'EFG(',i4,')=',i5)
 4 CONTINUE
 123 CONTINUE
```

```
      END
*----
      subroutine PNCOD1(K,nDS1)
      INTEGER K,nDS1,INT,NSTAG
      real SINIT1(20),XN1(200,20),XINIT1(200,20)
      real CON1(20),FEED1(200,20)
      common INT,NSTAG
      common SINIT1(20),XINIT1(200,20),CON1(20),FEED1(200,20)
      common XN1(200,20)
      common SINIT2(20),XINIT2(200,20),CON2(20),FEED2(200,20)
      common XN2(200,20)
      sum=0.
      do 29 i=1,NSTAG
        sum=sum+CON1(i)*SINIT1(i)
 29   continue
      xsum=amod(sum,2.)
      if (SINIT1(1).EQ.1.) then
        nDS1=1
      else
        nDS1=0
      endif
      do 39 j=2,NSTAG
        SINIT1(j-1)=SINIT1(j)
 39   continue
      SINIT1(NSTAG)=xsum
* Storing the phase of shift register for calculating PNCOD chips after first
*  chip
      do 423 IL=1,NSTAG
        XINIT1(K,IL)=SINIT1(IL)
 423  continue
      return
```

```
end
*----
subroutine PNCOD2(K,nDS2)
INTEGER K,nDS2,INT,NSTAG
real SINIT2(20),XN2(200,20),XINIT2(200,20)
real CON2(20),FEED2(200,20)
common INT,NSTAG
common SINIT1(20),XINIT1(200,20),CON1(20),FEED1(200,20)
common XN1(200,20)
common SINIT2(20),XINIT2(200,20),CON2(20),FEED2(200,20)
common XN2(200,20)
sum=0.
do 30 i=1,NSTAG
  sum=sum+CON2(i)*SINIT2(i)
 30 continue
xsum=amod(sum,2.)
if (SINIT2(1).EQ.1.) then
  nDS2=1
else
  nDS2=0
endif
do 40 j=2,NSTAG
  SINIT2(j-1)=SINIT2(j)
 40 continue
SINIT2(NSTAG)=xsum
* Storing the phase of shift register for calculating PNCOD chips after first
*  chip
do 424 IL=1,NSTAG
  XINIT2(K,IL)=SINIT2(IL)
 424 continue
return
```

```
end
*----
subroutine UNIFORM(ISEED,RANDX)
integer ISEED
real RANDX
ISEED=2045*ISEED+1
ISEED=ISEED-(ISEED/1048576)*1048576
RANDX=real(ISEED+1)/1048577.0
return
end
```

A.3 Source code for asynchronous short codes in land mobile satellite channels with Rayleigh flat fading and power control error

```
* Program 'pe4rafmsnw4.f'(Asynchronous Short Conc. Codes + noise
* with 4 Walsh groups, Shadow, Power control error
* with Rayleigh flat fading from mobile to satellite
* with Databit Synchronous and Pseudonoise code Asynchronous
REAL SUMSIG(0:126),XDAT(200),k1(1),SINIT(20),g(2),Ray(200)
real PL(200),pe(200)
integer ind(200),t,I
integer H(128,128,128),H1,HP
INTEGER DS,JX,IC(200,0:1),EFG(10000),vase(10000),ind1(200)
COMMON SINIT(20),XINIT(200,20),NSTAG,CON(20),FEED(200,20,9)
COMMON INT,XN1(200,20)
NSTAG=7
NCODE=9
NCHIP=127
NBT=10000
```

```
NSIM=NCHIP*NBT
INT=1
        CALL WAL(H)
        open (unit=57,file='npe4rfmsw-4',status='new')
write(57,*) 'RESULTS of Short Conc. Codes with AWGN, Walsh(4) '
        write(57,*) ' Shadow, Power cont. error '
write(57,*) ' with Rayleigh flat fading from Mob.to Sat.'
        write(57,*) ' with Databit Synch. and Pseudonoise code Asynch.'
* Calculating Prob.of error for different no.of users
        DO 123 NUSER=4,18,2
 write(57,*)
  write(57,341) NUSER
 341   format(1x,'No.of users=',i3)
        nMG=4
  indivg=NUSER/nMG
write(57,8) nMG
 8 format(1x,'$$$ (',i2,') groups')
write(57,10) indivg
 10 format(1x,'No.of people in one group=',i2)
        ISEED=156
n=0
  t=0
m=0
N1=1
k1(1)=0.
DO 232 I1=1,10000
  EFG(I1)=0
 232 CONTINUE
KK=0
ERR=0.
NBIT=0
```

```
itotal=0
nf=0
* Assigning different phase for each user
DO 111 I4=1,NUSER
  IF (I4.GT.1) THEN
    DO 222 I6=1,NSTAG
      XN1(I4,I6)=XN1(I4-1,I6)
      XINIT(I4,I6)=XINIT(I4-1,I6)
 222     CONTINUE
  ENDIF
  I6=1
 2   XN1(I4,I6)=XN1(I4,I6)+1
  XINIT(I4,I6)=XINIT(I4,I6)+1
  IF (MOD(XN1(I4,I6),2.).EQ.0) THEN
    XN1(I4,I6)=0.
    XINIT(I4,I6)=0.
    I6=I6+1
    GO TO 2
  ENDIF
  call UNIFORM(ISEED,yy)
  ind1(I4)=yy*127
  ind(I4)=mod(ind1(I4),NCHIP)
* Calculating a path loss (due to shadowing,sigma=5db) for each user
call agauss(3.1622777,g)
        PL(I4)=(10**(-g(1)/10))
* Calculating power control error (sigma=0.5db) for all users
call agauss(1.1220185,g)
        pe(I4)=(10**(-g(1)/10))
 111 CONTINUE
ind(INT)=0
* Installing feedback connections of the polynomial
```

```
DO 216 I8=1,NUSER
  DO 218 I9=1,NSTAG
    DO 219 I10=1,NCODE
      FEED(I8,I9,I10)=0.
              IF(I9.EQ.1) FEED(I8,I9,I10)=1.
      IF((I9.EQ.2).AND.(I10.EQ.4)) FEED(I8,I9,I10)=1.
              IF((I9.EQ.2).AND.(I10.EQ.6)) FEED(I8,I9,I10)=1.
      IF((I9.EQ.2).AND.((I10.EQ.8).OR.(I10.EQ.9))) FFED(I8,I9,I10)=1.
      IF((I9.EQ.3).AND.(I10.EQ.4)) FEED(I8,I9,I10)=1.
      IF((I9.EQ.3).AND.((I10.EQ.5).OR.(I10.EQ.9))) FEED(I8,I9,I10)=1.
      IF((I9.EQ.4).AND.((I10.EQ.3).OR.(I10.EQ.4))) FEED(I8,I9,I10)=1.
      IF((I9.EQ.4).AND.((I10.EQ.5).OR.(I10.EQ.6))) FEED(I8,I9,I10)=1.
      IF((I9.EQ.5).AND.((I10.EQ.1).OR.(I10.EQ.2).OR.(I10.EQ.3))) THEN
                FEED(I8,I9,I10)=1.
      ENDIF
      IF((I9.EQ.5).AND.((I10.EQ.5).OR.(I10.EQ.8))) FEED(I8,I9,I10)=1.
      IF((I9.EQ.6).AND.((I10.EQ.2).OR.(I10.EQ.3).OR.(I10.EQ.4))) THEN
                FEED(I8,I9,I10)=1.
      ENDIF
              IF((I9.EQ.6).AND.(I10.EQ.5)) FEED(I8,I9,I10)=1.
              IF((I9.EQ.6).AND.((I10.EQ.6).OR.(I10.EQ.9))) THEN
                FEED(I8,I9,I10)=1.
      ENDIF
      IF((I9.EQ.7).AND.((I10.EQ.2).OR.(I10.EQ.4).OR.(I10.EQ.5))) THEN
                FEED(I8,I9,I10)=1.
      ENDIF
      IF((I9.EQ.7).AND.((I10.EQ.7).OR.(I10.EQ.8))) FEED(I8,I9,I10)=1.
 219          CONTINUE
 218   CONTINUE
 216 CONTINUE
*
```

```
J=1
 116 JX=MOD(I,NCHIP)
SUMSIG(JX)=0.
* Generation of a random data bit for NUSERs every 127 chips
IF(JX.NE.1) GO TO 40
DO 70 J=1,NUSER
    CALL UNIFORM (ISEED,XX)
  IF (XX.LE.0.5) THEN
    XDAT(J)=-1.
  ELSE
    XDAT(J)=1.
  ENDIF
if(ind(INT).eq.0.and.ind(J).gt.0.and.N1.ge.2) n=1
  IF(XX.LE.0.11) IC(J,n)=1
  IF(XX.GT..11.AND.XX.LE.0.22) IC(J,n)=2
  IF(XX.GT..22.AND.XX.LE.0.33) IC(J,n)=3
  IF(XX.GT..33.AND.XX.LE.0.44) IC(J,n)=4
  IF(XX.GT..44.AND.XX.LE.0.55) IC(J,n)=5
  IF(XX.GT..55.AND.XX.LE.0.66) IC(J,n)=6
  IF(XX.GT..66.AND.XX.LE.0.77) IC(J,n)=7
  IF(XX.GT..77.AND.XX.LE.0.88) IC(J,n)=8
  IF(XX.GT..88.AND.XX.LE.1.0) IC(J,n)=9
n=t
 70 CONTINUE
* Spreading with summation of multiusers
 40 DO 60 K=1,NUSER
*  CALCULATING ray.flat fad.for each user every 20 databits
if (mod(I,20*NCHIP).eq.1) then
  call agauss(0.05,g)
  Ray(K)=sqrt(g(1)*g(1)+g(2)*g(2))
endif
```

```
  if(JX.le.ind(K).and.n.eq.0.and.N1.eq.1) then
    DS=0
    g(1)=0.
    go to 61
  endif
  if(JX.le.ind(K).and.n.eq.1) m=0
          if(JX.gt.ind(K).and.n.eq.1) m=1
  if(JX.gt.ind(K).and.ind(K).gt.ind(INT).and.N1.ge.2) then
    IC(K,0)=IC(K,1)
  endif
         ii=K/indivg+1
         if(mod(K,indivg).eq.0) ii=K/indivg
JC=mod((I-ind(K)),NCHIP)
if(JC.eq.0) JC=127
 H1=H(128,ii,JC)
          DO 34 IZ=1,NSTAG
    CON(IZ)=FEED(K,IZ,IC(K,m))
    SINIT(IZ)=XINIT(K,IZ)
 34   CONTINUE
  CALL PNCOD(K,DS)
* Signal-to-noise ratio (13 dB)
 call agauss(0.05,g)
 61   SUMSIG(JX)=SUMSIG(JX)+DS*XDAT(K)*Ray(K)*((PL(K)*pe(K))**0.5)*H1+g(1)
 60 CONTINUE
if(JX.eq.0) then
  n=1
t=1
  k1(1)=XDAT(INT)
endif
IF (I.lt.(N1*127+ind(INT))) THEN
  I=I+1
```

```
  if(I.eq.N1*127) n=1
  GO TO 116
ENDIF
* Calculating no. of data bits (for intended user)
n=0
t=0
m=0
 N1=N1+1
 NBIT=NBIT+1
* Despread with 127 chip summation (for intended user)
  DO 68 IY=1,NSTAG
    CON(IY)=FEED(INT,IY,IC(INT,n))
    SINIT(IY)=XN1(INT,IY)
 68   CONTINUE
  SUMBIT=0.
i1=INT/indivg+1
if(mod(INT,indivg).eq.0) i1=INT/indivg
 M1=0
  DO 140 J9=ind(INT)+1,ind(INT)+127
  M1=M1+1
    l=mod(J9,NCHIP)
          HP=H(128,i1,M1)
    CALL PNCOD(INT,DS)
    SUMBIT=SUMBIT+SUMSIG(l)*DS*HP
 140   CONTINUE
* Phase of shift register after 127 chips (used for demodulation)
  DO 484 JR=1,NSTAG
    XN1(INT,JR)=XINIT(INT,JR)
 484   CONTINUE
* Quantization of recovered data bits
  IF (SUMBIT.GT.0.) THEN
```

```
    DET=1.
  ELSE
    DET=-1.
  ENDIF
* Calculation of prob. of error
  IF (DET.NE.k1(1)) THEN
    ERR=ERR+1
    NBE=NBE+1
  ENDIF
* Could be used to determine #of consecutive errors
IF(DET.EQ.k1(1)) THEN
  NBE=0
ENDIF
* Determining error free gaps
IF(DET.eq.XDAT(INT)) THEN
  KK=KK+1
  IF(NBIT.EQ.NBT) EFG(KK)=EFG(KK)+1
ELSE
  IF(KK.EQ.0) GO TO 3
  EFG(KK)=EFG(KK)+1
  KK=0
ENDIF
 3 vase(NBIT)=KK
  I=I+1
         do 33 k=1,NUSER
  if(ind(k).le.ind(INT)) IC(k,0)=IC(k,1)
 33 continue
IF (I.LE.NSIM+ind(INT)) GO TO 116
PROBER=ERR/NBT
write(57,342) nMG,PROBER
 342 format(1x,'*Prob.of error for(',i2,')groups of users=',f10.8)
```

```
max1=vase(1)
do 211 NBITT=2,NBT
  if (vase(NBITT).gt.max1) max1=vase(NBITT)
 211 continue
do 7 ie=1,max1
  itotal=itotal+EFG(ie)
 7 continue
nEFG=ERR-itotal
q=alog(1-PROBER)
write(57,18) q
 18 format(1x,'Ln q =',f7.4)
  write(57,*) ('ERROR FREE GAPS:')
write(57,9) nEFG
 9 format(1x,'EFG(0)=',i5)
DO 4 KK=1,max1
 if(EFG(KK).eq.0) then
    nf=nf+1
 else
    if(nf.ne.0) then
              write(57,11) nf
 11        format(1x,'no. of zeros=',i4)
            endif
    write(57,343) KK,EFG(KK)
 343      format(1x,'EFG(',i5,')=',i5)
    nf=0
 endif
 4 CONTINUE
 123  CONTINUE
write(57,*) '=-=-=-=-=-=-=STOP=-=-=-=-=STOP=-==--==-='
END
*----
```

```
      subroutine PNCOD(K,DS)
integer K,DS,INT
real SINIT(20),XINIT(200,20)
common SINIT(20),XINIT(200,20),NSTAG,CON(20),FEED(200,20,9)
common INT,XN1(200,20)
sum=0.
do 30 i=1,NSTAG
  sum=sum+CON(i)*SINIT(i)
 30 continue
xsum=amod(sum,2.)
if (SINIT(1).EQ.1.) then
  DS=1
else
  DS=-1
endif
do 40 j=2,NSTAG
  SINIT(j-1)=SINIT(j)
 40 continue
SINIT(NSTAG)=xsum
* Storing the phase of shift register for calculating PNCOD chips after first
*  chip
do 424 IL=1,NSTAG
  XINIT(K,IL)=SINIT(IL)
 424 continue
return
end
*----
subroutine UNIFORM(ISEED,RANDX)
integer ISEED
real RANDX
ISEED=2045*ISEED+1
```

```
ISEED=ISEED-(ISEED/1048576)*1048576
RANDX=real(ISEED+1)/1048577.0
return
end
*---
subroutine agauss(a,g)
real a,sigma,g(2)
integer mean
sigma=sqrt(a)
mean=0
do 35 n=1,2
 36 vi1=0.
vi2=0.
call UNIFORM(ISEED,pm)
vi1=2*pm-1
call UNIFORM(ISEED,pp)
vi2=2*pp-1
r=vi1*vi1+vi2*vi2
if(r.ge.1) go to 36
g(n)=(vi1*sqrt(-2*alog(r)/r))*sigma+mean
 35  continue
return
end
*----
subroutine WAL(H)
integer H(128,128,128),ord,ir,jc,in
H(1,1,1)=1
ord=128
in=1
 13 do 14 ir=1,in
     do 15 jc=1,in
```

```
    H(2*in,ir,jc)=H(in,ir,jc)
    H(2*in,ir,jc+in)=H(2*in,ir,jc)
            H(2*in,ir+in,jc)=H(2*in,ir,jc)
    H(2*in,ir+in,jc+in)=-H(2*in,ir,jc)
 15   continue
 14 continue
if(in.lt.ord/2) then
  in=2*in
  go to 13
endif
return
end
```

A.4 Source code for long codes in land mobile satellite channels with Rician frequency selective fading and perfect power control

```
* Program 'po1rismsnw4.f'(Long Code + noise
* with 4 Walsh groups, Rician Selective frequency fading, and Shadow
* from mobile to satellite)
REAL SUMSIG(0:126),XDAT(200),SINIT(20),g(2),Ric1(200),Ric2(200)
real phi1(200),phi2(200),sig(200,127),sigt,A,PL(200)
INTEGER DS,JX,EFG(10000),vase(10000),itaw(200)
integer H(128,128,128),H1,HP
COMMON SINIT(20),XINIT(200,20),NSTAG,CON(20),FEED(200,20)
COMMON INT,XN1(200,20)
pi=3.1415927
NSTAG=15
NCHIP=127
NBT=10000
```

```
NSIM=NCHIP*NBT
INT=1
CALL WAL(H)
        open(unit=94,file='npolismsw-4',status='new')
write(94,*) 'RESULTS of LONG CODE with AWGN,Walsh(4), Shadow program'
write(94,*) 'with Rician Sel.Freq.fad. from Mob. to Sat.'
* Calculating Prob.of error for different no.of users
DO 123 NUSER=4,18,2
 write(94,*)
  write(94,341) NUSER
 341   format(1x,'No.of users=',i3)
nMG=4
indivg=NUSER/nMG
        write(94,8) nMG
 8 format(1x,'$$$ (',i2,') groups')
        write(94,10) indivg
 10 format(1x,'No.of people in one group=',i2)
A=1.
        ISEED=156
DO 232 I1=1,10000
  EFG(I1)=0
 232 CONTINUE
KK=0
        ERR=0.
NBIT=0
* Assigning different phase for each user
DO 111 I4=1,NUSER
  IF (I4.GT.1) THEN
    DO 222 I6=1,NSTAG
      XN1(I4,I6)=XN1(I4-1,I6)
      XINIT(I4,I6)=XINIT(I4-1,I6)
```

```
 222     CONTINUE
  ENDIF
  I6=1
 2   XN1(I4,I6)=XN1(I4,I6)+1
  XINIT(I4,I6)=XINIT(I4,I6)+1
  IF (MOD(XN1(I4,I6),2.).EQ.0) THEN
    XN1(I4,I6)=0.
    XINIT(I4,I6)=0.
    I6=I6+1
    GO TO 2
  ENDIF
* CALCULATING a delay time for each user
call UNIFORM(ISEED,zz)
itaw(I4)=zz*127
if(itaw(I4).eq.0) itaw(I4)=1
* Calculating a path loss (due to shadowing,sigma=5db) for each user
call agauss(3.1622777,g)
        PL(I4)=(10**(-g(1)/10))
 111 CONTINUE
* Installing feedback connections of the polynomial
DO 216 I8=1,NUSER
  DO 218 I9=1,NSTAG
    FEED(I8,I9)=0
    IF (I9.EQ.1) FEED(I8,I9)=1
    IF (I9.EQ.15) FEED(I8,I9)=1
 218   CONTINUE
 216 CONTINUE
*
I=1
 116 JX=MOD(I,NCHIP)
SUMSIG(JX)=0.
```

```
* Generation of a random data bit for NUSERs every 127 chips
IF(JX.NE.1) GO TO 40
DO 70 J=1,NUSER
    CALL UNIFORM (ISEED,XX)
  IF (XX.LE.0.5) THEN
    XDAT(J)=-1.
  ELSE
    XDAT(J)=1.
  ENDIF
 70 CONTINUE
 40 if(JX.ne.0) then
  JC=JX
else
  JC=127
endif
* Spreading with summation of multiusers
  DO 60 K=1,NUSER
 if(mod(K,indivg).eq.1) ii=K/indivg+1
         if(mod(K,indivg).eq.0) ii=K/indivg
 H1=H(128,ii,JC)
* CALCULATING rice sel.fad.for each user every 20 databits
sigt=0.
if(mod(I,20*NCHIP).eq.1) then
 call agauss(0.05,g)
 Ric1(K)=sqrt((A+g(1))*(A+g(1))+g(2)*g(2))
 if(K.ne.INT) then
theta=0.
  call UNIFORM(ISEED,theta)
  phi1(K)=2*pi*theta
 else
  phi1(INT)=0.
```

```
 endif
 call agauss(0.05,g)
 Ric2(K)=sqrt((A+g(1))*(A+g(1))+g(2)*g(2))
theta=0.
 call UNIFORM(ISEED,theta)
 phi2(K)=2*pi*theta
endif
  DO 34 IZ=1,NSTAG
    CON(IZ)=FEED(K,IZ)
    SINIT(IZ)=XINIT(K,IZ)
 34   CONTINUE
  CALL PNCOD(K,DS)
* Signal-to-noise ratio (13 dB)
 call agauss(0.05,g)
        pl=(PL(K)**0.5)
 SUMSIG(JX)=SUMSIG(JX)+DS*XDAT(K)*Ric1(K)*cos(phi1(K))*pl*H1+g(1)
do 36 IR=1,itaw(K)
 sig(K,IR-1)=sig(K,IR)
 36 continue
sig(K,itaw(K))=XDAT(K)*DS*pl*H1
sigt=sig(K,0)*Ric2(K)*cos(phi2(K))
SUMSIG(JX)=SUMSIG(JX)+sigt
 60 CONTINUE
IF (JX.NE.0.) THEN
  I=I+1
  GO TO 116
ENDIF
* Calculating no.of data bits (for intended user)
       NBIT=NBIT+1
* Despread with 127 chip summation (for intended user)
  DO 68 IY=1,NSTAG
```

```
    CON(IY)=FEED(INT,IY)
    SINIT(IY)=XN1(INT,IY)
 68   CONTINUE
  SUMBIT=0.
i1=INT/indivg+1
if(mod(INT,indivg).eq.0) i1=INT/indivg
  DO 140 J9=1,NCHIP
          l=mod(J9,NCHIP)
HP=H(128,i1,J9)
    CALL PNCOD(INT,DS)
    SUMBIT=SUMBIT+SUMSIG(1)*DS*HP
 140   CONTINUE
* Phase of shift register after 127 chips (used for demodulation)
  DO 484 JR=1,NSTAG
    XN1(INT,JR)=XINIT(INT,JR)
 484   CONTINUE
* Quantization of recovered data bits
  IF (SUMBIT.GT.0.) THEN
    DET=1.
  ELSE
    DET=-1.
  ENDIF
* Calculation of prob. of error
  IF (DET.NE.XDAT(INT)) THEN
    ERR=ERR+1
    NBE=NBE+1
  ENDIF
* Could be used to determine #of consecutive errors
       IF(DET.EQ.XDAT(INT)) THEN
  NBE=0
ENDIF
```

```
* Determining error free gaps
IF(DET.eq.XDAT(INT)) THEN
  KK=KK+1
  IF(NBIT.EQ.NBT) EFG(KK)=EFG(KK)+1
ELSE
  IF(KK.EQ.0) GO TO 3
  EFG(KK)=EFG(KK)+1
  KK=0
ENDIF
 3 vase(NBIT)=KK
  I=I+1
IF (I.LE.NSIM) GO TO 116
PROBER=ERR/NBT
write(94,342) nMG,PROBER
 342 format(1x,'*Prob.of error for(',i2,')groups of users=',f10.8)
max1=vase(1)
do 211 NBITT=2,NBT
  if (vase(NBITT).gt.max1) max1=vase(NBITT)
 211 continue
do 7 ie=1,max1
  itotal=itotal+EFG(ie)
 7 continue
nEFG=ERR-itotal
q=alog(1-PROBER)
write(94,18) q
 18 format(1x,'Ln q =',f7.4)
  write(94,*) ('ERROR FREE GAPS:')
write(94,9) nEFG
 9 format(1x,'EFG(0)=',i5)
DO 4 KK=1,max1
 if(EFG(KK).eq.0) then
```

```
    nf=nf+1
 else
    if(nf.ne.0) then
              write(94,11) nf
 11       format(1x,'no. of zeros=',i4)
            endif
    write(94,343) KK,EFG(KK)
 343     format(1x,'EFG(',i5,')=',i5)
    nf=0
 endif
 4 CONTINUE
 123 CONTINUE
write(94,*) '=-=-=-=-=-=-=STOP=-=-=-=-=STOP=-==--==-='
END
*----
subroutine PNCOD(K,DS)
integer DS,INT,K
real SINIT(20),XINIT(200,20)
common SINIT(20),XINIT(200,20),NSTAG,CON(20),FEED(200,20)
common INT,XN1(200,20)
sum=0.
do 30 i=1,NSTAG
  sum=sum+CON(i)*SINIT(i)
 30 continue
xsum=amod(sum,2.)
if (SINIT(1).EQ.1.) then
  DS=1
else
  DS=-1
endif
do 40 j=2,NSTAG
```

```
  SINIT(j-1)=SINIT(j)
 40 continue
SINIT(NSTAG)=xsum
* Storing the phase of shift register for calculating PNCOD chips after first
*  chip
do 424 IL=1,NSTAG
  XINIT(K,IL)=SINIT(IL)
 424 continue
return
end
*----
subroutine UNIFORM(ISEED,RANDX)
integer ISEED
real RANDX
ISEED=2045*ISEED+1
ISEED=ISEED-(ISEED/1048576)*1048576
RANDX=real(ISEED+1)/1048577.0
return
end
*---
subroutine agauss(a,g)
real a,sigma,g(2)
integer mean
sigma=sqrt(a)
mean=0
do 35 n=1,2
 36 vi1=0.
vi2=0.
call UNIFORM(ISEED,pm)
vi1=2*pm-1
call UNIFORM(ISEED,pp)
```

```
vi2=2*pp-1
r=vi1*vi1+vi2*vi2
if(r.ge.1) go to 36
g(n)=(vi1*sqrt(-2*alog(r)/r))*sigma+mean
 35  continue
return
end
*----
subroutine WAL(H)
integer H(128,128,128),ord,ir,jc,in
H(1,1,1)=1
ord=128
in=1
 13 do 14 ir=1,in
     do 15 jc=1,in
    H(2*in,ir,jc)=H(in,ir,jc)
    H(2*in,ir,jc+in)=H(2*in,ir,jc)
            H(2*in,ir+in,jc)=H(2*in,ir,jc)
    H(2*in,ir+in,jc+in)=-H(2*in,ir,jc)
 15   continue
 14 continue
if(in.lt.ord/2) then
  in=2*in
  go to 13
endif
return
end
```