Investigating the Capacity of a Cellular CDMA System

Yannis Avrithis

Supervised by Dr. A. Manikas

This report is submitted in partial fulfilment of the requirements for the degree of Master of Science (M.Sc.) and the Diploma of the Imperial College (D.I.C.)

Department of Electrical and Electronic Engineering
Imperial College of Science, Technology and Medicine
University of London

September 1994

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor, Dr. A. Manikas, who always supported my work and kept an open mind to my ideas throughout my project. Also, I would like to thank Mr. Y. L. Guan for his kind co-operation. His comments and suggestions have proved very useful for my work. Finally, a great thanks to all my fellow students and friends, and especially my family, whose moral support throughout the academic year was really invaluable.

ABSTRACT

Code Division Multiple Access (CDMA) is a multiple access scheme based on spread spectrum techniques, that has been used for many years for military communications, and quite recently for commercial applications, such as satellite and digital cellular radio communications. Certain inherent characteristics of spread spectrum, such as interference and multipath suppression capabilities, privacy, and more efficient spectrum reuse, make CDMA advantageous for mobile cellular communications. It is the intention of this project to provide an examination of how these properties of CDMA can be used to increase capacity comparing with conventional multiple access techniques. For this purpose, the capacity of a single-cell, power controlled, asynchronous direct-sequence (DS) CDMA system is first investigated using Gold codes and both binary and quadrature phase-shift-keying (BPSK and QPSK) modulation. The investigation carries on with the calculation of both the forward and reverse link capacity of a multiple-cell CDMA system by means of analytical calculations and Monte Carlo simulations. Finally, a straightforward comparison with conventional techniques such as FDMA and TDMA shows that CDMA can indeed provide much higher capacity.

1. INTRODUCTION

Spread spectrum techniques have been developed since the mid-50's and initially they were mainly used in military anti-jamming tactical communications and anti-multipath systems [1,2]. The inherent privacy feature as well as the interference and multipath attenuation characteristics of spread spectrum made it ideal in such hostile environments, but its use in non-military applications was limited since its bandwidth and energy efficiency was considered to be inferior comparing with other modulation techniques. In recent years, however, there has been an increasing interest in the use of spread spectrum for commercial applications, such as wireless local area networks (LAN's), personal communication networks (PCN) and digital cellular radio [3]. The main reason is that once the threat of designing a system that is capable of combating an intentional, smart jammer is removed, the receiver can be redesigned in order to make the system more efficient and more practical for commercial applications.

Indeed, while the use of spread spectrum means that a large amount of spectrum is used for each transmission, this can be compensated for by the interference reduction capability, so that a considerable number of users might use the same spectral band. This is the main idea behind *code division multiple access* (CDMA) which appears to be one of the most popular civilian applications of spread spectrum today. In contrast with conventional multiple access techniques such as frequency division multiple access (FDMA) and time division multiple access (TDMA), where different users are served by assigning different frequency bands or different time slots, with CDMA all users simultaneously use the entire spectrum devoted to the system. The users are distinguished by means of different *pseudo-random sequences*, or codes. In theory, it does not matter whether the spectrum is divided into frequencies, time slots or codes; the capacity provided from all these multiple access schemes is the same. However, depending on the characteristics of a certain application, we may find that one multiple access scheme is more appropriate than another. In a mobile cellular communications system, in particular, which is arguably one of the hottest topics in communications today, there are several advantages to using CDMA [4].

One of the main advantages of CDMA in a cellular environment is a major increase in *capacity*, which is the maximum number of users that the system can accommodate, while maintaining an acceptable level of performance. Although the CDMA capacity was until recently considered to be inferior to that of other techniques [5], it was later recognised that since the CDMA capacity is only interference limited, unlike FDMA and TDMA capacities that are mainly bandwidth limited, any reduction in interference leads directly to an increase in capacity [6]. This fact can be exploited in many ways [7, chap. 9]. For instance, one can take advantage of the

natural shape of human conversation. Since voice signals are intermittent with an activity cycle of approximately 35%, *voice activity detection* techniques can be employed and transmission can be suppressed during the quiet periods of each user so that other users face less interference. The capacity can thus be increased by an amount inversely proportional to the duty factor of voice signals. This technique can also be applied with FDMA or TDMA but in this case perfect synchronisation, more signalling information and complex channel assignment procedures are required, whereas in CDMA all users can operate independently and asynchronously. Furthermore, *multi-sectored antennas* can provide spatial isolation of users, thus reducing interference and proportionally increasing capacity. Again in FDMA and TDMA sectorisation cannot be exploited for capacity as the isolation provided by the antennas is not enough to permit reuse of a channel in two sectors of the same cell.

Moreover, since isolation among cells is only provided by path loss, which is typically proportional to the fourth power of the distance, conventional multiple access schemes have to reuse the same channel in only one of every seven cells in present systems, whereas CDMA reuses the entire spectrum for all cells. This is of course the most efficient frequency reuse scheme we can find. Naturally, this scheme introduces more interference, yet this is compensated for by the interference suppression capability of spread spectrum. Finally, CDMA requires no *guard bands* or *guard times*. In FDMA guard bands are required between contiguous frequency channels because of the non-ideal filter characteristics. Similarly, guard times are required between time slots in TDMA because of synchronisation errors, propagation delays and the *time delay spread* that is present in the mobile radio environment. From this point of view, CDMA is more efficient as all the users operate continuously, using the entire spectrum. Taking account of all the above factors, the capacity of a cellular CDMA system is calculated in [8], and it is found to be almost 18 times the capacity of analog FM/FDMA systems and 6 times the capacity of digital FDMA and TDMA systems.

There are many other advantages of cellular CDMA. For instance, since wideband waveforms are employed, signals encounter less severe *fading* as their reception takes advantage of the natural frequency diversity over the wideband channel [4]. Also, spread spectrum modulation allows different propagation paths to be separated when the difference in the path delays for the various paths exceeds a certain amount of time, which is typically in the order of 1 microsecond [9]. Therefore, *multipath*, which is often a fundamental limitation to system performance, can be suppressed with CDMA. Moreover, *equalisers*, that are usually needed in both FDMA and TDMA to reduce the intersymbol interference caused by time delay spread, are not needed in CDMA. Finally, hard hand-off from one frequency to another while a mobile user moves from cell to cell does not happen in CDMA. In such a case, only the code sequences change. This is called a *soft hand-off* [7,chap.9].

Code division multiple access is mainly accomplished by two different modulation techniques. The first is *direct sequence* (DS) modulation, where the signal that corresponds to each user is modulated by a pseudo-random sequence waveform. At the receiver, the sum of the signals that correspond to all active users is driven to a correlator, where the correlation with the desired user's sequence is calculated. Because sequences, or codes, are designed in such a way so that any two codes corresponding to different users are uncorrelated, the desired user's signal can be separated from the interference. In the second technique, *frequency hopping* (FH), many different channels are employed for each user, so that the corresponding signal hops over many different frequencies in some determined hopping pattern. Frequency hopping can be either fast, when there are two or more hops for each symbol, or slow, when there are two or more symbols for each hop. However, slow frequency hopping does not increase capacity, while the technology for fast frequency hopping at 800 MHz is not yet available [7,chap.9] (800MHz is the usual carrier frequency for terrestrial cellular communications, according to FCC regulations [9], as explained in Chapter 5). For this reason, this report concentrates only on DS modulation.

An important issue that usually has to be resolved in a CDMA system is the *"near-far"* problem. Because all users actually operate in the same spectral band and therefore interfere with each other, strong signals that are received from users that are near a cell site will tend to mask weak signals arriving from distant users. For this reason a *power control scheme* [4] is usually applied on the *reverse link* (from mobile users to the cell site). The received power from each mobile user is measured at the cell site receiver, which then commands a power adjustment to equalise all received signals within a cell. Similarly for the *forward link* (from the cell site to mobile users), the power received from neighbouring cell sites is measured at each mobile and the necessary power allocation is determined at the cell site transmitters according to the needs of individual users. Throughout this report, perfect power control is assumed to be applied in both directions (forward and reverse link).

The objective of this report is to provide a thorough analysis of a cellular CDMA system in order to understand the main parameters that affect the system performance and to calculate, in terms of these parameters, the capacity of such a system, thereby verifying the capacity advantage of CDMA over conventional techniques. In Chapter 2, the generation of pseudo-random sequences appropriate for CDMA is studied and Gold sequences are selected for the implementation of the experimental systems mainly because of the good cross-correlation properties they possess. In Chapter 3, a single-cell, asynchronous, power controlled DS-CDMA system is considered using BPSK and QPSK modulation, while perfect pseudo-random sequence and carrier synchronisation is assumed at the receiver (the code acquisition and tracking problem at the receiver is thoroughly studied in [10,chap.9-10]). The simulations concentrate on the

calculation of the bit error rate (BER) as a function of the E_b / I_o ratio at the receiver, where E_b is the energy per bit and I_o the interference (plus thermal noise) power spectral density. The required E_b / I_o ratio is calculated for the system to work with an acceptable BER performance, and based on this ratio, the performance of a multiple cell DS-CDMA system is examined in Chapter 4. By using analytical calculations and Monte-Carlo simulations, the appropriate interference terms are estimated for the forward as well as the reverse link. The capacity of both links is thus estimated and the results are shown to be more accurate than those in [8]. In particular, in accordance with [4], the total system capacity is shown to be limited by the forward and not the reverse link as in [8,11]. Finally, as shown by a straightforward comparison in Chapter 5, a considerable increase in capacity is achieved by CDMA comparing with FDMA and TDMA.

2. PSEUDO-RANDOM SEQUENCES

Code Division Multiple Access (CDMA) is accomplished by means of a set of different *pseudo-random* or *pseudo-noise (PN) sequences* or *codes* that are assigned to each user. The performance of a CDMA system depends to a great extent on the design of the PN sequences that are employed, therefore the properties of different PN sequences are examined in this chapter. After a brief introduction to the CDMA concept and a presentation of the desired properties of good PN sequences in the following section, *maximal length* or *m*-sequences are considered in section 2.2 while *Gold codes* are examined in section 2.3.

2.1 Desired Properties of PN Sequences

Code Division Multiple Access. In the *direct sequence (DS)* CDMA environment that we will consider we assume that K users are present in the system. The *k*-th user's data signal $d_k(t)$, which represents the user's binary information sequence, is a sequence of unit amplitude, positive and negative, rectangular pulses of duration T. Thus $d_k(t)$ can be expressed as

$$d_k(t) = \sum_{j=-\infty}^{\infty} d_j^{(k)} p_T(t - jT) \tag{1}$$

where $d_j^{(k)} \in \{+1, -1\}$, $p_\tau(t) = \text{rect}(t/\tau)$ and

$$\text{rect}(t) = \begin{cases} 1, & 0 \le t \le 1 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The *k*-th user is assigned a code sequence $\{a_j^{(k)}\}$. If $a_j^{(k)} \in \{+1, -1\}$ are the elements of $\{a_j^{(k)}\}$, the code waveform $a_k(t)$ is formed as

$$a_k(t) = \sum_{j=-\infty}^{\infty} a_j^{(k)} p_{T_c}(t - jT_c) \tag{3}$$

Then similarly $a_k(t)$ is a sequence of unit amplitude, positive and negative, rectangular pulses of duration $T_c << T$. At each user's transmitter, $d_k(t)$ is multiplied by $a_k(t)$ and the resultant signal is then modulated using a digital modulation technique (e.g. BPSK or QPSK). The received signal, which is actually the sum of all the transmitted signals plus noise, is the input to a correlation receiver matched to the transmitted signal of a desired (say *i*-th) user. Thus the desired signal is separated from the other user interference, demodulated and despread. This model can be used for the forward as well as the reverse link under certain assumptions, as will be described in greater detail in Chapter 4.

Processing Gain. Notice that by the above transmission procedure the data signal is in fact modulated by the much faster code waveform. Since the necessary bandwidth for the transmission of the data signal $d_k(t)$ is equal to $B = 1/T$ and the necessary bandwidth for transmission of $d_k(t) \cdot a_k(t)$ is $B_{ss} = 1/T_c$, the bandwidth is actually *spread* and the *processing gain* [1] is defined as

$$G_P = \frac{B_{ss}}{B} = \frac{T}{T_c} \tag{4}$$

As will be shown later, the greater the processing gain of a CDMA system, the better its performance. Usually the processing gain is in the range of 10 to 1000 (or 10dB to 30dB).

Desired Properties. If the code sequences are not totally uncorrelated, there is always an interference component at the output of the receiver which is proportional to the cross-correlation between different code sequences, as will be explained in chapter 3. Therefore it is desired that this cross-correlation is made as small as possible. In addition, in order for the CDMA system to be able to combat multipath, consecutive samples of the code sequences should be uncorrelated, i.e. code sequences should have impulse-like autocorrelation functions. Pure random sequences could be used for this purpose as code sequences, but since the receiver needs a replica of the desired code sequence in order to despread the signal, PN sequences are used instead in practice. PN sequences are deterministic, periodic sequences with period N. We assume that $N = T/T_c = G_P$ so that there is one code period per data symbol. This choice for N will be explained later. Therefore the PN sequences used in a CDMA system have to:

1. be easy to generate
2. have randomness properties
3. have low cross-correlation

Other properties are also desired for the PN sequences of a spread spectrum system designed to operate in a hostile environment. For example the sequences should have long periods and they should be difficult to reconstruct from a short segment [1]. These properties however are not required for a CDMA system in a non-hostile environment. The first of the above properties is easily achieved with the generation of PN sequences by means of *shift registers,* while the second, or randomness property, is achieved by appropriately selecting the feedback connections of the shift registers, as will be shown in the next two sections. Finally for the third property we make the following definitions.

Correlation Properties. The *periodic cross-correlation function* $\theta_{k,i}$ [12] for the code sequences $\{a_k\}$ and $\{a_i\}$ can be defined as

$$\theta_{k,i}(l)=\sum_{j=0}^{N-1}a_j^{(k)}a_{j+l}^{(i)} \qquad 0\le l<N \tag{5}$$

whereas the *discrete aperiodic cross-correlation function* $C_{k,i}$ is defined as

$$C_{k,i}(l)=\begin{cases}\sum_{j=0}^{N-1-l}a_j^{(k)}a_{j+l}^{(i)}, & 0\le l\le N-1\\ \sum_{j=0}^{N-1+l}a_{j-l}^{(k)}a_j^{(i)}, & 1-N\le l<0\\ 0, & |l|\ge N\end{cases} \tag{6}$$

It is easy to see that $\theta_{k,i}(l)=C_{k,i}(l)+C_{k,i}(l-N)$. Therefore we can similarly define the *odd cross-correlation function* $\hat{\theta}_{k,i}(l)=C_{k,i}(l)-C_{k,i}(l-N)$ [13]. The name of this function follows from the property $\hat{\theta}_{k,i}(l)=-\hat{\theta}_{k,i}(N-l)$. Notice that the periodic or *even cross-correlation function* has the property $\theta_{k,i}(l)=\theta_{k,i}(N-l)$. For a single code sequence *k*, the corresponding *autocorrelation functions* $\theta_{k,k}$, $\hat{\theta}_{k,k}$ and $C_{k,k}$ will be denoted by θ_k, $\hat{\theta}_k$ and C_k respectively.

For best system performance, all of the above cross-correlation functions should be as small as possible, since they are proportional to the interference from other users. The out-of-phase (i.e. for lag not equal to zero) autocorrelation functions should also be made as small as possible, since these affect the multipath suppression capabilities and the acquisition and tracking performance of the receivers [10,chap.9-10]. We thus define the peak cross-correlation parameters θ_c, $\hat{\theta}_c$ and C_c as the maxima of $|\theta_{k,i}(l)|$, $|\hat{\theta}_{k,i}(l)|$ and $|C_{k,i}(l)|$ respectively, where the maximum is taken over all l and all *i* and *k* such that *i*<*k*. Similarly we define the peak autocorrelation parameters θ_a, $\hat{\theta}_a$ and C_a as the maxima of $|\theta_k(l)|$, $|\hat{\theta}_k(l)|$ and $|C_k(l)|$ respectively, where the maximum is taken over all $l\neq 0 \pmod N$ and all *k*. Finally we define $\theta_{\max}=\max\{\theta_a,\theta_c\}$, $\hat{\theta}_{\max}=\max\{\hat{\theta}_a,\hat{\theta}_c\}$ and $C_{\max}=\max\{C_a,C_c\}$.

With the above definitions we can see that the smaller the peak correlation parameters $\theta_{\max}$, $\hat{\theta}_{\max}$ and $C_{\max}$, the better the performance of a system. These parameters, however, cannot be made as small as we wish. For example, according to the Welch lower bound [14,15],

$$\theta_{\max}\ge N\sqrt{(K-1)/(NK-1)} \tag{7}$$

and

$$C_{\max}\ge N\sqrt{(K-1)/(2NK-K-1)} \tag{8}$$

for a set of K sequences of period N. Therefore for large values of K and N the lower bounds on $\theta_{\max}$ and $C_{\max}$ are approximately $\sqrt{N}$ and $\sqrt{N/2}$ respectively. Moreover, based on equations (6) and (10) of [15] one can show that

$$\theta_a^2 + \theta_c^2 > N \tag{9}$$

and

$$C_a^2 + C_c^2 > N/2 \tag{10}$$

The above shows that not only is there a lower bound on the maximum correlation parameters, but also a *trade-off* between the peak autocorrelation and cross-correlation parameters. Thus the autocorrelation and cross-correlation functions cannot be both made small simultaneously (for a more strict bound see also equations (2.21) and (5.32) of [12]). The design of the code sequences should be therefore very careful so that all the of above quantities of interest remain as small as possible.

2.2 Maximal Length Sequences

Generation of *m*-sequences. Maximal length or *m*-sequences [10,chap.7] are very widely used in spread spectrum systems because of their very good autocorrelation properties. They are generated by means of *linear feedback shift registers* as shown in Figure 1.

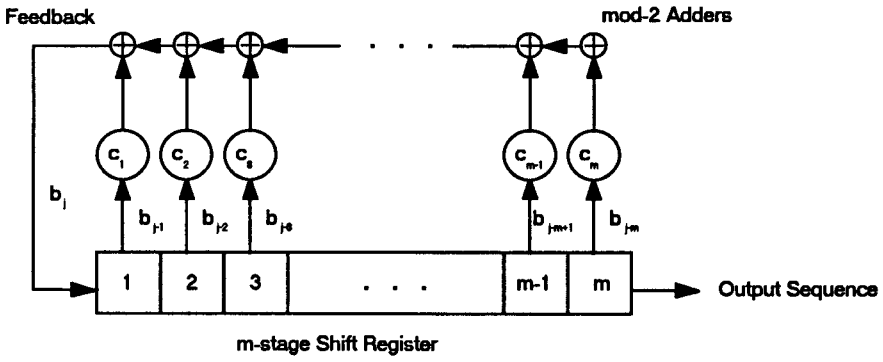

Figure 1. *Linear Feedback Shift Register with m stages.*

As it is obvious from the figure the output sequence $\{b_j\}$ is a periodic sequence that satisfies the following *linear recursion relation:*

$$b_j = c_1 b_{j-1} + c_2 b_{j-2} + \cdots + c_m b_{j-m} = \sum_{i=1}^{m} c_i b_{j-i} \tag{11}$$

where all elements of the sequence $\{b_j\}$ as well as the coefficients c_i, $i = 1,\ldots,m$ are assumed to be either 0 or 1 and the summation is modulo-2. For an m-stage shift register, we always have

$c_m = 1$. The initial contents of the shift register are called *initial conditions.* The output sequence is said to be an *m*-sequence when its period is equal to $N = 2^m - 1$ (i.e. the maximum possible period for the above shift register generator). The period *N* depends on the feedback connections, i.e. on the coefficients c_i and is maximum when the *characteristic polynomial*

$$f(x) = 1 + c_1 x + c_2 x^2 + \cdots + c_m x^m = \sum_{i=0}^{m} c_i x^i, \quad c_0 = 0 \tag{12}$$

is a *primitive polynomial* of degree *m* [10,chap.7]. Table 8-2 of [16] and Appendix C of [17] provide lists of primitive polynomials of degree up to 80 and were used for the simulation programs in order to generate *m*-sequences.

Properties of *m*-sequences. Maximal length sequences have the following very important "randomness" properties [1]:

1. There is an approximate *balance* of "zeros" and "ones" (2^{m-1} ones and $2^{m-1} - 1$ zeros in any period of length $2^m - 1$).
2. In any period, half of the runs of consecutive zeros or ones are of length one, one fourth are of length two, one eighth are of length three etc.
3. If we define the ± 1 sequence $a_j = 1 - 2b_j$ as the code sequence used in (3), the periodic (even) autocorrelation function of $\{a_j\}$ will be *two-valued:*

$$\theta(l) = \begin{cases} N, & l = 0 \,(\text{mod } N) \\ 1, & \text{otherwise} \end{cases} \tag{13}$$

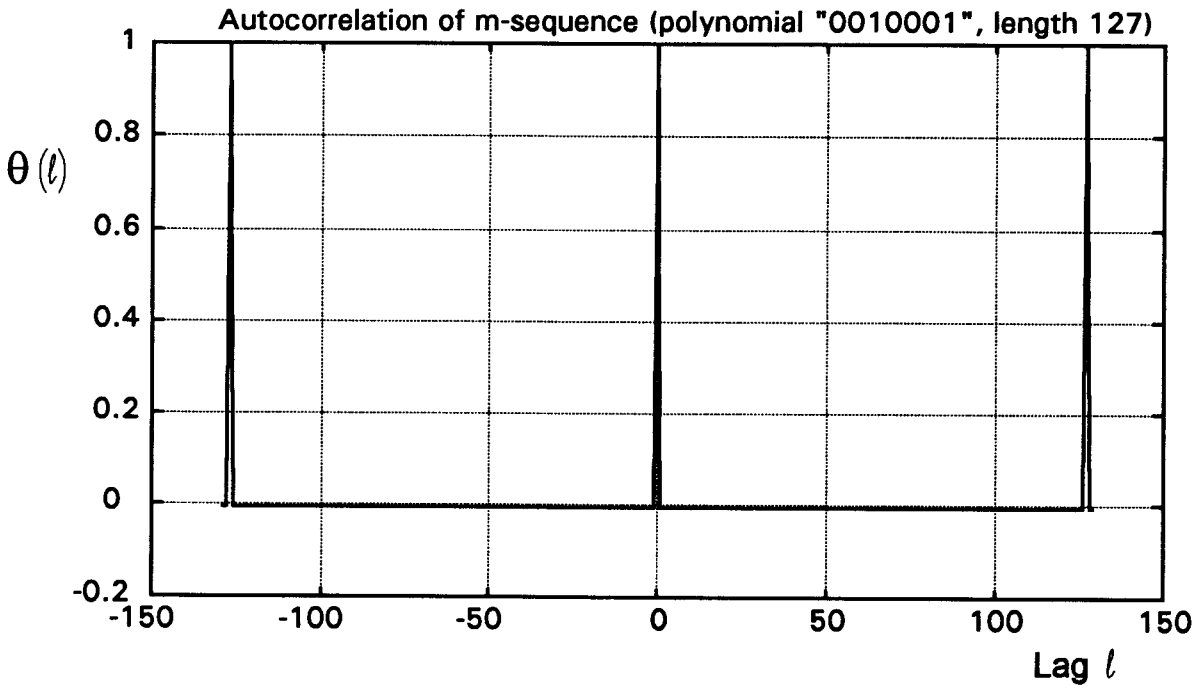

Figure 2. *Normalised autocorrelation function of m-sequence of length 127.*

An example of the autocorrelation function of an *m*-sequence of length $2^7 - 1 = 127$ is given in Figure 2, where the normalised autocorrelation function $(1/N)\theta(l)$ is plotted versus l. We normalise the autocorrelation function so that its maximum value (for lag $l = 0$) is unity. It is obvious that this function is impulse-like as it is desired, while the peak autocorrelation parameter θ_a is equal to 1 (1/127 for the normalised function). With such an autocorrelation function, it is clear that almost all delayed copies of the same signal that arrive at the receiver due to multipath are attenuated by a factor of 1/127. Thus multipath is suppressed, and this makes CDMA ideal for a mobile environment where multipath is one of major problems that limit the system performance. In Figure 2, the corresponding primitive polynomial of degree 7 is defined by a set of coefficients $\{c_1, c_2, \ldots, c_m\}$ which is given as a string of zeros and ones.

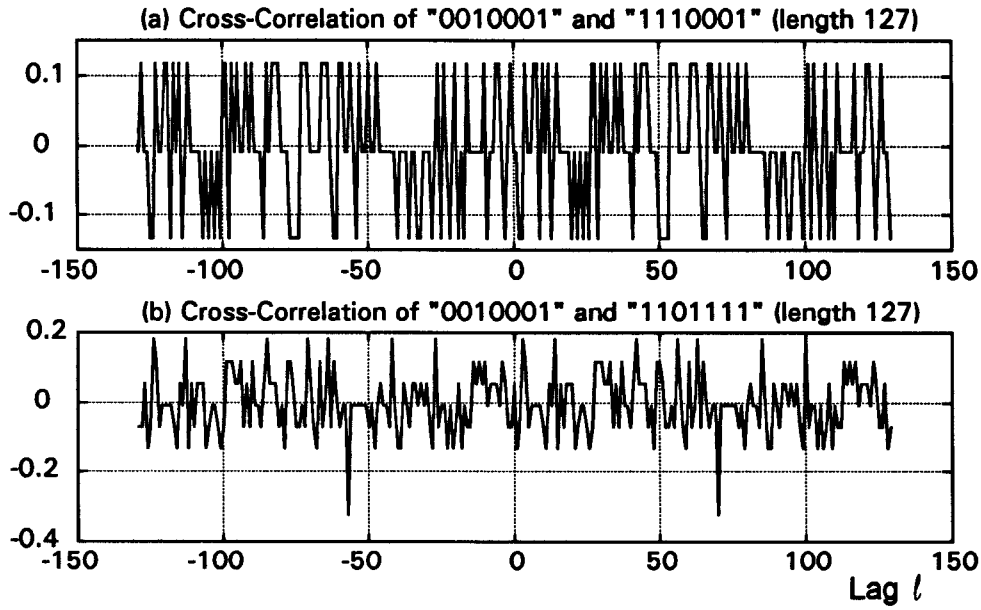

Figure 3. *Normalised cross-correlation function for two pairs of m-sequences of length 127.*

Cross-correlation function of *m*-sequences. From the above we can see that *m*-sequences are easily generated and possess excellent randomness properties. However, property 3 of the previous section is not satisfied, as the cross-correlation function can become very large for certain pairs of *m*-sequences (i.e. of primitive polynomials). This is illustrated in Figure 3, where the (normalised) even cross-correlation functions for two different pairs of *m*-sequences of length 127 are plotted versus l (the corresponding primitive polynomials of degree 7 are again defined by sets of coefficients $\{c_1, c_2, \ldots, c_m\}$ which are given as strings of zeros and ones).

The cross-correlation between *m*-sequences can actually be analytically evaluated as in [12], and thus the peak cross-correlation parameters can be calculated [10,chap.11]. In particular, there exist pairs of primitive polynomials with corresponding *m*-sequences $\{a_j^{(k)}\}$ and $\{a_j^{(i)}\}$ for which the (even) cross-correlation function $\theta_{k,i}(l)$ is *three-valued* as in Figure 3a and

$$|\theta_{k,i}(l)| \leq \theta_c = 2^{(m+1)/2} + 1 \quad \text{for } m \text{ odd}$$

$$|\theta_{k,i}(l)| \leq \theta_c = 2^{(m+2)/2} + 1 \quad \text{for } m \text{ even}, \quad m \neq 0 \pmod 4 \tag{14}$$

These pairs of polynomials are called *preferred pairs.* However, for most other pairs of polynomials that are not preferred pairs, the cross-correlation function, and therefore θ_c , is much higher as is the case in Figure 3b. Even if we find many pairs of preferred polynomials, it is still difficult to find a large set of *m*-sequences such that any two sequences of the set have low cross-correlation.

***M*-sequences for CDMA.** Because of the high cross-correlation between *m*-sequences, the interference between different users in a CDMA environment will be large, therefore *m*-sequences are not suitable for CDMA applications. Note however that in a completely synchronised CDMA system, different time offsets of the same basic *m*-sequence can be assigned to different users. This way the cross-correlation function between two code sequences will actually be equal to the autocorrelation function of the basic code which is very small. Unfortunately this scheme cannot operate in an asynchronous environment - which will be considered from now on - as in this case the time offsets cannot be determined.

2.3 Gold Sequences

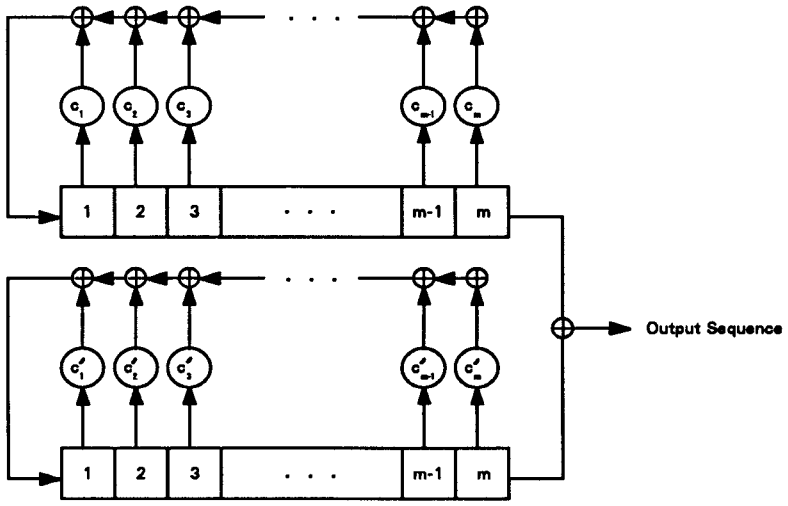

Figure 4. *Gold code generator.*

Generation of Gold sequences. As discussed in the previous section, although *m*-sequences possess excellent randomness (and especially autocorrelation) properties, they are not generally used for CDMA purposes as it is difficult to find a set of *m*-sequences with low cross-correlation for all possible pairs of sequences within the set. By slightly relaxing the conditions on the

autocorrelation function, however, we can obtain a family of code sequences with the high cross-correlation peaks eliminated. Such an encoding family can be achieved by *Gold sequences* or *Gold codes* [10,chap.11] which can also be generated by shift registers as shown in Figure 4.

The Gold code set is actually obtained by the modulo-2 sum of two *m*-sequences with different phase shifts for the first *m*-sequence relative to the second. Since there are $N = 2^m - 1$ different relative phase shifts, and since we can also have the two *m*-sequences alone, we can obtain a set with a total of $2^m + 1$ sequences of period $N = 2^m - 1$. These sequences, however, are *not* maximal length sequences.

Correlation properties of Gold codes. The sets of coefficients $\{c_1, c_2, \ldots, c_m\}$ and $\{c'_1, c'_2, \ldots, c'_m\}$ shown in Figure 4 cannot be arbitrary; they have to correspond to a preferred pair or primitive polynomials. It has been proved that if this is the case, then the (even) cross-correlation function of any two codes of the set is again *three-valued* and satisfies the same inequality (14) as the cross-correlation function of the two *m*-sequences of the preferred pair [18,19]. The same holds for the (out of phase) autocorrelation function of any code of the set. In fact, Gold codes form an *optimal* set of codes with respect to a certain lower bound on the peak correlation parameter θ_{max} [12]. As can be seen by direct comparison of (13) and (14), however, the (out of phase) autocorrelation function is considerably higher than that of an *m*-sequence. This agrees with result (9), since Gold codes have higher θ_a and lower θ_c than *m*-sequences, and the *trade-off* between these parameters is thus verified.

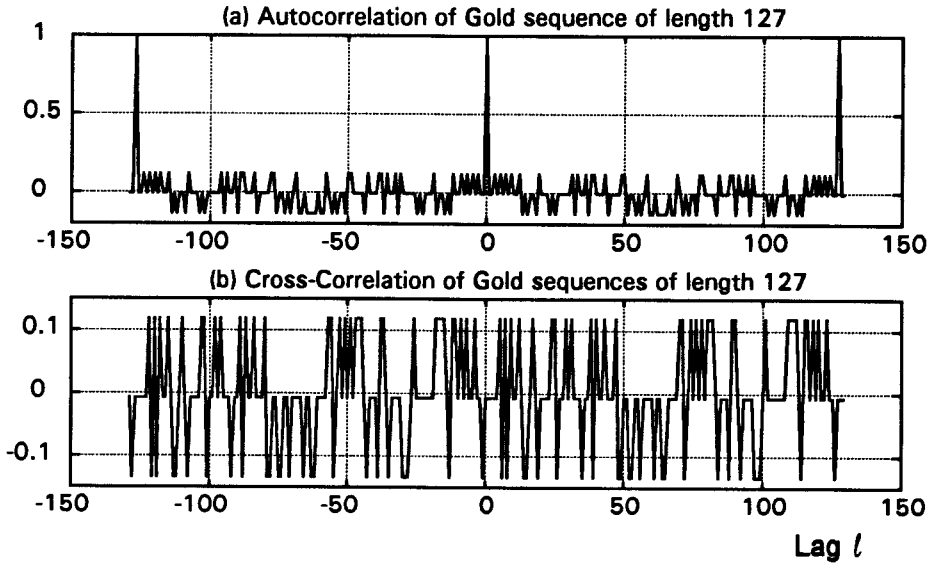

Figure 5. *Normalised autocorrelation and cross-correlation functions of Gold codes of length 127.*

As an example, a typical (normalised) autocorrelation function of a Gold sequence of length 127 as well as a cross-correlation function of two such sequences are plotted in Figure 5 versus the lag l. It can be seen from this figure that the autocorrelation function is impulse-like as well, but now the its out-of-phase components (i.e. for $l \neq 0 \pmod N$) are indeed notably higher than the corresponding components in the case of *m*-sequences. The cross-correlation function, however, is always bound and does not have peaks as in Figure 3b. Both the peak correlation parameters θ_a and θ_c are indeed equal to $2^{(7+1)/2}+1=17$ ($17/127$ for the normalised cross-correlation function) as predicted by (14).

Note that the *odd* cross-correlation for Gold does not satisfy inequality (14). An upper bound on $\hat{\theta}_c$, however, can be found in [12]:

$$\hat{\theta}_c \leq 2^{m-1}+2^{[m/2]}+1 \tag{15}$$

where [x] denotes the integer part of x. Although this bound is considerably higher than that of (14) for θ_c, on the average the odd cross-correlation function is not much higher than the even one. This is illustrated in Figure 6, where a typical odd cross-correlation function of two Gold codes of length 127 is plotted versus l. As will be shown in chapter 3, for CDMA applications the odd cross-correlation function is as important as the periodic (even) one. However, the odd cross-correlation function has receiver considerably less attention in the literature. Some relevant results can be found in [12].

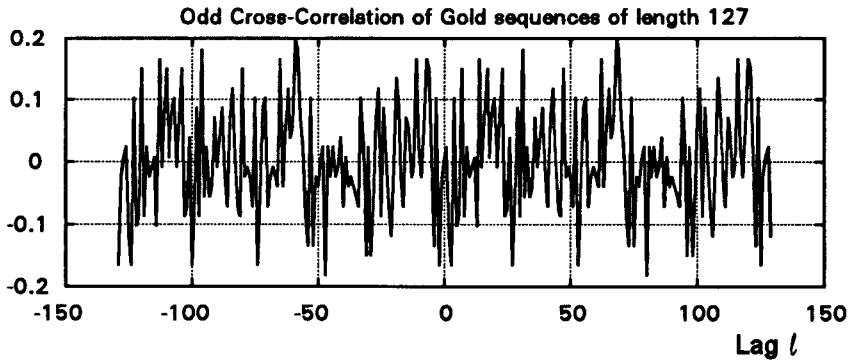

Figure 6. *Odd cross-correlation function of two Gold codes of length 127.*

Balanced Gold codes. We should also note that not all Gold codes generated by the above procedure are balanced, i.e. the number of "ones" in a in a code period does not always exceed the number of "zeros" by one as is the case for *m*-sequences. For example, as can be found in [10,chap.11], for *m* odd only $2^{m-1}+1$ code sequences of the total 2^m+1 are balanced, while the rest 2^{m-1} code sequences have an excess or a deficiency of ones. For $m=7$, for instance, only

65 balanced Gold codes can be produced, out of a total possible of 129. Of these, 63 are non-maximal and two are maximal length sequences. Because balanced codes have more desirable spectral characteristics, only those are used in the experimental CDMA system of the following chapter. Balanced Gold codes are generated by appropriately selecting the relative phases of the two original *m*-sequences, as shown in [10,chap.11].

Concluding this section, note that with the above described procedure, and by selecting any preferred pair of primitive polynomials it is easy to construct a very large set of ideal code sequences that minimise other-user interference. Instead of having each user employ a different *m*-sequence we assign to each user a member of the code set. This set has the property any two codes that belong the set have cross-correlation function that is upper bound as in inequality (14), thus the high cross-correlation peaks are eliminated. The sequences obtained in this way are nonmaximal, hence their autocorrelation function is not two-valued; however the out-of-phase autocorrelation function still satisfies inequality (14) as proved in [18]. Other encoding families (e.g. Kasami sequences) can also be constructed, as described in [12] and [20].

3. ANALYSIS OF A SINGLE-CELL DIRECT SEQUENCE CDMA SYSTEM

In order to gain an insight of how a CDMA system operates and to illustrate how the performance of such a system depends on the number of users present in the system, a *single-cell, asynchronous direct-sequence (DS) CDMA system* is analysed and simulated in this chapter. *Gold codes* are used as code sequences, because of their good cross-correlation properties that make them ideal for CDMA applications, as it was highlighted in the previous chapter. Two digital modulation techniques are used, namely *binary phase-shift-keying* (BPSK) and *quadrature-phase-shift-keying* (QPSK), which are developed in sections 3.1 and 3.2 respectively, while their performance comparison is carried out in section 3.3.

In all that follows in this chapter, K users are assumed present in the system, whose DS modulated and transmitted signals are received, together with thermal noise, at a single correlation receiver. The (coherent) receiver is assumed matched to the transmitted signal of a single, desired user, while all the other signals constitute the total interference. This can serve as a model of both the *forward* and the *reverse link* of a cellular CDMA system under certain assumptions, as will be described in greater detail in the next chapter. The objective of this chapter is to relate the performance of the system, which is measured by the *bit-error-rate* (BER) with the total number K of users as well as with the E_b / I_0 ratio at the receiver, where E_b is the energy per bit and I_0 the (one-sided) interference plus thermal noise power spectral density. The E_b / I_0 ratio is also called the *equivalent energy utilisation efficiency* (EUE_{equ}) in a CDMA environment, in analogy with the *energy utilisation efficiency* ($EUE = E_b / N_0$, where N_0 is the (one-sided) thermal noise power spectral density) that is often used in ordinary digital communication systems. The maximum number of users, or equivalently the minimum required E_b / I_0 ratio is calculated for the system to work with an acceptable BER performance, and an initial estimate for the system capacity can be obtained in this way. Based on this analysis, the performance of a multiple-cell DS-CDMA system is examined in Chapter 4, where other factors of a cellular environment are also considered, such as fading and other-cell interference.

The systems that are considered in this chapter are assumed perfectly *power-controlled* so that all transmitted signals arrive at the receiver with exactly the same power and the *near-far problem* referred to in the introduction is eliminated. Power control can often be implemented in practice with great accuracy and more details of the validity of the above assumption will be given in the following chapter. Furthermore, the systems are assumed totally *asynchronous*, that is, there is no common timing reference for the K transmitters. This is actually an advantage of

CDMA over other multiple access techniques, because all users can transmit independently and no signalling information is required.

3.1 The Binary PSK DS-CDMA System Model

System Description. For BPSK modulation, the DS-CDMA environment that we will consider is mainly based on [13] and it is shown in Figure 7 for K users.

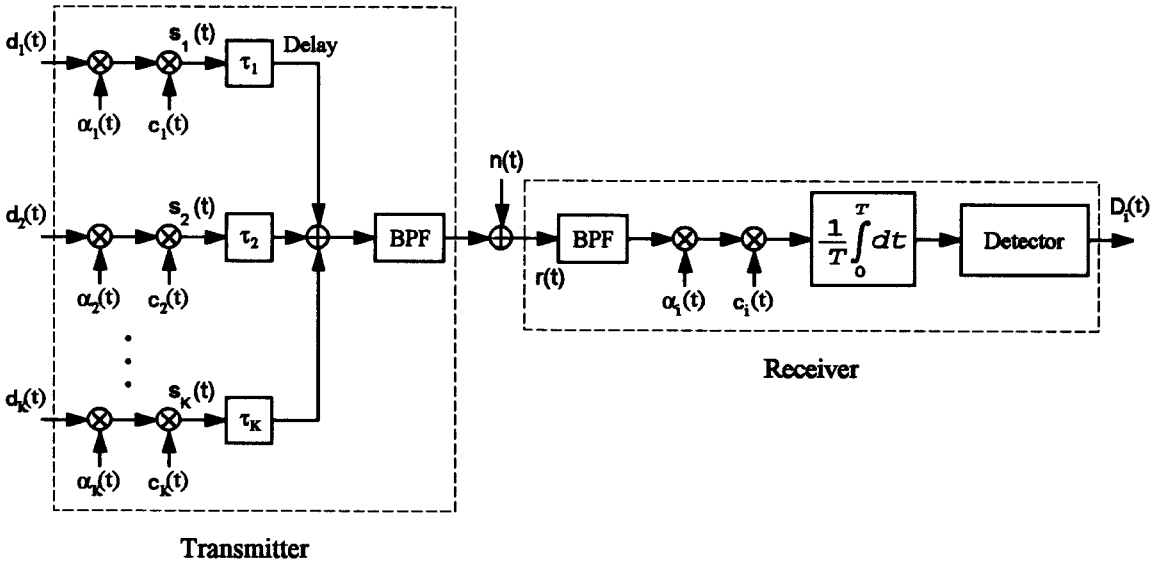

Figure 7. *Binary PSK DS-CDMA system model.*

As described in section 2.1, the *k*-th user's data signal $d_k(t)$, which represents the user's binary information sequence, is a sequence of unit amplitude, positive and negative, rectangular pulses of duration T and can be expressed as

$$d_k(t) = \sum_{j=-\infty}^{\infty} d_j^{(k)} p_T(t - jT) \tag{16}$$

where $d_j^{(k)} \in \{+1, -1\}$, $p_\tau(t) = \text{rect}(t/\tau)$ and the "rect" function has been defined in (2). Similarly, the *k*-th user's code waveform $a_k(t)$ is a sequence of unit amplitude, positive and negative, rectangular pulses of duration $T_c << T$ (which are called *chips*) and can be expressed as

$$a_k(t) = \sum_{j=-\infty}^{\infty} a_j^{(k)} p_{T_c}(t - jT_c) \tag{17}$$

where $\{a_j^{(k)}\}$ is the *k*-th user's code sequence and $a_j^{(k)} \in \{+1,-1\}$ for all the elements of $\{a_j^{(k)}\}$. The period of each user's code sequence is selected as $N = T/T_c$, and therefore there is one code period per data bit (or N chips per bit). For the BPSK case thus $N = G_p$. At each user's transmitter, $d_k(t)$ is first multiplied by $a_k(t)$ and then with the carrier $c_k(t)$, which is given by

$$c_k(t) = \sqrt{2P}\cos(\omega_0 t + \theta_k) \tag{18}$$

where *P* is the signal power, ω_0 is the centre frequency and θ_k is the phase of the *k*-th carrier. *P* and ω_0 are assumed common for all carriers. The transmitted signal $s_k(t)$ of the *k*-th user is therefore

$$s_k(t) = \sqrt{2P}a_k(t)d_k(t)\cos(\omega_0 t + \theta_k) \tag{19}$$

Since the transmitters are not time-synchronous, there is a different time delay τ_k for each signal $s_k(t)$ before it reaches the receiver, such that $0 \le \tau_k < T$ for $k = 1,2,\ldots,K$. The carrier phases θ_k are also assumed different so that $0 \le \theta_k < 2\pi$ for $k = 1,2,\ldots,K$. Thus, ignoring for the time being the two band-pass filters at the transmitters and the receiver, the received signal $r(t)$, which is actually the sum of all the transmitted signals plus noise, is given by

$$r(t) = \sum_{k=1}^{K}\sqrt{2P}a_k(t-\tau_k)d_k(t-\tau_k)\cos(\omega_0 t + \varphi_k) + n(t) \tag{20}$$

where $\varphi_k = \theta_k - \omega_0\tau_k$ and $n(t)$ is the thermal noise of the channel, which is modelled as a white Gaussian random process with (one-sided) power spectral density N_0. The received signal is then driven to a correlation receiver matched to the transmitted signal of a desired (say *i*-th) user. Without loss in generality, we shall assume that $\tau_i = \theta_i = 0$ for the *i*-th user. The output of the receiver at $t = T$ is thus

$$Z_i = \frac{1}{T}\int_0^T r(t)a_i(t)\cos\omega_0 t\, dt \tag{21}$$

Finally, the output of the correlation receiver is the input to a detector, where the estimate of $d_i(t)$ is produced, called $D_i(t)$, by taking the sign of Z_i . Thus a decision is taken on the transmitted signal $d_i(t)$. If $Z_i > 0$, the decision is that $d_0^{(i)} = +1$ was transmitted. If, however, $Z_i < 0$, the decision is that $d_0^{(i)} = -1$ was transmitted. This of course holds if the two transmission probabilities $P_0 = P\{d_0^{(i)} = +1\}$ and $P_1 = P\{d_0^{(i)} = -1\}$ are equal. If they are not, a different decision threshold can be used, according to a certain detection criterion.

Calculation of the Other-User Interference. By substituting (16) and (20) into (21), one can find [13] that, at $t = T$, the output Z_i becomes

$$Z_i = \sqrt{P/2}\, d_0^{(i)} + \sum_{\substack{k=1 \\ k\neq i}}^{K} u_{k,i}(\tau_k, \varphi_k) + \frac{1}{T}\int_0^T n(t)a_i(t)\cos\omega_0 t\, dt \tag{22}$$

where $u_{k,i}(\tau_k, \varphi_k)$ is the contribution of the *k*-th signal $s_k(t)$ to the output Z_i, defined as

$$u_{k,i}(\tau_k, \varphi_k) = A\,[d_{-1}^{(k)} R_{k,i}(\tau_k) + d_0^{(k)} \hat{R}_{k,i}(\tau_k)] \cos\varphi_k \tag{23}$$

where $A = (1/T)\sqrt{P/2}$ and

$$R_{k,i}(\tau) = \int_0^{\tau} a_k(t-\tau) a_i(t)\,dt \tag{24}$$

$$\hat{R}_{k,i}(t) = \int_{\tau}^{T} a_k(t-\tau) a_i(t)\,dt \tag{25}$$

are the *continuous-time partial cross-correlation functions* between the *k*-th and *i*-th code waveforms, defined for $0 \le \tau \le T$. In (23), it is assumed that $\omega_0 T >> 1$, so that the double frequency components of the integrand of (21) can be ignored. This condition is always satisfied in a practical CDMA system. Furthermore, using the definition (17), the above cross-correlation functions can be expressed as

$$R_{k,i}(\tau) = C_{k,i}(l-N)T_c + [C_{k,i}(l+1-N) - C_{k,i}(l-N)](\tau - lT_c) \tag{26}$$

$$\hat{R}_{k,i}(\tau) = C_{k,i}(l)T_c + [C_{k,i}(l+1) - C_{k,i}(l)](\tau - lT_c) \tag{27}$$

for $0 \le lT_c \le \tau \le (l+1)T_c \le T$, where $C_{k,i}$ is the corresponding *discrete aperiodic cross-correlation function* defined in (6). Finally, using the *even* and *odd cross-correlation functions* $\theta_{k,i}(l) = C_{k,i}(l) + C_{k,i}(l-N)$ and $\hat{\theta}_{k,i}(l) = C_{k,i}(l) - C_{k,i}(l-N)$ respectively, equation (23) becomes

$$u_{k,i}(\tau_k, \varphi_k) = A d_0^{(k)} \{\theta_{k,i}(l_k)T_c + [\theta_{k,i}(l_k+1) - \theta_{k,i}(l_k)](\tau_k - l_k T_c)\} \cos\varphi_k \tag{28}$$

for $d_0^{(k)} = d_{-1}^{(k)}$, and

$$u_{k,i}(\tau_k, \varphi_k) = A d_0^{(k)} \{\hat{\theta}_{k,i}(l_k)T_c + [\hat{\theta}_{k,i}(l_k+1) - \hat{\theta}_{k,i}(l_k)](\tau_k - l_k T_c)\} \cos\varphi_k \tag{29}$$

for $d_0^{(k)} \ne d_{-1}^{(k)}$, where l_k is an integer such that $l_k T_c \le \tau_k \le (l_k+1)T_c$.

By the above analysis. several conclusions can be drawn. Firstly, it is evident from (22) that the output of the correlation receiver contains three terms: the desired term $\sqrt{P/2}\,d_0^{(i)}$ corresponding to the desired *i*-th user, the sum of the contributions of the $K-1$ signals of the other users, which is the total interference, and finally the contribution of the thermal noise. In the absence of the noise and the interference, $Z_i = \sqrt{P/2}\,d_0^{(i)}$, so that the decision taken at the detector as described above always gives the correct result. In the presence, however, of noise and interference, Z_i becomes a random variable so that there are non-zero error probabilities $P\{Z_i > 0 \,|\, d_0^{(i)} = -1\}$ and $P\{Z_i < 0 \,|\, d_0^{(i)} = +1\}$. From the distribution of this random variable, the total error probability (probability of wrong decision at the detector) can be calculated, as shown in the following section, giving an estimate of the bit-error-rate of the CDMA system.

Secondly, it is obvious from (28) and (29) that the contribution of the *k*-th interfering signal to Z_i is proportional to the (even or odd) periodic cross-correlation function between the *k*-th and *i*-th code waveforms. Therefore, designing the code waveforms so that these cross-correlation functions are as small as possible is essential for reducing the total other-user interference and thus enhancing the system performance.

Thirdly, since all data sequences $\{d_j^k\}$ are usually sequences of independent, identically distributed random variables taking the values +1 and -1, the events $d_0^{(k)} = d_{-1}^{(k)}$ and $d_0^{(k)} \neq d_{-1}^{(k)}$ have equal probabilities so that the situations that result in (28) and (29) occur one half of the time on average. Thus for CDMA applications the odd cross-correlation function is indeed as important as the even cross-correlation function.

Finally, note that if the number of chips per bit is less than the period *N* of the code waveforms, the cross-correlation functions appearing in (28) and (29) will be *partial*, as the integration in (24) and (25) will be over a small part only of the code period. Since partial cross-correlation functions are usually much larger [10,chap.11] this will result in much greater interference.

3.1.1 System Analysis

Probability of Error without Band-Pass Filtering. Based on the previous development of the system model, and still ignoring the effect of the band-pass filters, the probability of error can be calculated by means of the distribution of the random variable Z_i. In particular the *Gaussian approximation* may be used, as explained in [3], i.e. the multiple access interference and thus Z_i may be approximated by a Gaussian random variable. This approximation holds sufficiently well when *K* and *N* become large, i.e. when many users are present in the system and when long code sequences are employed. More accurate bounds on the probability of error can be found in [21] and [22]. Because of the symmetry involved in the CDMA system when $P_0 = P_1 = 1/2$, only the case $d_0^{(i)} = +1$ can be considered, so that the probability of error, based on the Gaussian approximation, can be expressed as

$$P_e = P_0 \cdot P\{Z_i < 0 \,|\, d_0^{(i)} = +1\} + P_1 \cdot P\{Z_i > 0 \,|\, d_0^{(i)} = -1\} \Rightarrow$$

$$P_e = P\{Z_i < 0 \,|\, d_0^{(i)} = +1\} = T\{m/\sigma\} \tag{30}$$

where

$$T\{x\} = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-x^2/2} dx \tag{31}$$

is the upper tail function of the standard (zero mean, unit variance) Gaussian probability density function,

$$m = \mathrm{E}\{Z_i | d_0^{(i)} = +1\} = \sqrt{P/2} \tag{32}$$

and [13]

$$\sigma^2 = \mathrm{Var}\{Z_i | d_0^{(i)} = +1\} = \frac{P}{12N^3} \sum_{\substack{k=1 \\ k \neq i}}^{K} [2\mu_{k,i}(0) + \mu_{k,i}(1)] + \frac{N_0}{4T} \tag{33}$$

where the *correlation parameters* $\mu_{k,i}(n)$ are defined as

$$\mu_{k,i}(n) = \sum_{l=1-N}^{N-1} C_{k,i}(l)\, C_{k,i}(l+N) \tag{34}$$

In (32) and (33) the expectations are taken with respect to the data symbols $d_0^{(k)}$, $d_{-1}^{(k)}$, the phase shifts φ_k and the random delays τ_k which are assumed to be mutually independent random variables. The data symbols $d_0^{(k)}$, $d_{-1}^{(k)}$ are assumed to take values +1 and -1 with equal probability, φ_k is assumed uniformly distributed in $[0, 2\pi]$ and τ_k uniformly distributed in $[0, T]$. Expression (33) can be further simplified since it is suggested in [15] that $r_{k,i} \equiv 2\mu_{k,i}(0) + \mu_{k,i}(1) \cong 2N^2$. This value of $2N^2$ is actually the expectation of $r_{k,i}$ when pure random code sequences are employed, and is a good approximation as *N* increases. For example, for *N*=127 that was used in the simulations of the following section, a typical value for $r_{k,i}$ for the first two of the Gold sequences that were generated was $r_{k,i} = 3.1782 \times 10^4$ while $2N^2 = 3.2258 \times 10^4$. Using this approximation, (33) becomes

$$\sigma^2 \cong \frac{P(K-1)}{6N} + \frac{N_0}{4T} \tag{35}$$

so that

$$P_e = T\left\{\left(\frac{K-1}{3N} + \frac{N_0}{2E_b}\right)^{-1/2}\right\} \tag{36}$$

where $E_b = PT$ is the energy per data bit. This relation is very useful as it relates P_e directly with the number of users *K*. Thus by substituting appropriate values for the system parameters N, E_b, N_0, one can find the maximum number of users *K* so that the probability of error does not exceed a certain value. An estimate for the *capacity* of the system can thus be obtained, as shown in the following section. It is also evident from (36) that, since $N = G_p$, the processing gain is a very important parameter in a CDMA system, as an increase in the processing gain leads directly to an increase in capacity. Notice finally that the *signal-to-noise-plus-interference ratio* at the output of the *i*-th correlation receiver is

$$SNR_{out} = \frac{E^2\{Z_i | d_0^{(i)} = +1\}}{\mathrm{Var}\{Z_i | d_0^{(i)} = +1\}} = \frac{m^2}{\sigma^2} = \left(\frac{K-1}{3N} + \frac{N_0}{2E_b} \right)^{-1} \tag{37}$$

so that $P_e = T\{\sqrt{SNR_{out}}\}$. The signal-to-noise-plus-interference ratio is an important performance measure together with the probability of error P_e.

Probability of Error with Band-Pass Filtering. Since in a real-life application the CDMA signal has to use a limited amount of spectrum and to avoid interference with other signals at nearby frequencies outside this spectral band, each user's signal $s_k(t)$ has to be band-pass filtered before being transmitted. Due to the linearity of the filters, this is equivalent to taking the sum of all signals $s_k(t)$, $k = 1,2,\ldots,K$ and then band-pass filtering as shown in Figure 7. The necessary bandwidth for transmission of the CDMA signal is $B_{ss} = 1/T_c$ centred at frequency $f_0 = \omega_0 / 2\pi$, i.e. the transmitter filter pass band should extend from $f_0 - 1/2T_c$ to $f_0 + 1/2T_c$. At the receiver, $r(t)$ is passed through a similar band-pass filter in order to eliminate the thermal noise as well as the interference from other signals outside the desired spectral band.

The various effects of the transmitter and receiver filters on the autocorrelation curves of the transmitted signals are studied in [10,chap.8]. In particular, apart from the delay that is introduced by the filters, the autocorrelation curves are broadened, introducing *correlation loss.* False clocking in the code tracking loop of the coherent receiver can also occur due to the non-linear amplitude and phase response of the filters. However, the exact calculation of the effect of the filters on the system performance, especially if non-linear amplitude and phase distortion is considered, is rather involved and a simple approximation can be adopted instead, by assuming that the other-user interference behaves as *white Gaussian noise* in the frequency band that extends from $f_0 - 1/2T_c$ to $f_0 + 1/2T_c$. In this case the system is equivalent to a BPSK system in the presence of white Gaussian noise of (one sided) power spectral density $I_0 = (K-1)P/B_{ss} + N_0$, since all users are assumed to transmit at equal power *P*. The probability of error of such a system is [23,pp.275-279]

$$P_e = T\left\{\sqrt{2EUE_{equ}}\right\} = T\left\{\sqrt{2E_b / I_0}\right\} \tag{38}$$

or equivalently

$$P_e = T\left\{\left(\frac{2E_b}{(K-1)PT_c + N_0} \right)^{1/2}\right\} = T\left\{\left(\frac{K-1}{2N} + \frac{N_0}{2E_b} \right)^{-1/2}\right\} \tag{39}$$

since $B_{ss} = 1/T_c$, $E_b = PT$ and $T/T_c = N$. Note that this expression resembles very much to equation (36) and the only difference is that the factor of 3 in (36) is here replaced by a factor of 2. Also, the signal-to-noise-plus-interference ratio at the output of the receiver can now be found as

$$SNR_{out} = 2G_P \cdot SNR_{in} = 2G_P \frac{P}{I_0 B_{ss}} = 2\frac{T}{T_c}\frac{P}{I_0 / T_c} = 2\frac{E_b}{I_0} = 2EUE_{equ} \tag{40}$$

where SNR_{in} is the signal-to-noise-plus-interference ratio at the input of the receiver, or

$$SNR_{out} = \left(\frac{K-1}{2N} + \frac{N_0}{2E_b}\right)^{-1} \tag{41}$$

so that again $P_e = T\{\sqrt{SNR_{out}}\}$. Finally, the system performance is expected to be deteriorated in this case, since the SNR_{out} is lower than in (37), giving a higher probability of error. This is natural, as the filters cause correlation loss and therefore degrade the receiver performance.

3.1.2 Experimental Results

Illustration of System Operation. In order to estimate the accuracy of the above presented theoretical analysis, an asynchronous BPSK direct sequence CDMA system was implemented using MATLAB, based on Figure 7. Experimental results were thus obtained for the probability of error P_e as a function of the number of users *K* as well as the E_b / I_0 ratio at the receiver, with and without the transmitter and receiver filters.

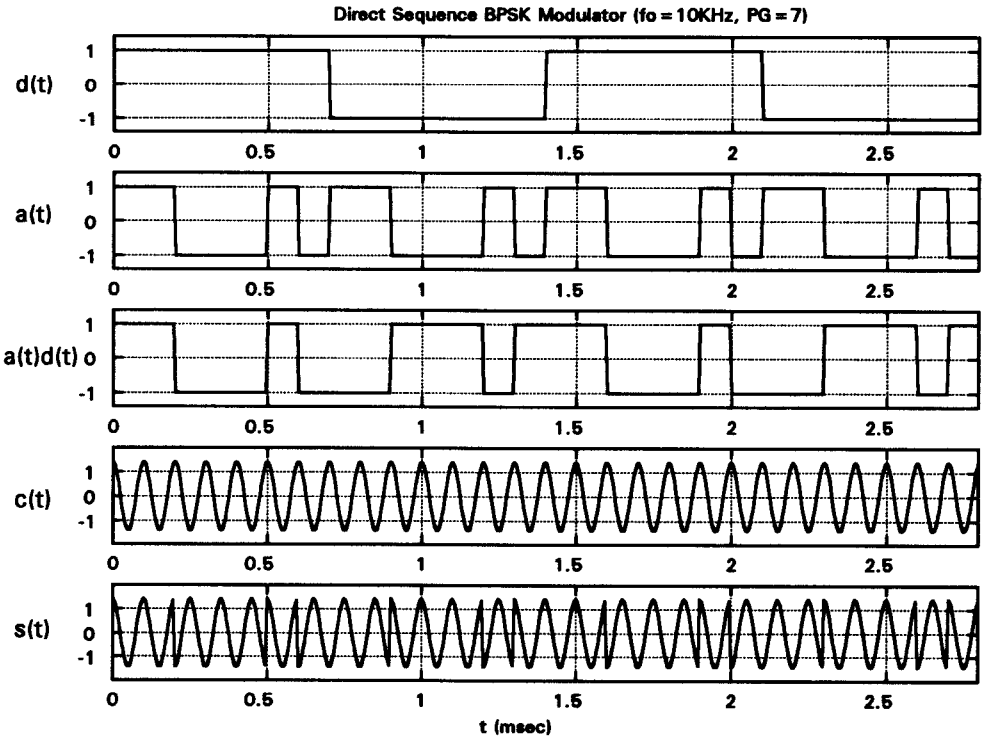

Figure 8. *DS-BPSK modulation for a single transmitter.*

Before presenting these results, an illustration of the DS-BPSK modulation and demodulation procedures is shown in Figures 8-12. In Figure 8, the modulation procedure is shown for a single transmitter. The data signal $d(t)$ corresponds to the data "0101" (note that a "0" corresponds to $d(t)=+1$ while a "1" corresponds to $d(t)=-1$) while the code waveform $a(t)$ corresponds to the *m*-sequence "0011101" of length 7. Note that in this case $N=G_p=7$. The carrier $c(t)$ has initial phase $\theta=0$ and frequency $f_0=1/T_c\equiv f_c=10KHz$, where $\omega_0=2\pi f_0$, so that there is one only carrier period per chip. Finally, the carrier power is P=1/2, so that the transmitted signal $s(t)$ has unity amplitude. Note that in practice, the carrier is much faster and the code sequence much longer.

In Figure 9, the combination of two signals $s_1(t)$ and $s_2(t)$ in a CDMA environment is illustrated, as well as the effects of the transmitter filter and the noise. Here, again P=1/2 and $f_c=1/T_c=10KHz$, while $f_0=2f_c=20KHz$, so that there are two carrier periods per chip. Because only digital signal representations can be used in the simulations, all signals are sampled at a sampling frequency $f_s=20f_0=0.4MHz$, so that there are 20 samples per carrier period. Gold sequences of length 127 (generated by 7-stage shift registers) are used as code sequences, while 10 chip periods are only shown in Figure 9. The initial carrier phases θ_1, θ_2 are uniform random variables in $[0,2\pi]$, and the delay τ_2 is a uniform random variable in $[0,T]$, while $\tau_1=0$. The filtered signal $s_f(t)$ is produced by passing $s_1(t)+s_2(t)$ through a bandpass filter with pass band from $f_1=f_0-f_c/2$ (15KHz) to $f_2=f_0+f_c/2$ (25KHz), while the signal $r(t)$ also includes white Gaussian noise of power equal to the signal power, P.

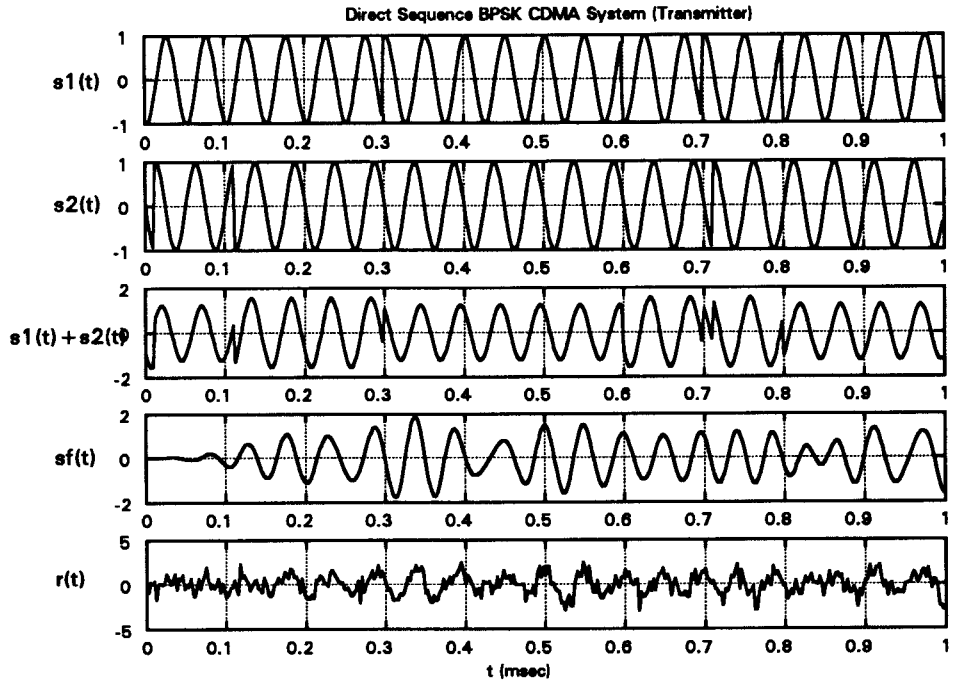

Figure 9. *Combination of signals in a CDMA environment.*

The transmitter filter is a digital, bandpass, elliptic filter of order 12, whose amplitude response $G(F)$, where $F = f / f_s$, is shown in Figure 10 in logarithmic scale. The frequency axis is normalised with respect to the sampling frequency. For example, the normalised lower cut-off frequency is $F_1 = (f_0 - f_c/2)/f_s = 0.0375$, while the normalised upper cut-off frequency is $F_2 = (f_0 + f_c/2)/f_s = 0.0625$. In order to find the actual frequency, one has to multiply the frequency shown in Figure 10 with the sampling frequency $f_s = 0.4MHz$. Finally, the ripple in the pass band is 0.1dB and the attenuation in the stop band is 60dB.

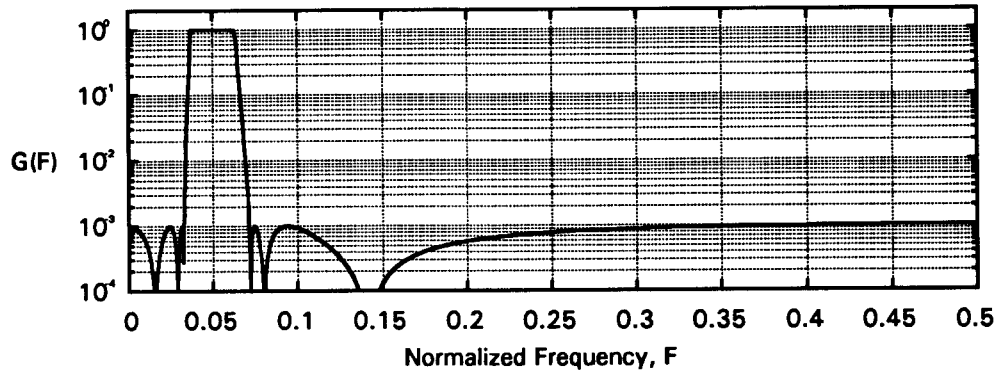

Figure 10. *Amplitude response of the transmitter filter.*

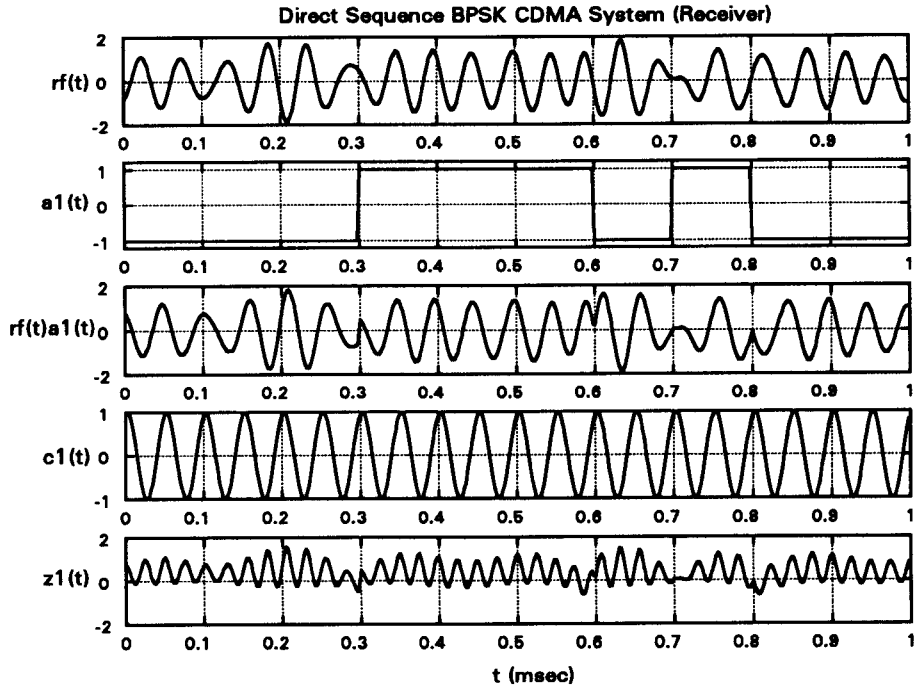

Figure 11. *DS-BPSK demodulation.*

In Figure 11, the demodulation procedure at the receiver is illustrated. The received signal $r(t)$ is first bandpass filtered giving the signal $r_f(t)$ and then multiplied with the code

waveform $a_1(t)$ and the carrier $c_1(t)$ of the first user, producing the signal $z_1(t) = r(t)a_1(t)c_1(t)$. The receiver filter is the same with the transmitter filter, but in the simulations the data were passed through the receiver filter in reverse order, effectively removing all phase distortion. The signal $z_1(t)$, which is the input of the integrator, is positive most of the time, so that the output Z_1 of the correlation receiver is also positive, corresponding to a correct decision, as the data bit "0" is being sent in this example.

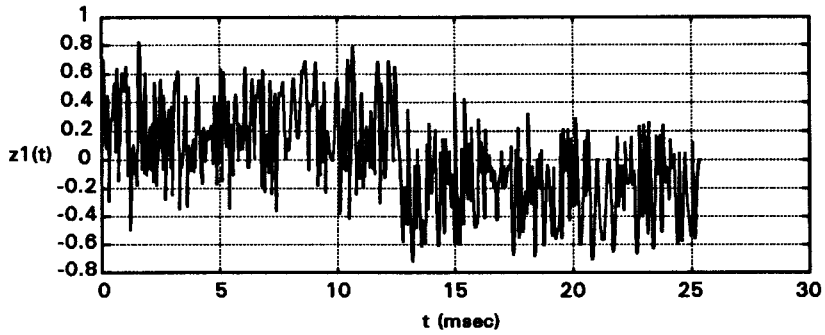

Figure 12. *Input of the integrator in a DS-BPSK receiver.*

The decision procedure is better illustrated in Figure 12, where the signal $z_1(t)$ is shown for two data bit periods corresponding to the data "01". It is obvious in this figure that the mean value of $z_1(t)$ is positive or negative corresponding to the data that is being sent. Naturally, the situation is much worse if more than two users are present in the system.

Results on System Performance. The performance results of the DS-BPSK CDMA system are shown in Figures 13-14, where the experimental bit error rate and the theoretical probability of error are shown as functions of the number *N* of users and of the E_b / I_0 ratio at the receiver. In the experiments, the parameters *P*=1/2, $f_c = 1.25MHz$, $f_0 = 2f_c = 2.5MHz$, $f_s = 20f_0 = 50MHz$ were used, so that there are still 20 samples per carrier period and two carrier periods per chip. Gold sequences of length *N*=127 (generated by 7-stage shift registers) were used as code sequences. Thus the processing gain is $G_P = N = 127$ and the *bit rate* is $R = 1/T = f_c/127 \cong 9800$ bits/sec. The number of users varies from 20 to 64, which is almost the maximum number of balanced Gold codes of length 127 (actually all the 63 non-maximal as well as the one only maximal length sequence were used). The noise power is kept equal to the signal power *P*. In this case, note that $E_b / N_0 = PT/(P/B_{ss}) = T/T_c = N = G_P$ and this value was used for the theoretical calculations (36) and (39).

A total of 10000 data bits were used to test the system with equal transmission probabilities $P_0 = P_1 = 1/2$. The experiment was implemented by 500 iterations of 20 bits at a time, while a different set of carrier phases θ_k, delays τ_k and code sequences were used for each iteration (a different *permutation* of the 64 code sequences was actually used each time) so

that an average BER could be calculated with respect to all these parameters. The signal-to-noise-plus-interference ratio (SNR_{out}) at the output of the receiver was also calculated in the experiment, and from this another estimate for the probability of error was obtained through the relation $P_e = T\{\sqrt{SNR_{out}}\}$.

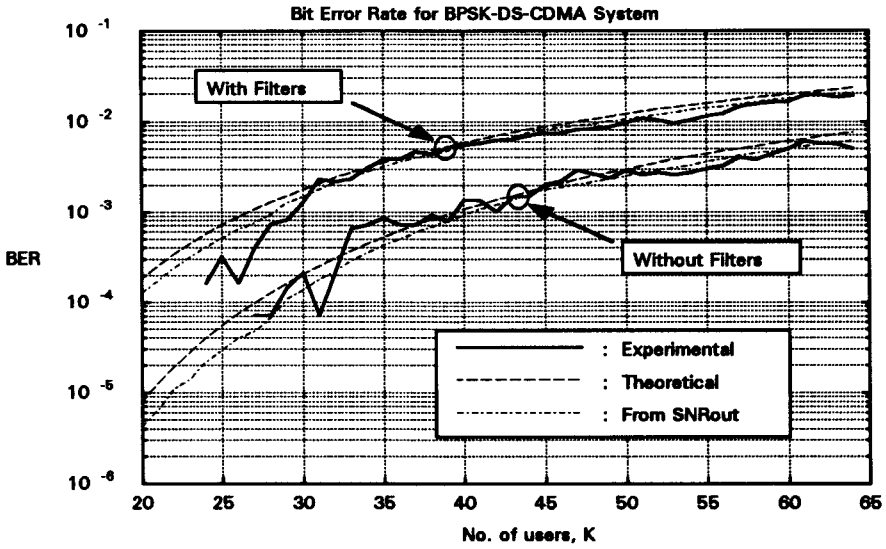

Figure 13. *BER performance of a DS-BPSK CDMA system with and without filters.*

In Figure 13, the experimental bit error rate (BER) of a system with and without bandpass filters is shown as a function of the number *N* of users, together with the theoretical probability of error as well as the one estimated through the SNR_{out}. It can be seen that the experimental BER (percentage of bits that were incorrectly detected), the theoretical probability of error obtained by (36) (without filters) or (39) (with filters) and the probability of error obtained through the SNR_{out} all agree. If the BER must not exceed 10^{-3}, it can be seen from the figure that the system can accommodate up to *K*=40 users without filters and up to 27 users with filters. This is expected as the system performance is deteriorated in the presence of filters. However, *K*=27 is a much more realistic result since in a practical CDMA system filters are always used. Therefore the capacity of such a system can be estimated as *K*=27 users. However this is a *soft* capacity limit since a few more users can be accommodated at the expense of a little inferior performance, in contrast with other multiple access techniques where the capacity (i.e. available number of channels) is fixed.

We shall see in the next chapter that in a mobile cellular environment there is always a non-zero probability that the BER exceeds a certain value, and a different technique will be used in order to estimate the system capacity. In particular, BER can be estimated as a function of the E_b / I_0 ratio at the receiver, as shown in Figure 14. This figure was obtained by plotting the experimental BER as well as the theoretical probability of error (given by (36) or (39)) versus E_b / I_0 (given by (40) and (41)) for different values of *K*. It can be seen that an E_b / I_0 ratio of

approximately 7 dB is desired in order to maintain BER below 10^{-3} (with filters included). This will be used in the next chapter in order to estimate the capacity of a multiple-cell CDMA system.

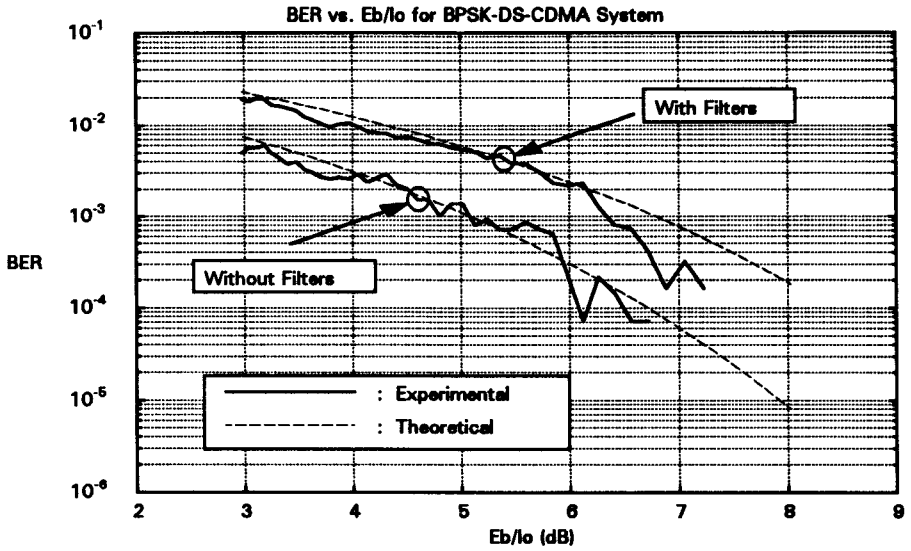

Figure 14. *BER as a function of E_b / I_0 for a DS-BPSK CDMA system.*

Finally, it should be noted that in a practical system a typical value of the total bandwidth $B_{ss} = 1/T_c = f_c$ is 1.25MHz, while a typical value for the *bit rate* $R = 1/T$ is 10Kbits/sec for an acceptable quality vocoder for digital telephony applications [8,9], resulting in a processing gain $G_p \cong 125$, which is very close to the value of 127 that was used in the experiment. Hence, the system implementation is very close to a real-life application. However, the carrier frequency in a practical system is much higher (about 800MHz for digital cellular telephony [9]). This could not be implemented in the simulations, as it would require a huge amount of signal samples for the same number of bits. Still, the accuracy of the results is not affected since the system performance does not depend on the carrier frequency.

3.2 The Quadrature PSK DS-CDMA System Model

System Description. For QPSK modulation, the DS-CDMA system that will be considered is mainly based on [24] and is shown in Figure 15 for *K* users.

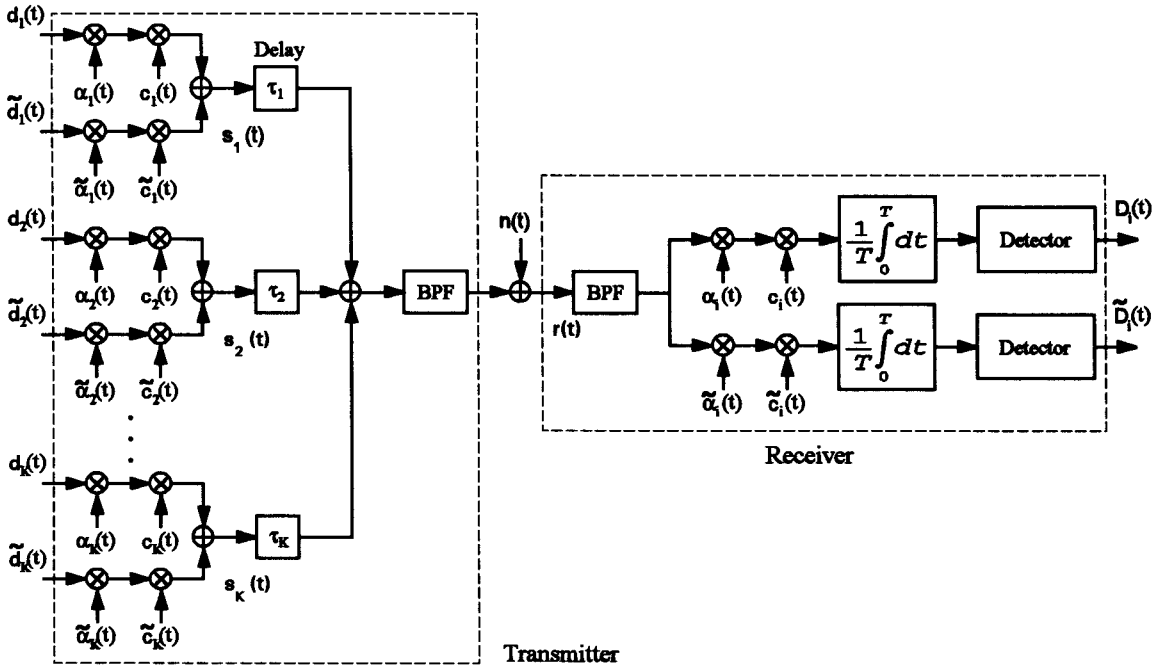

Figure 15. *Quadrature PSK DS-CDMA system model.*

This system is very similar to the BPSK system described in the previous section. The main difference is that data bits of each user are now grouped in *symbols* of two bits. Thus the *k*-th user's binary information is now represented by two different data signals $d_k(t)$ and $\tilde{d}_k(t)$ that can be expressed as

$$d_k(t) = \sum_{j=-\infty}^{\infty} d_j^{(k)} p_T(t - jT) \tag{42}$$

$$\tilde{d}_k(t) = \sum_{j=-\infty}^{\infty} \tilde{d}_j^{(k)} p_T(t - jT) \tag{43}$$

where $d_j^{(k)} = \delta_{2j}^{(k)}$, $\tilde{d}_j^{(k)} = \delta_{2j+1}^{(k)}$ and $\delta_j^{(k)} \in \{-1, +1\}$ represent the initial data sequence. Thus, $d_k(t)$ is formed by the even data bits and $\tilde{d}_k(t)$ by odd ones. At each transmitter, the two data signals are first multiplied by two different code waveforms $a_k(t)$ and $\tilde{a}_k(t)$, and then by two different carriers

$$c_k(t) = \sqrt{P}\cos(\omega_0 t + \theta_k) \tag{44}$$

$$\tilde{c}_k(t) = \sqrt{P}\sin(\omega_0 t + \theta_k) \tag{45}$$

while the transmitted signal is formed by the sum of these two components:

$$s_k(t) = \sqrt{P}\cdot[a_k(t)d_k(t)\cos(\omega_0 t + \theta_k) + \tilde{a}_k(t)\tilde{d}_k(t)\sin(\omega_0 t + \theta_k)] \tag{46}$$

Note that now the amplitude of the carriers has changed from $\sqrt{2P}$ to $\sqrt{P}$ so that the power of the transmitted signal $s_k(t)$ still remains equal to *P*. Ignoring again the effects of the bandpass filters, the received signal $r(t)$ can be expressed as

$$r(t) = \sqrt{P}\sum_{k=1}^{K} a_k(t-\tau_k)d_k(t-\tau_k)\cos(\omega_0 t + \varphi_k) +$$

$$+\sqrt{P}\sum_{k=1}^{K} \tilde{a}_k(t-\tau_k)\tilde{d}_k(t-\tau_k)\sin(\omega_0 t + \varphi_k) + n(t) \tag{47}$$

The received signal is the input to two different correlation receivers matched to the signal $s_i(t)$, whose outputs at $t = T$ is

$$Z_i = \frac{1}{T}\int_0^T r(t)a_i(t)\cos\omega_0 t\, dt \tag{48}$$

$$\tilde{Z}_i = \frac{1}{T}\int_0^T r(t)\tilde{a}_i(t)\sin\omega_0 t\, dt \tag{49}$$

Finally, Z_i and $\tilde{Z}_i$ are the inputs to two detectors and, based on their signs, a decision is made on the transmitted signals $d_i(t)$ and $\tilde{d}_i(t)$ in a similar way as in the BPSK case. Estimates of $d_i(t)$ and $\tilde{d}_i(t)$ are thus obtained, named $D_i(t)$ and $\tilde{D}_i(t)$.

Calculation of the Other-User Interference. Again, by substituting (42) and (47) in (48) (and similarly (43) and (47) in (49)), it can be found [24] that, at $t = T$, the outputs Z_i and $\tilde{Z}_i$ become

$$Z_i = \sqrt{P/4}\, d_0^{(i)} + \sum_{\substack{k=1\\k\neq i}}^{K} u_{k,i}(\tau_k, \varphi_k) + \frac{1}{T}\int_0^T n(t)a_i(t)\cos\omega_0 t\, dt \tag{50}$$

$$\tilde{Z}_i = \sqrt{P/4}\, \tilde{d}_0^{(i)} + \sum_{\substack{k=1\\k\neq i}}^{K} \tilde{u}_{k,i}(\tau_k, \varphi_k) + \frac{1}{T}\int_0^T n(t)\tilde{a}_i(t)\sin\omega_0 t\, dt \tag{51}$$

where $u_{k,i}(\tau_k, \varphi_k)$ is the contibution of the *k*-th signal $s_k(t)$ to the output Z_i and $\tilde{u}_{k,i}(\tau_k, \varphi_k)$ is the contibution of $s_k(t)$ to the output $\tilde{Z}_i$. Assuming that the code waveforms are $\tilde{a}_k(t) = a_k(t)$ for all $k = 1, 2, \ldots, K$, these contributions can be expressed as

$$u_{k,i}(\tau_k, \varphi_k) = A[d_{-1}^{(k)}R_{k,i}(\tau_k) + d_0^{(k)}\hat{R}_{k,i}(\tau_k)]\cos\varphi_k +$$

$$+A[\tilde{d}_{-1}^{(k)}R_{k,i}(\tau_k) + \tilde{d}_0^{(k)}\hat{R}_{k,i}(\tau_k)]\sin\varphi_k \tag{52}$$

$$\tilde{u}_{k,i}(\tau_k,\varphi_k) = -A\,[d_{-1}^{(k)}R_{k,i}(\tau_k) + d_0^{(k)}\hat{R}_{k,i}(\tau_k)]\sin\varphi_k + \\ + A\,[\tilde{d}_{-1}^{(k)}R_{k,i}(\tau_k) + \tilde{d}_0^{(k)}\hat{R}_{k,i}(\tau_k)]\cos\varphi_k \tag{53}$$

where $R_{k,i}(\tau)$ and $\hat{R}_{k,i}(\tau)$ are defined as in (24) and (25). Similar results can be found for $\tilde{a}_k(t) \neq a_k(t)$, though a slightly more complicated notation has to be used. Furthermore, the interference contributions $u_{k,i}(\tau_k,\varphi_k)$ and $\tilde{u}_{k,i}(\tau_k,\varphi_k)$ can be expressed in terms of the discrete aperiodic cross-correlation functions $\theta_{k,i}(l)$ and $\hat{\theta}_{k,i}(l)$, as in section 3.1. However, it is already obvious from equations (52) and (53) that the total interference is the same as in the BPSK system, since there are two independent terms like that in (23) (therefore the interference is doubled), while the factor *A* is now divided by $\sqrt{2}$ (therefore the interference power is divided by 2). This result can be used directly for the calculation of the probability of error of the QPSK system, as will be explained in the next section.

3.2.1 System Analysis

Probability of Error without Band-Pass Filtering. Based on the previous development of the system model, and ignoring the effect of the band-pass filters, the probability of error can be calculated by means of the distribution of the random variables Z_i and $\tilde{Z}_i$, as in the BPSK case. Considering only the case $d_0^{(i)} = +1$ again (and similarly $\tilde{d}_0^{(i)} = +1$), and using the Gaussian approximation, the probability of error of the two correlation receivers is the same, and can be expressed as

$$p = P\{Z_i < 0 \,|\, d_0^{(i)} = +1\} = P\{\tilde{Z}_i < 0 \,|\, \tilde{d}_0^{(i)} = +1\} = T\{m/\sigma\} \tag{54}$$

where

$$m = \mathrm{E}\{Z_i \,|\, d_0^{(i)} = +1\} = \mathrm{E}\{\tilde{Z}_i \,|\, \tilde{d}_0^{(i)} = +1\} = \sqrt{P/4} \tag{55}$$

and, using the observation the interference is the same as in the BPSK system, σ is given by (33) and approximated by (35). Thus, defining $E_s = PT$ as the energy per data symbol and since each symbol contains two bits, so that the energy per bit is $E_b = E_s/2 = PT/2$, (54) becomes

$$p = T\left\{\left(\frac{2(K-1)}{3N} + \frac{N_0}{2E_b}\right)^{-1/2}\right\} \tag{56}$$

The total symbol error probability for the QPSK system can be found as $P_s = 1-(1-p)^2 = 2p - p^2$ [23,pp.283-290] so that the bit error probability is $P_e = P_s/2 = p - p^2/2 \cong p$:

$$P_e \cong T\left\{\left(\frac{2(K-1)}{3N}+\frac{N_0}{2E_b}\right)^{-1/2}\right\} \tag{57}$$

This relation is very similar to (36), except for a factor of 2 that shows that the QPSK system behaves as having twice the number of users, comparing to the BPSK system. Thus, although the data rate can be doubled with the QPSK system (since two data bits can be transmitted simultaneously), the other-user interference is higher, resulting in inferior performance. Finally, the signal-to-noise-plus-interference ratio at the output of the i-th correlation receiver is

$$SNR_{out}=\frac{E^2\{Z_i|d_0^{(i)}=+1\}}{\mathrm{Var}\{Z_i|d_0^{(i)}=+1\}}=\frac{E^2\{\tilde{Z}_i|\tilde{d}_0^{(i)}=+1\}}{\mathrm{Var}\{\tilde{Z}_i|\tilde{d}_0^{(i)}=+1\}}=\frac{m^2}{\sigma^2}=\left(\frac{2(K-1)}{3N}+\frac{N_0}{2E_b}\right)^{-1} \tag{58}$$

so that $P_e \cong T\{\sqrt{SNR_{out}}\}$, as in the BPSK case.

Probability of Error with Band-Pass Filtering. Assuming again that in the presence of transmitter and receiver filters the other-user interference behaves as white Gaussian noise of (one sided) power spectral density $I_0=(K-1)P/B_{ss}+N_0$ in the frequency band that extends from $f_0-1/2T_c$ to $f_0+1/2T_c$, the bit error probability is [23,pp.283-290] $P_e=p-p^2/2\cong p$, where

$$p=T\left\{\sqrt{2EUE_{\text{equ}}}\right\}=T\left\{\sqrt{2E_b/I_0}\right\} \tag{59}$$

or equivalently

$$P_e \cong T\left\{\left(\frac{2E_b}{(K-1)PT_c+N_0}\right)^{1/2}\right\}=T\left\{\left(\frac{2(K-1)}{2N}+\frac{N_0}{2E_b}\right)^{-1/2}\right\} \tag{60}$$

since $B_{ss}=1/T_c$, $E_b=PT/2$ and $T/T_c=N$. Again, this relation is very similar to (39), except for the same factor of 2 that shows that even with the transmitter and receiver filters the QPSK system behaves as having twice the number of users comparing to the BPSK system. Note also that this expression resembles very much to equation (57) and the only difference is that the factor of 3 in (57) is here replaced by a factor of 2. Thus, the system performance is expected to deteriorate in the presence of filters. Finally, the signal-to-noise-plus-interference ratio at the output of the receiver is again $SNR_{out}=2EUE_{\text{equ}}$, or

$$SNR_{out}=\left(\frac{2(K-1)}{2N}+\frac{N_0}{2E_b}\right)^{-1} \tag{61}$$

so that $P_e \cong T\{\sqrt{SNR_{out}}\}$.

3.2.2 Experimental Results

As for the BPSK case, an asynchronous direct sequence QPSK CDMA system was implemented using MATLAB, based on Figure 15, and experimental results were obtained for the probability of error P_e as a function of the number of users *K* as well as the E_b / I_0 ratio at the receiver, with and without the transmitter and receiver filters. These results are shown in Figures 16-17.

In Figure 16, the experimental bit error rate (BER) of a system *without* bandpass filters is shown as a function of the number *N* of users, together with the theoretical probability of error as well as the one obtained through the SNR_{out}. Here the parameters *P*=1/2, $f_c = 1/T_c = 1.25MHz$, $f_0 = 2f_c = 2.5MHz$ and $f_s = 20f_0 = 50MHz$ were used again, and Gold sequences of length *N*=127 (generated by 7-stage shift registers) were used as code sequences. The same code sequence was used for the in-phase as well as the quadrature component of the QPSK signals. The number of users varied from 10 to 64, while the noise power was kept equal to the signal power *P*. A total of 10000 data symbols were used to test the system, in 500 iterations of 20 bits each time, as for the BPSK system. The two experiments were thus very similar. The main difference, however, is that for the same values of T and T_c, the QPSK system has twice the bit rate $R = 2/T = 2f_c/127 \cong 19600$ bits/sec, and half the processing gain $G_p = B_{ss} / (2/T) = 63.5$.

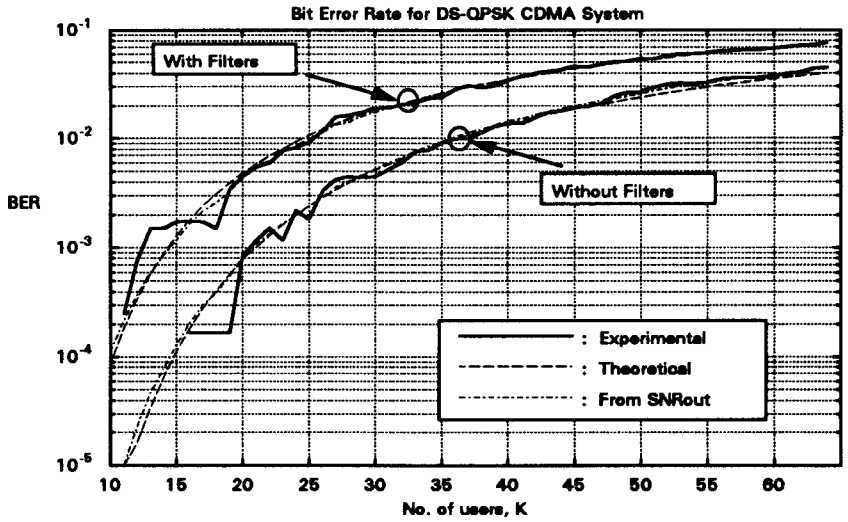

Figure 16. *BER performance of a DS-QPSK CDMA system with and without filters.*

In Figure 16, it can be seen that the experimental BER, the theoretical probability of error obtained by (57) (without filters) or (60) (with filters) and the probability of error obtained through the SNR_{out} all agree. The system performance is again deteriorated in the presence of filters, thus the capacity is expected to decrease, as for the BPSK system. Indeed, if $BER < 10^{-3}$ is required, the capacity is 21 users without filters and 14 users with filters. Note that this capacity is

almost the half of the respective capacity of the BPSK system. However, the two systems have different processing gain and different bit rates, therefore a straightforward comparison is not possible. A comparison between the two systems is presented in the following section.

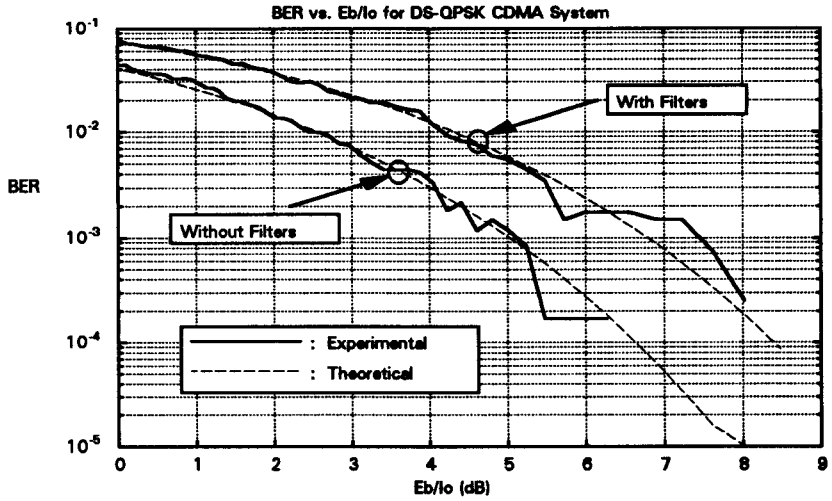

Figure 17. *BER as a function of E_b / I_0 for a DS-QPSK CDMA system.*

Finally, in Figure 17, the bit error rate is given as a function of E_b / I_0. It can be seen from this figure that the results are also the same with the BPSK system. This can be easily explained by comparing equation (38) of the BPSK system with (59) and (60) of the QPSK system. Again, an E_b / I_0 ratio of approximately 7 dB (with filters) is required for a performance $BER < 10^{-3}$.

3.3 Performance Comparison of BPSK and QPSK Systems

The BPSK and QPSK systems were implemented with the same values of T and T_c, so that N also remains the same and the same code sequences can be used. However, this means that the QPSK system has twice the bit rate and half the processing gain of the BPSK system, as noted in the previous section, therefore a straightforward comparison is not possible. A comparison can be made only if a single user of the QPSK system is regarded as a pair of users of the BPSK system, since for the QPSK system the bit rate is double. Thus we can consider that the number of users of the QPSK system is two times the number that was actually used in the experiment. This way the two systems can be considered to have the same bit rate and processing gain, and the result is that the curves of Figures 16-17 are shifted to the right, as shown in Figure 18, where the performance of BPSK and QPSK systems is compared (with filters included). It can be seen with this modification that the QPSK system has slightly better performance than the BPSK. In particular, the capacity is about 27 users for BPSK and 28 users for QPSK, with a required performance $BER < 10^{-3}$.

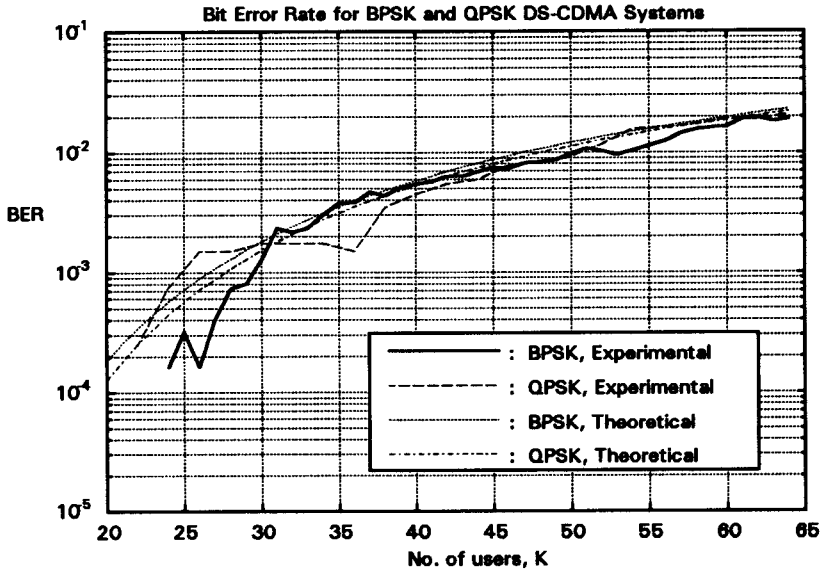

Figure 18. *DS BPSK and QPSK performance comparison (BER vs. number of users).*

In Figure 19, the BER performance of the two systems is shown as a function of E_b / I_0, with filters included. It can be seen that again, the performance is almost the same, and the two theoretical curves coincide. An E_b / I_0 ratio of 7 dB approximately is required for both systems for a required performance $BER < 10^{-3}$. This value will be used in the next chapter for the evaluation of the capacity of a multiple-cell system, but some modifications will be necessary in order to take other factors into account, such as multipath fading and error correction coding, that are present in a real application.

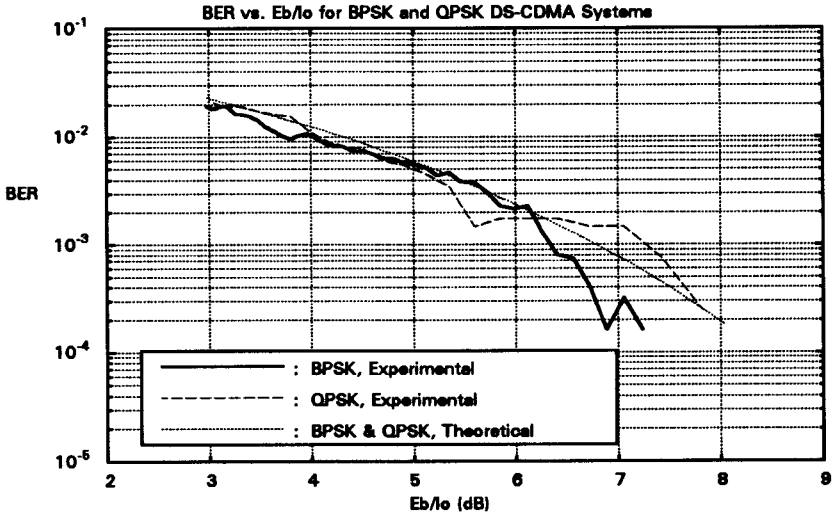

Figure 19. *DS BPSK and QPSK performance comparison (BER vs. E_b / I_0).*

Since a processing gain of 127 was used for the CDMA systems, this means that the total spread spectrum bandwidth used is 127 times the necessary bandwidth for transmission of the data signals. Therefore, using the same total bandwidth with another multiple access

technique, such as FDMA or TDMA, would provide a capacity of almost 127 users. From this simple comparison, it is obvious that the spectrum efficiency of the CDMA system is much inferior to that of other techniques. However, this situation can be remedied through voice activity detection and sectorisation, as explained in the introduction and analysed in the following chapter. Also, CDMA makes much more efficient frequency reuse since it reuses the whole available spectrum in all cells. Thus, as shown in the next chapter, the capacity of CDMA can exceed the capacity of other techniques.

For all the systems analysed and simulated in this chapter, we made the assumption that the code period *N* is equal to the processing gain T/T_c, resulting in one code period per bit. Thus, the cross-correlation functions involved are not partial and the resulting interference is less. However, in practical applications *long* code periods are often desired (e.g. in the order of $N = 2^{30} - 1$), not only because of the privacy feature they provide, but mainly because in this case the number of available individual codes is extremely large (in our experiment, only 64 codes were available). Also, there is flexibility with respect to multiple bit rates and variable processing gains, and there is no need to change the code when a mobile crosses a cell boundary, i.e. in a hand-off situation. In [25] it is shown that the loss in the BER performance from the use of such long codes is tolerable. In particular, for a BER performance of 10^{-3}, the difference in the required E_b / I_0 ratio is in the order of 1 dB. Thus, because of their advantages, long codes may often be preferred in a practical application.

Finally, it should be noted that the results obtained in this chapter agree with the respective results of Chapter 3 of [26], although in [26] the theoretical and experimental results do not agree so well as in this report. However, a straightforward comparison between these results is difficult, since different code sequences and generally different system parameters were used in [26].

4. ANALYSIS OF A MULTIPLE-CELL POWER CONTROLLED DS-CDMA SYSTEM

A mobile cellular communications network consists of a number of *mobile subscribers,* or *users* that communicate with one or multiple *cell sites*, or *base stations*, which are interconnected with the public switched telephone network. The systems that have been studied in the previous chapter can serve as a model for a single-cell CDMA system only, because *K* perfectly power-controlled users have been assumed to communicate with a single cell site, so that all the users' signals arrive at the cell site receiver with the same power, *P*, while no propagation attenuation has been considered. In a multiple-cell environment, however, the situation becomes more complicated for the *forward link* (from cell sites to users) and the *reverse link* (from users to cell sites) as well.

For the reverse link, for example, users are power controlled by the cell sites of their own cell. Thus, their interference to another cell is no longer constant, but varies proportionally to the attenuation in the path to that cell and inversely proportionally to the attenuation in the path to their own cell site. Therefore a propagation model has to be used in order to calculated the total interference from all cells, since it is this interference that determines the capacity of the link. Also, for the forward link, power control takes the form of *power allocation* at the transmitter of the cell site, according to the needs of individual users. Therefore, in general, the signals that correspond to different users do not arrive at another user's receiver with equal power, and the model of the previous chapter cannot be used directly. Several modifications have then to be considered for both links. The reverse link is analysed in Section 4.1, while the forward link in Section 4.2.

Notice however that although in the single-cell case it is actually possible to analyse or simulate the whole CDMA environment by adding *K* equal-power DS modulated signals, this kind of analysis becomes very difficult in the multiple-cell case since a huge number of such signals is involved. The situation is further complicated due to multipath fading and shadowing losses that are present in mobile environment. For this reason, an analysis based on the E_b / I_0 ratio at the receiver will be considered in this chapter. In [8] it is proposed that, assuming direct sequence QPSK modulation and coherent receiver, the required value for E_b / I_0 is 5dB (3.16), in order to ensure that $BER \leq 10^{-3}$. This takes account of not only the other-user interference, as in Section 3.2, but also multipath fading, error correction coding and interleaving. This value will be adopted for the forward link, since coherent reception is easy to implement due to the presence of a *pilot*

signal that is transmitted by each base station. For the reverse link, however, the receiver is usually non-coherent and an E_b / I_0 of 7db (5.01) is required instead.

Note that due to inaccuracy in the power control mechanism, the required E_b / I_0 can be modelled by a log-normally distributed random variable with standard deviation between 1 and 2 dB [11]. In order to simplify the calculations, however, the required E_b / I_0 will be considered constant and equal to 5dB for the forward and 7dB for the reverse link.

4.1 Reverse Link Analysis

Assuming that *K*-1 interfering signals of equal power *S*, as well as a total other-cell interference of power I arrive at a cell site receiver, the E_b / I_0 ratio is

$$E_b / I_0 = \frac{S / R}{(K-1)S / B_{ss} + I / B_{ss} + N_0} = \frac{G_P}{(K-1) + I / S + N_0 B_{ss} / S} \tag{62}$$

where *R* is the bit rate, B_{ss} is the total bandwidth occupied by the system, $G_P = B_{ss} / R$ is the processing gain and N_0 is the (one-sided) power spectral density of the thermal noise. Note that $R = 1/T$, where T is the bit period, and $B_{ss} = 1/T_c$, where T_c is the chip period. Therefore as *K* increases, E_b / I_0 decreases and therefore the probability of error increases. However, the relation (62) is not as simple as the relevant expressions in the previous chapter, since now I is a random variable that also depends on the number of users per cell, *K*. The calculation of the distribution of this random variable will enable us to calculate the reverse link capacity.

Before doing this calculation, we should note that the system performance can be improved through *voice activity detection* and *sectorisation*, as explained in the introduction. In particular, the power of one user's signal at a specific time instant can be expressed as νS instead of *S*, where *v* is the voice activity binary random variable which equals 1 with probability a and 0 with probability $1-a$. Extensive studies have shown that a is between 0.35 and 0.5 [8,11,3]. A value of 3/8=0.375 will be adopted, as in [8]. Furthermore, each cell can be divided in three sectors by means of three directional antennas, each having 120° beamwidth. If we denote the effective number of *users per sector* by $K_s = K/3$, (62) can be modified as

$$E_b / I_0 = G_P \cdot \left(\sum_{k=1}^{K_s - 1} \nu_k + F + n \right)^{-1} \tag{63}$$

where ν_k is the *k*-th user's voice activity random variable, $n = N_0 B_{ss} / S$ and $F = I / S$ is the other-cell interference-to-signal ratio, or the normalised other-cell interference *per sector*. It is clear that with these two techniques the total interference is seriously decreased (by almost a

factor of $3/a=8$) and therefore the total capacity per cell is increased by almost the same factor.

In order to estimate the distribution of the random variable *F*, the propagation in a mobile radio environment has to be considered. The propagation attenuation is usually modelled as the product of the fourth power of distance by a log-normally distributed random variable representing the *shadowing loss* or *long-term fading* [7,chap.1]. Thus for a user at a distance r from a base station, the attenuation is proportional to $r^4 10^{\xi/10}$, where ξ is a Gaussian random variable with zero mean and standard deviation $\sigma=8$. This model of course does not consider the fast *short-term fading* due to multipath, but this generally does not affect the *average* received power level.

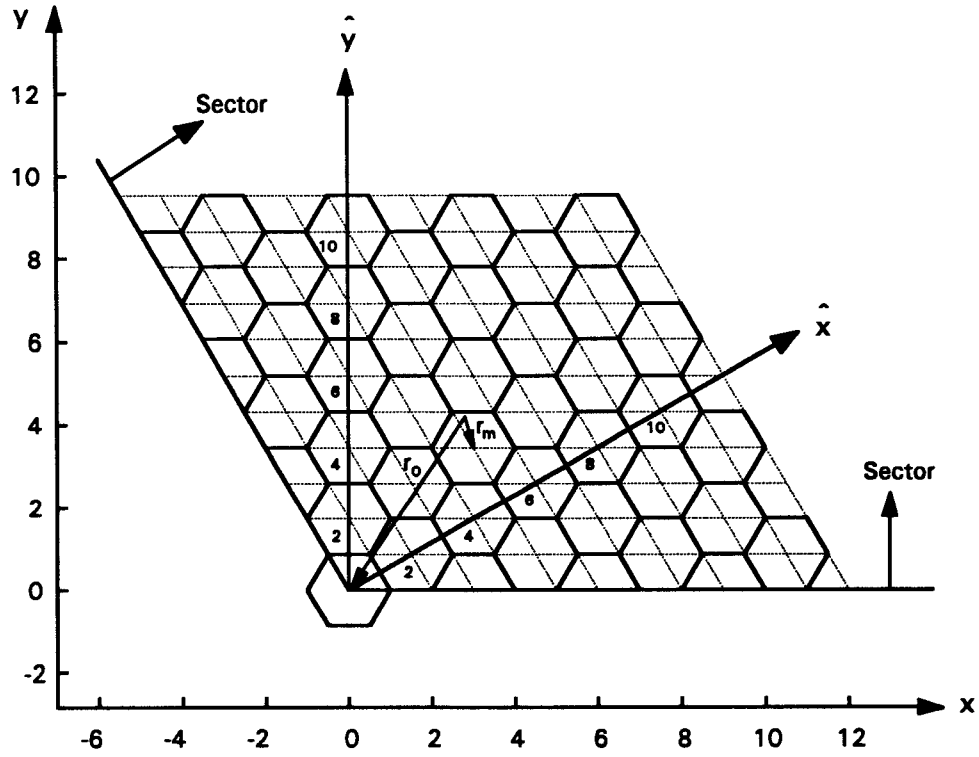

Figure 20. *Hexagonal cell structure used in our cellular environment..*

In Figure 20, the hexagonal cell structure that is used in most cellular environments is illustrated. Based on the above propagation model, and if we suppose that an interfering user j is at a distance r_c from its own cell site and r_0 from the cell site of a desired user, as shown in Figure 20, then due to power control this user produces an interference term F_j in the desired user's cell site, which is equal to

$$F_j = \frac{I_j(r_0, r_c)}{P} = (r_0^{-4} 10^{-\xi_0/10}) / (r_c^{-4} 10^{-\xi_c/10}) = (r_c / r_0)^4 10^{(\xi_c - \xi_0)/10} \quad (64)$$

where ξ_0 and ξ_c are independent zero-mean normal random variables with variance σ^2. Note that since cell membership of a user is determined by minimum attenuation in the path from the user to the nearest cell sites, the term F_j must be always less than unity so that user j does not belong to the same cell as the desired user.

The total normalised interference is then $F = \sum_j F_j$, where the summation is taken over all the users of a sector that do not belong to the cell of the desired user. Since this summation involves an infinitely large number of subscribers randomly placed in the sector, the calculation can become quite complicated. For this reason we assume that users are uniformly placed in space, which is of course an average situation. In Section 4.1.1, a continuous, uniform density of users is assumed, so that the above sum is substituted by an integral over the whole sector area of Figure 20 and an analytical solution is found for *F*, based on [8]. In Section 4.1.2, the sum is calculated with Monte-Carlo simulation, assuming again a uniform distribution of discrete users on a rectangular grid. Finally, in Section 4.1.3 the reverse link capacity is estimated based on equation (63).

4.1.1 Other-Cell Interference: Analytical Solution

Assuming a continuous, uniform density of users over the whole sector area, the above sum is substituted by an integral and the total normalised other-cell interference *F* can be expressed as

$$F = \iint_A v(r_c / r_0)^4 10^{\chi/10} L(r_c / r_0, \chi) \rho \, dx dy \tag{65}$$

where *A* is the two-dimensional sector area, $\chi = \xi_c - \xi_0$ is a zero-mean, normal random variable with variance $\sigma_1^2 = 2\sigma^2$, and $\rho = 2K/(3\sqrt{3}) = 2K_s/\sqrt{3}$ is the user density if we normalise the hexagonal cell radius to unity, as in Figure 20, where the hexagonal cell structure is also illustrated. The function $L(r_c / r_0, \chi)$ is given by

$$L(r_c / r_0, \chi) = \begin{cases} 1, & if \ (r_c / r_0) 10^{\chi/10} \le 1 \\ 0, & \text{otherwise} \end{cases} \tag{66}$$

and is used to limit the integration outside the cell of the desired user (as we have previously noted that F_j must be always less than unity). In [8] it is shown that in this case *F* is a Gaussian random variable with mean

$$m_F = E\{F\} = a \iint_A f_m(r_c / r_0) \rho \, dx dy \tag{67}$$

where

$$f_m(r_c/r_0) = (r_c/r_0)^4 f(r_c/r_0) \tag{68}$$

$$f(r_c/r_0) = e^{(\beta\sigma)^2} T\{40\log_{10}(r_c/r_0)/\sigma_1 + \beta\sigma_1\} \tag{69}$$

where $\beta = (\ln 10)/10$, and $T\{x\}$ is the Gaussian tail function defined by (31). Also, *F* has variance

$$\sigma_F^2 = \mathrm{Var}\{F\} = a\iint_A f_v(r_c/r_0)\rho\, dxdy \tag{70}$$

where

$$f_v(r_c/r_0) = (r_c/r_0)^8 [g(r_c/r_0) - af^2(r_c/r_0)] \tag{71}$$

$$g(r_c/r_0) = e^{(2\beta\sigma)^2} T\{40\log_{10}(r_c/r_0)/\sigma_1 + 2\beta\sigma_1\} \tag{72}$$

Therefore the above equations provide an analytical solution for the probability distribution of *F*. This distribution is completely determined if we calculate the two deterministic integrals given by (67) and (70). However, the integrals can only be numerically computed because of their complexity. Also, three main difficulties are involved in this calculation.

1. The calculation becomes quite complicated if the interfering user base station is chosen according to minimum attenuation, as is usually done in practice, since in this case a very large number of base stations should be considered and only the one for which the attenuation in the path to a specific user is minimum should be selected as this user's base station. For this reason, we made the simplifying assumption that the base station is chosen according to minimum distance and not to minimum attenuation. This assumption is also made in [8], while in [27] a more general case is studied, where the base station is chosen as the one with minimum attenuation among a set of N_c nearest stations.

2. Even with the above assumption, the distances r_0 and r_c appearing in (67) and (70) have to be calculated as functions of the co-ordinates x and y, and for this reason the co-ordinates x_c and y_c of the nearest cell station have to be found. Since this task is not trivial, a pair of *transformed, non-orthogonal co-ordinates* $\hat{x}$ and $\hat{y}$ was employed and the whole integrations were performed using these co-ordinates. The co-ordinates $\hat{x}$ and $\hat{y}$ are defined as

   $$\hat{x} = x + cy \tag{73}$$

   $$\hat{y} = 2cy \tag{74}$$

   where $c = 1/\sqrt{3}$, and are illustrated in Figure 20. If we know the transformed co-ordinates $\hat{x}$ and $\hat{y}$ of a specific user then it can be easily verified in Figure 20 that the transformed co-ordinates $\hat{x}_c$ and $\hat{y}_c$ of the nearest base station can be determined by the following very simple algorithm:

   - Find $m = [\hat{x}] + [\hat{y}] \pmod 3$, where [x] is the integer part of *x*.
   - If $m = 0$, then $\hat{x}_c = [\hat{x}]$ and $\hat{y}_c = [\hat{y}]$.
   - If $m = 1$, then $\hat{x}_c = [\hat{x}] + 1$ and $\hat{y}_c = [\hat{y}] + 1$.

- Finally, if $m = 2$, then:
  - if $\hat{x} - [\hat{x}] > \hat{y} - [\hat{y}]$, then $\hat{x}_c = [\hat{x}] + 1$ and $\hat{y}_c = [\hat{y}]$.
  - if $\hat{x} - [\hat{x}] < \hat{y} - [\hat{y}]$, then $\hat{x}_c = [\hat{x}]$ and $\hat{y}_c = [\hat{y}] + 1$.

The distances r_0 and r_c can then be found by using the fact that the actual Euclidean distance between two points A, B with transformed co-ordinates $(\hat{x}_A, \hat{y}_A)$ and $(\hat{x}_B, \hat{y}_B)$ is given by

$$d_{A,B}^2 = (\hat{x}_A - \hat{x}_B)^2 + (\hat{y}_A - \hat{y}_B)^2 - (\hat{x}_A - \hat{x}_B) \cdot (\hat{y}_A - \hat{y}_B) \tag{75}$$

3. Finally, we can observe that the integrands of (67) and (70) are singular not only at point (0,0) where $r_0 = 0$, but also at all locations of base stations, where $r_c = 0$. This problem of course does not arise in practice and is due to the fact that the propagation model that we used is valid for long distances only. Instead of using a more accurate (and of course more complicated) propagation model, we limited r_0 and r_c so that they are never below a small value 1/Q (Q=1000 was used for the integrations) in order to overcome this problem.

Using the transformed co-ordinates $\hat{x}$ and $\hat{y}$, it can also be shown that the integration area *A* becomes rectangular, therefore the integration limits become very simple. Indeed, (67) can be expressed as

$$m_F = E\{F\} = a\rho \int_0^\infty \int_0^\infty f_m(r_c / r_0) J^{-1} \, d\hat{x} d\hat{y} \tag{76}$$

where *J* is the Jacobian of the transformation

$$J = \begin{vmatrix} 1 & c \\ 0 & 2c \end{vmatrix} = 2c = 2/\sqrt{3} \tag{77}$$

Since $\rho = 2K_s / \sqrt{3}$, the integral becomes

$$m_F = E\{F\} = aK_s \int_0^{y_m} \int_0^{x_m} f_m(r_c / r_0) \, d\hat{x} d\hat{y} \tag{78}$$

where we have also assumed that the upper limits of integration x_m and y_m are finite, so that numerical computation can be applied. In Fig. 21, $f_m(r_c / r_0)$ is plotted as a function of x and y. This figure actually represents the *mean* interference at the central base station receiver at the origin (0,0) caused by a user at location (x,y). The hexagonal cell structure is clearly distinguished, while it is obvious that, due to power control, users far from their base stations (near cell boundaries) have to transmit at greater power levels and therefore cause greater interference to the desired cell site receiver at the origin. It is also obvious that $f_m(r_c / r_0)$ tends to zero for large x and y, therefore the assumption that x_m and y_m can be finite is justified.

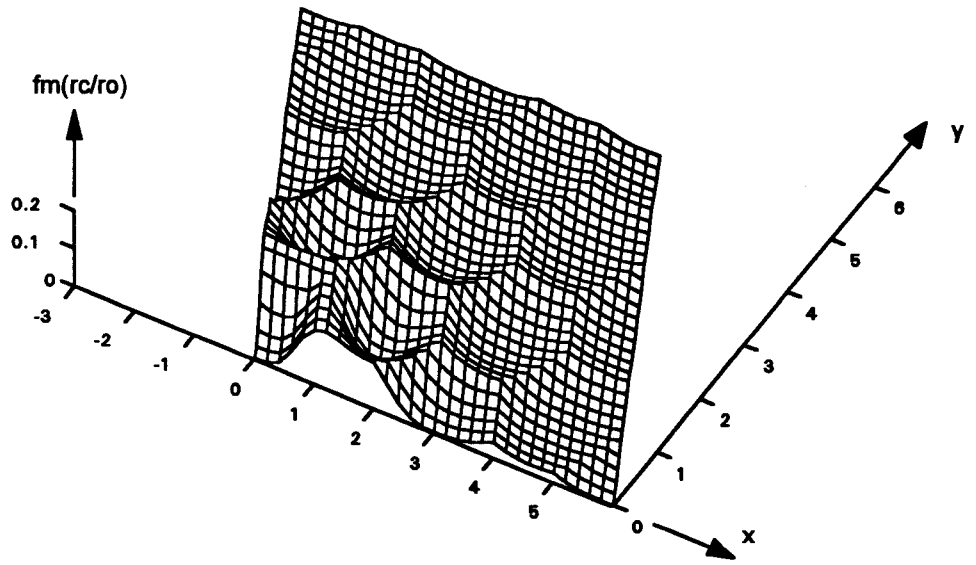

Figure 21. *Mean interference at the origin as a function of x and y.*

Similarly, the variance integral becomes

$$\sigma_F^2 = \mathrm{Var}\{F\} = aK_s \int_0^{y_m} \int_0^{x_m} f_v(r_c / r_0)\, d\hat{x}\, d\hat{y} \tag{79}$$

The upper limits x_m and y_m were selected according to the desired accuracy of the computation. For example, for an accuracy of 10^{-6} it was found that $x_m = y_m = 100$ is enough for the mean and $x_m = y_m = 50$ is enough for the variance. Under all the above assumptions, (78) and (79) were numerically computed using the MATLAB function *quad8*, which applies an adaptive recursive Newton Cotes 8 panel rule [28]. With an accuracy of 10^{-6}, the following results were found:

$$m_F = E\{F\} = 0.425628K_s \tag{80}$$

$$\sigma_F^2 = \mathrm{Var}\{F\} = 0.132347K_s \tag{81}$$

The results are quite different from those of [8], where it is found that $m_F = 0.247K_s$ and $\sigma_F^2 = 0.078K_s$. However, the same results (80) and (81) are approximately found with a different method in the following section, verifying the accuracy of both results. Moreover, it can be found from the results of [27] that m_F is indeed much larger than the one found in [8], in agreement with the above results.

4.1.2 Monte-Carlo Simulation

Using a uniform distribution of users on a rectangular grid (using the transformed co-ordinates $\hat{x}$ and $\hat{y}$), the probability density function of the total normalised other-cell interference *F* was

evaluated using Monte-Carlo simulation. In this case *F* was calculated as a sum instead of an integral:

$$F = \sum_{k_x=0}^{k_m} \sum_{k_y=0}^{k_m} v(r_c / r_0)^4 10^{\chi/10} L(r_c / r_0, \chi) \tag{82}$$

where r_c and r_0 are the distances that correspond to the transformed co-ordinates $\hat{x} = k_x d$ and $\hat{y} = k_y d$, while the upper bound k_m is such that $k_m d = x_m = y_m = 20$. The step d was chosen as $d = 0.1$, so that there are $(1/d) \times (1/d) = 10 \times 10 = 100$ users in each cell sector (since the cell radius is normalised to 1, as in Figure 20). Note that r_c and r_0 are calculated from $\hat{x}$ and $\hat{y}$ in exactly the same way as in the previous section, and in order to avoid singularities, r_c and r_0 are limited again, as described previously. Note also that (82) is a discrete form of (65), which enables us to use real random values for the binary random variable *v* and the normal random variable χ and therefore to obtain a large number of samples of the random variable *F*. In fact, approximately 40000 samples were calculated, from which a histogram was constructed using 40 bins, shown in Figure 22. It is obvious that this histogram indeed approximates a Gaussian probability density function. A real Gaussian pdf was obtained using the mean and variance of the samples, and is also drawn in Figure 22 with a continuous curve.

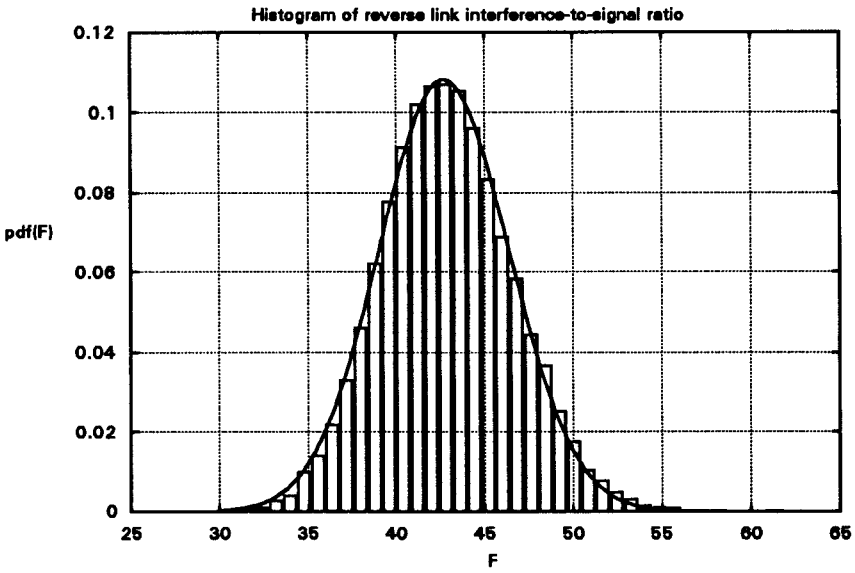

Figure 22. *Histogram of the other-cell interference-to-signal ratio F.*

The mean and variance were obtained from the samples as $m_F = 42.7112$ and $\sigma_F^2 = 13.5816$. Since these values correspond to $K_s = 100$ users per sector, and since it is shown in [8] that m_F and σ_F^2 are proportional to K_s, it can be deduced that in general

$$m_F = E\{F\} \cong 0.427112\, K_s \tag{83}$$

$$\sigma_F^2 = \mathrm{Var}\{F\} \cong 0.135816\, K_s \tag{84}$$

The assumption that m_F and σ_F^2 are proportional to K_s is justified since *F* is in fact the sum of K_s independent random variables. Moreover, it was verified by repeating the same simulation for $K_s = 5 \times 5 = 25$ users per sector. However, the calculation cannot be repeated for every possible value of K_s, as the computation time required is very large. For this reason, the approximate relations (83) and (84) can be used.

Comparing (83) with (80) and (84) with (81) we can see that the two different methods of computation have given almost the same results. Thus, the validity of both is verified. However, only (80) and (81) will be used in the following section for the evaluation of the reverse link capacity, since they are more accurate.

4.1.3 Reverse Link Capacity

Based on the analysis of the previous sections, the reverse link capacity can be evaluated using equation (63). Since the E_b / I_0 ratio has to be greater than $(E_b / I_0)_{\min} = 7\,\text{dB}\,(5.01)$ in order for the bit error rate to be less than 10^{-3}, the probability that $BER > 10^{-3}$ is

$$P = P\{BER > 10^{-3}\} = P\left\{ G_P \left(\sum_{k=1}^{K_s - 1} v_k + F + n \right)^{-1} < (E_b / I_0)_{\min} \right\} \tag{85}$$

Since the random term in (85) is actually the sum of the binomial random variable $\sum_{k=1}^{K_s-1} v_i$ and the Gaussian random variable *F* with mean and variance given by (80) and (81), *P* can be calculated in terms of these two distributions and the result is [8]

$$P = \sum_{k=0}^{K_s - 1} \binom{K_s - 1}{k} a^k (1-a)^{K_s - 1 - k} T\left\{ \frac{\delta - k - \tilde{m}_F K_s}{\sqrt{\tilde{\sigma}_F^2 K_s}} \right\} \tag{86}$$

where $\tilde{m}_F = 0.425628$, $\tilde{\sigma}_F^2 = 0.132347$ and

$$\delta = G_P / (E_b / I_0)_{\min} - n \tag{87}$$

With voice activity factor $a = 3/8$, total bandwidth $B_{ss} = 1.25MHz$, bit rate $R = 8\,\text{kbit/sec}$ and noise-to-signal ratio $n = 1\text{dB}$ (as in [8]), so that $G_P = 156.25$ and $\delta \cong 29.917$, the probability *P* that the required performance is not achieved is given in Figure 23 as a function of K_s, the number of users per sector. Four curves are shown, corresponding to variable load levels for the surrounding cells. For example, the leftmost curve corresponds to fully loaded cells, while the rightmost to a single cell without any other-cell interference. Obviously, the more the surrounding cell load, the worse the system performance.

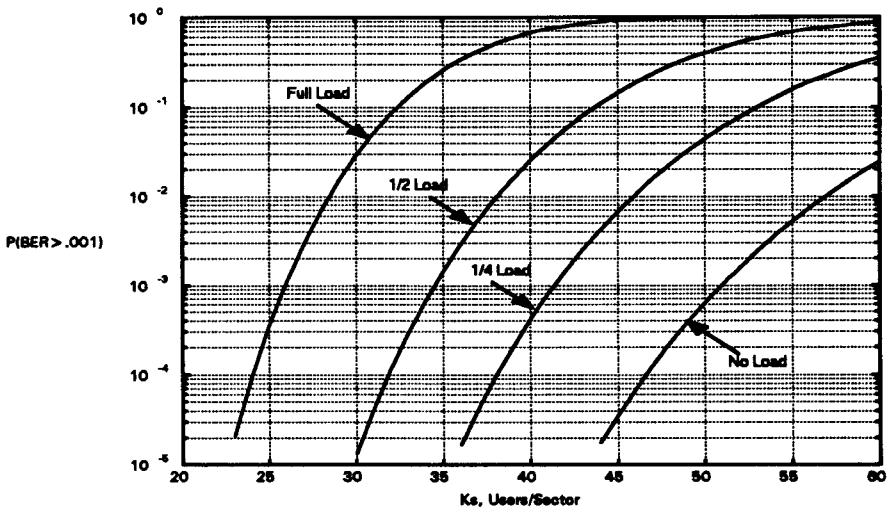

Figure 23. *Illustration of reverse link capacity per sector.*

The soft capacity limit feature of CDMA is obvious from this figure, since it is shown that more subscribers can be accommodated in the system at the expense of slightly inferior performance. For example, if the desired probability is $P<0.01$, up to 28 users/sector can be accommodated, but the capacity can be increased to 30 users/sector if the performance restriction is relaxed to $P<0.03$.

The reverse link capacity of 28 users/sector obtained from the above analysis is lower compared to the capacity of 36 users/sector found in [8]. This discrepancy is due to the fact that a different distribution was obtained for the other-cell interference-to-signal ratio F. This result, however, is more accurate since it was obtained through two different methods and it agrees with the results of [27]. Finally, for a cell of three sectors, the reverse link capacity is $K = 3K_s = 84$ users/cell.

4.2 Forward Link Analysis

As explained in the beginning of this chapter, for the forward link power control takes the form of power allocation at the base station transmitter according to the needs of individual users in a given cell. For this reason, each subscriber measures the received power level of signals from all base stations (actually the *pilot signal* powers are measured), and all measurements are transmitted to the selected base station (the one with the largest power), where a decision is made on the required transmitter power from the base station to the subscriber. Assuming that all but the *L* nearest base stations are negligible, since the power received from them is very small, we can rank the *L* received powers as

$$S_1 > S_2 > \cdots > S_L > 0 \tag{88}$$

where S_1 is the largest received power and therefore corresponds to the selected base station. In this case the received E_b / I_0 for the *i*-th user can be expressed as [8]

$$(E_b / I_0)_i = \frac{\gamma C_i S_1^{(i)} G_P}{\sum_{j=1}^{L} S_j^{(i)} + N_0 B_{ss}} \tag{89}$$

where γ is the fraction of the total base station power devoted to users (usually $\gamma = 0.8$ for all base stations), $1-\gamma$ is the fraction devoted to the pilot signal, and C_i is the fraction of the selected base station power devoted to the *i*-th user. The superscripts (i) refer to the received powers at the *i*-th user. Thus, in order for the *i*-th user to operate with acceptable performance ($BER > 10^{-3}$), C_i must be

$$C_i > \frac{(E_b / I_0)_{\min}}{\gamma G_P}\left[1 + \sum_{j=2}^{L} S_j^{(i)} / S_1^{(i)} + n_i\right] \tag{90}$$

where $(E_b / I_0)_{\min} = 5\,\text{dB}\,(3.16)$ and $n_i = N_0 B_{ss} / S_1^{(i)}$. However, the total power available for subscribers at the base station of the *i*-th user is γS_1, therefore for K_s users/sector we have

$$\sum_{i=1}^{K_s} C_i \le 1 \tag{91}$$

Thus, if we define the factor

$$g_i = 1 + \sum_{j=2}^{L} S_j^{(i)} / S_1^{(i)} \tag{92}$$

as the ratio of the sum of all received powers at the *i*-th user to the largest one, it follows from (90) and (91) that

$$G = \sum_{i=1}^{K_s} g_i \le \frac{\gamma G_P}{(E_b / I_0)_{\min}} - \sum_{i=1}^{K_s} n_i = \delta' \tag{93}$$

Ignoring the terms n_i, since they are very small, and with $(E_b / I_0)_{\min} = 5\,\text{dB}$ and $G_P = 156.25$ as before, we find that $\delta' \cong 39.528$. As explained in [8], the desired performance of

$BER < 10^{-3}$ cannot be achieved for all users within a sector if inequality (93) is not satisfied. Thus,

$$P = P\{BER > 10^{-3}\} = P\{G > \delta'\} \tag{94}$$

and the capacity again depends on the distribution of a random variable, *G*. However, an analytical solution cannot be found as for the reverse link. Thus, the forward link capacity is calculated by two different numerical methods. First a Chernoff bound is found in Section 4.2.1, and then a Gaussian approximation in Section 4.2.2. Finally, the forward link capacity is estimated in Section 4.2.3.

4.2.1 Chernoff Upper Bound

Because *G* is the sum of the K_s independent random variables g_i, the distribution of g_i can be first obtained and from this a Chernoff upper bound of *P* can be found. This method of course only provides a bound that overestimates the actual probability *P*, but the computation involved is rather simple. The distribution of g_i can be obtained through Monte-Carlo simulation. In particular, we used a set of points placed on a rectangular grid inside a sector area defined by $0 \le \hat{x} < 1$, $0 \le \hat{y} < 1$. This is shown as the shaded region in the middle of Figure 24. As we can see from this figure, all but the *L*=19 nearest cells are assumed to contribute negligible receiver power, so only two rings of cells around the central cell are considered for the calculation.

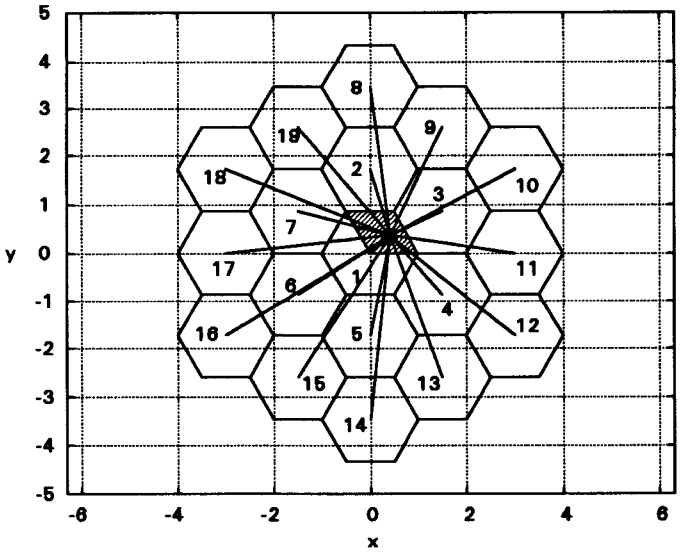

Figure 24. *Geometry for calculation of the forward link capacity.*

For each of the $20 \times 20 = 400$ points on the grid, the attenuation in the path to all 19 cell centres was calculated based on the previous propagation model as $r_k^4 10^{\xi_k/10}$, $k = 1, 2, \ldots, 19$, where r_k is the distance from a specific point to the *k*-th cell centre and ξ_k the respective Gaussian

random variable representing the shadowing loss. Assuming that all base stations transmit at equal powers, the received powers at each point are proportional to $r_k^{-4}10^{-\xi_k/10}$. Thus $g_i - 1$ was estimated by ranking all these 19 random variables $r_k^{-4}10^{-\xi_k/10}$ and finding the ratio of the sum of the 18 smallest to the maximum. Using the transformed co-ordinates $\hat{x}$ and $\hat{y}$, the experiment was repeated 20000 times for each of the 400 uniformly distributed points in the shaded area of Figure 24 and a histogram was obtained from the total of 8×10^6 samples. Figure 25 shows the estimated probability density function of the random variable $g_i - 1$. The result is in complete agreement with [8], where a very similar experiment was performed.

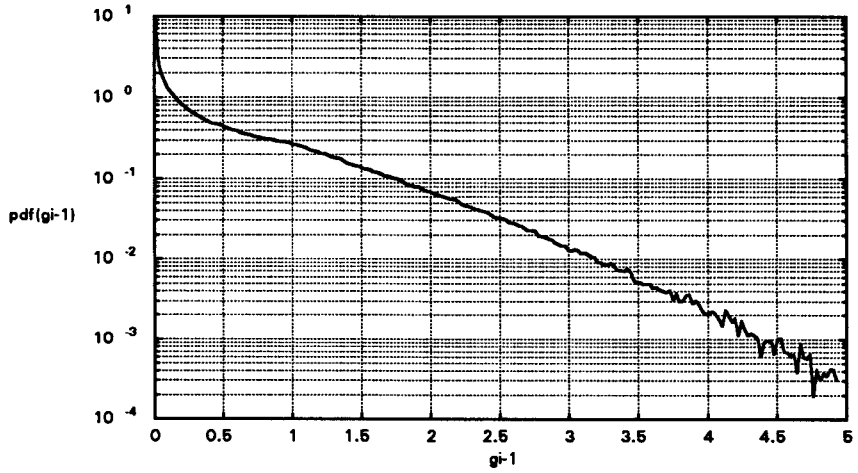

Figure 25. *Estimated probability density function of the random variable* $g_i - 1$.

From the pdf of $g_i - 1$, an upper bound on probability *P* can be obtain as a *Chernoff bound* [29]:

$$P = P\{G - \delta' > 0\} \le \min_{s>0} E\{e^{s(G-\delta')}\} = \min\left\{e^{-s\delta'} E\{\exp[s\textstyle\sum_{i=1}^{K_s} g_i]\}\right\}$$

But since all g_i are independent,

$$P \le \min_{s>0}\left\{e^{-s\delta'} \textstyle\prod_{i=1}^{K_s} E\{e^{sg_i}\}\right\} = \min_{s>0}\left\{e^{-s\delta'} [E\{e^{sg_i}\}]^{K_s}\right\}$$

From the histogram of g_i, $E\{e^{sg_i}\}$ can be estimated as $\sum_k p_k e^{su_k}$, where p_k is the probability that g_i falls in the *k*-th interval $[u_k, u_{k+1}]$ and the summation is over all histogram bins. Thus,

$$P \le \min_{s>0}\left\{e^{-s\delta'} [\textstyle\sum_k p_k e^{su_k}]^{K_s}\right\} \equiv \min_{s>0} f_c(s) \qquad (95)$$

The function $f_c(s)$ is shown in Figure 26 for different values of K_s from to 16 to 24. From this figure it obvious that the function has a local minimum at a specific value of s, and this minimum depends on K_s. Therefore we can obtain an upper bound on the probability *P* by expressing this minimum as a function of K_s. The result will be presented in Section 4.2.3 together with the result of the Gaussian approximation of the next section.

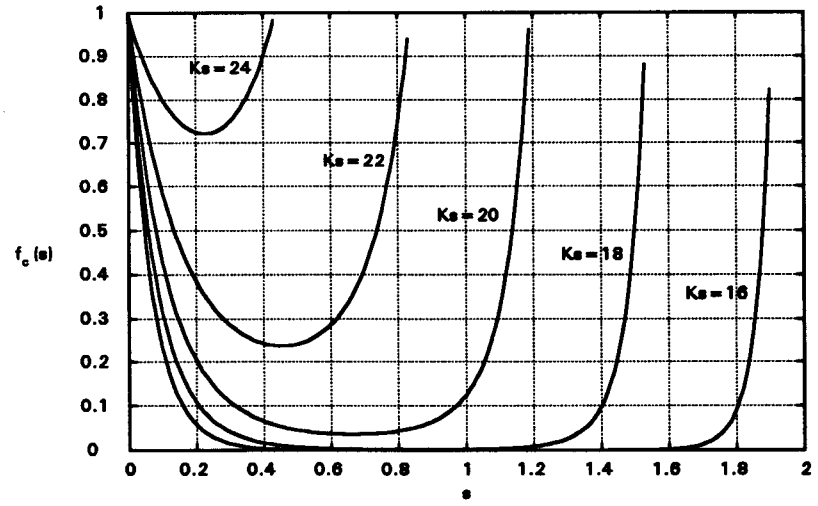

Figure 26. *The function* $f_c(s)$ for various values of K_s.

4.2.2 Gaussian Approximation

Instead of estimating the pdf of g_i and then calculating probability *P* based on a Chernoff bound, the pdf of the sum $G = \sum_{i=1}^{K_s} g_i$ can be directly estimated through a very similar Monte-Carlo simulation. Thus it can be shown that, for a large enough number of users, this pdf is approximately Gaussian, and the probability *P* can be simply expressed in terms of a tail function. This simulation, however, needs more computation time, since for a single sample of *G* a lot of samples of g_i are required.

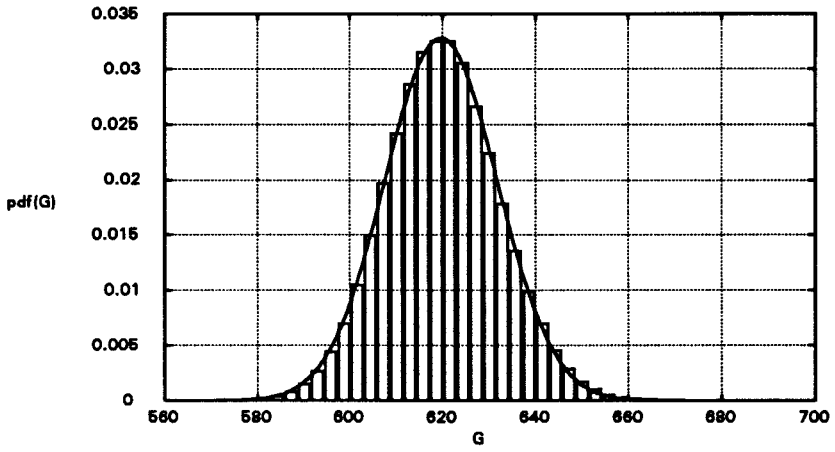

Figure 27. *Histogram of the random variable G, and its Gaussian approximation.*

Using the same rectangular grid (in the transformed co-ordinates $\hat{x}$ and $\hat{y}$) of $20 \times 20 = 400$ uniformly spaced points, i.e. for $\hat{x} = 0, d, 2d, \ldots, 1-d$ and $\hat{y} = 0, d, 2d, \ldots, 1-d$, where the step $d = 0.05$, 400 values were calculated for g_i as in the previous experiment and *G* was estimated as their sum. This experiment, therefore, corresponds to a uniform distribution of

400 (discrete) users per sector. A total of 3×10^5 samples of the random variable *G* were thus obtained, and a histogram of 40 bins was constructed, shown in Figure 27. It is obvious that this histogram approximates a Gaussian distribution. From the mean $m_G = E\{G\} \cong 619.67$ and variance $\sigma_G^2 = \mathrm{Var}\{G\} \cong 147.41$ obtained from the samples, a Gaussian probability density function was constructed, shown superimposed in Figure 27 as a continuous curve.

Since *G* is the sum of K_s independent random variables, it can be shown that, exactly as for the random variable *F* involved in the reverse link analysis, the mean and variance of *G* are proportional to K_s. Thus, since the results of the previous experiment correspond to $K_s = 400$ users/sector,

$$m_G = E\{G\} \cong 1.54917\, K_s \tag{96}$$

$$\sigma_G^2 = \mathrm{Var}\{G\} \cong 0.36853\, K_s \tag{97}$$

for any value of K_s. This was also verified by repeating the same experiment for $10 \times 10 = 100$ users/sector. Therefore, by approximating *G* with a Gaussian random variable of mean and variance given by (96) and (97) respectively, the probability *P* of not achieving adequate performance can be directly calculated as shown in the next section.

4.2.3 Forward Link Capacity

From equation (94) and the Gaussian approximation obtained in the previous section, the probability *P* can be expressed as

$$P = P\{BER > 10^{-3}\} = P\{G > \delta'\} = T\left\{\frac{\delta' - \tilde{m}_G K_s}{\sqrt{\tilde{\sigma}_G^2 K_s}}\right\} \tag{98}$$

where $\tilde{m}_G = 1.54917$, $\tilde{\sigma}_G^2 = 0.36853$ and $\delta' \cong 39.528$. In Figure 28, the probability *P* that is obtained by the Chernoff Bound as well as the Gaussian approximation is shown as a function of the number of users, K_s.

The soft capacity limit feature of CDMA is also obvious from this figure, exactly as for the reverse link, while it is shown that the Chernoff bound approach indeed overestimates the probability *P*. In order for *P* to remain below 10^{-2}, it can be seen that the total number of users/sector cannot exceed 22. It is important to remark that this forward link capacity is smaller than the respective reverse link capacity of 28 users/sector found in section 4.1.3. Therefore the CDMA capacity is limited by the forward link and not by the reverse link. This result is different from the result of [8] (where a forward link capacity of 38 users/sector is found), but is in

agreement with [7,chap.9]. Finally, for a cell of three sectors, the forward link and therefore the total capacity is $K = 3K_s = 66$ users/cell.

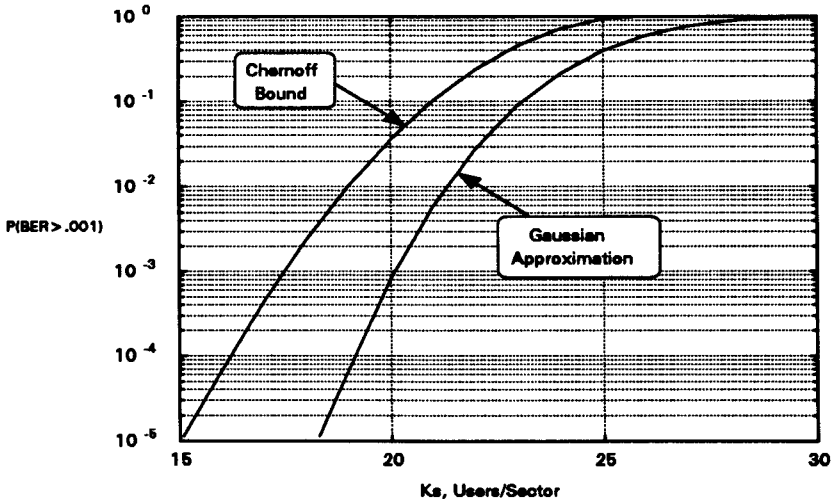

Figure 28. *Illustration of the forward link capacity per sector.*

5. COMPARISON OF CDMA WITH CONVENTIONAL CELLULAR SYSTEMS

CDMA capacity at full bandwidth. For the capacity of 66 users/cell that was calculated in chapter 4, a total channel bandwidth of $B_{ss} = 1.25MHz$ was assumed. According to FCC regulations since 1980 [30], a total bandwidth of 25MHz for the forward and 25MHz for the reverse link has been allocated for the provision of cellular services. The FCC has divided this allocation equally between two service providers, the Block A and Block B operators, as shown in Figure 29. Therefore a total bandwidth of 12.5MHz is available for each link of each service provider. This bandwidth, however, is not contiguous; it is sub-divided in two sub-bands. The minimum available contiguous bandwidth is thus 1.5MHz, as shown in Figure 29.

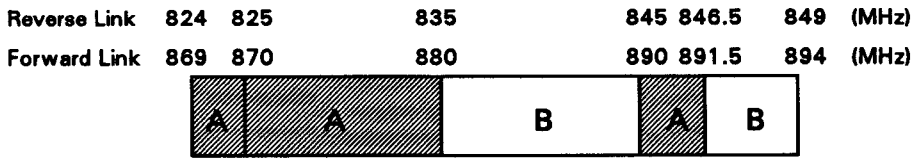

Figure 29. *Cellular frequency allocation.*

For this reason, it is suggested in [9] that a set of ten 1.25MHz bandwidth CDMA channels can be used in order to exploit the full bandwidth of 12.5MHz. In this case, the total capacity of a cellular CDMA system can be approximately 660 users/cell.

Capacity of analog FM/FDMA cellular systems. As explained analytically in [31], a frequency reuse scheme is usually applied in order to increase capacity in FDMA systems, which also creates co-channel interference. Although this interference is much smaller than in CDMA, since the same frequency is not reused in every cell as in CDMA, such systems do not have the interference rejection capabilities of CDMA, and therefore it is usually this co-channel interference that limits their capacity. In order to reduce co-channel interference, the distance between co-channel cells (i.e. cells within which the same set of frequency channels is used) has to be large compared to the cell size. In other words, denoting the cell radius by *R* and the distance between two adjacent co-channel cells by *D*, the *co-channel interference reduction factor q=D/R* has to be large.

In order to maintain adequate system performance (i.e. voice quality), it has been found by subjective intelligibility tests [31] that the carrier-to-interference ratio *C/I* at the receiver has to

be not below $(C/I)_{min} = 18\,\text{dB}$. Considering a hexagonal cell structure, the interference in a worse case comes from six interferers placed at equal distances from the desired receiver, yielding a *C/I* ratio

$$C/I = C/\left(\sum_{k=1}^{6} I_k + n\right) \tag{99}$$

where C is the received carrier power in a desired cell, *n* is the thermal noise and I_k is the interference from the *k*-th interfering cell. This approach assumes that more distant co-channel cells contribute negligible interference and holds for the forward and the reverse link as well [31]. Ignoring the thermal noise, and assuming fourth-order propagation law yields

$$C/I = R^{-4}/6D^{-4} = q^4/6 > (C/I)_{min} \Rightarrow q > [6(C/I)_{min}]^{1/4} \cong 4.411 \tag{100}$$

for $(C/I)_{min} = 18\,\text{dB}$. Moreover, the reduction factor *q* is equal to

$$q = \sqrt{3M} \tag{101}$$

where *M* is the number of cells in frequency reuse pattern [7,chap.5]. Therefore

$$M = q^2/3 > 6.48 \tag{102}$$

and for this reason a frequency reuse pattern of 7 cells is most often used. Effectively, therefore, the total allocated bandwidth is divided in seven sub-bands, each of which is used in one cell only in a certain pattern. With a total bandwidth $B_T = 12.5MHz$ and a required channel bandwidth $B = 30KHz$ for analog FM, up to $B_T/B = 416$ channels per cell pattern can be accommodated, or $416/7 \cong 60$ channels per cell. Note that using 3KHz single-sideband (AM-SSB) channels does not increase capacity, since in this case $(C/I)_{min}$ has to be increased to 38dB. Thus, an analog FDMA system capacity cannot be more than 60 users/cell and it is evident that the CDMA capacity is 11 times the capacity of an analog FDMA system.

Capacity of digital FDMA/TDMA systems. Digital cellular systems are less susceptible to interference, since voice signals are treated digitally and can be protected by error correction coding. Consequently, for the same voice quality, the required $(C/I)_{min}$ is usually lower than that in analog systems. In [31] it is suggested that $(C/I)_{min}$ should be about 10 to 12 dB for 30KHz channel bandwidth and 16 to 18 dB for 10KHz bandwidth. Thus, keeping $(C/I)_{min} = 18\,\text{dB}$, three digital FDMA or TDMA channels occupy the same bandwith as an analog channel and therefore the capacity of such systems is 3 times the capacity of analog FDMA systems, or 180 users/cell. Still, the CDMA capacity is almost 4 times higher.

6. CONCLUSIONS

Code Division Multiple Access is a multiple access scheme that has been used for many years for satellite communications and quite recently for digital cellular radio. Although CDMA possesses certain properties that are very important in a cellular environment, such as multipath suppression, it was not until recently [8] that it was recognised that a cellular CDMA can also provide greater capacity, in users/cell, than a conventional FDMA or TDMA system. In fact, extensive field trials and demonstrations of practical cellular CDMA systems have been carried out until today and it is now seen that CDMA can not only solve the short-term capacity concerns of cellular industry, but also provide a next generation of wireless communication services in the United States [9].

The objective of this report was first to examine two different spread spectrum modulation techniques that are commonly used for CDMA purposes and then to actually calculate the capacity of a multiple-cell CDMA system. Several conclusions can be drawn from the above analysis. First of all, it was demonstrated that code sequence design for DS modulation is very important, since the multipath suppression capability depends on the autocorrelation functions of code sequences, the other-user interference depends on the respective cross-correlation functions, and finally the autocorrelation and cross-correlation functions cannot be made very small simultaneously. Gold codes were selected because of their very good correlation properties, and it was indeed verified by the simulations that they behave as pure random codes, which are ideal for CDMA but cannot be used in practice.

From the analysis of the direct sequence BPSK and QPSK systems, it can be seen that both systems perform equally well in a CDMA environment, although the performance of the QPSK system is slightly better. One of the most important parameters in the design of such systems is the processing gain, which should be as high as possible for a low probability of error. The processing gain is however limited by the total available bandwidth for a given bit rate. The bit rate should be made as small as possible by applying efficient voice coding. One other important factor is the code period, which was equal to the processing gain in our simulations, because in this case the other-user interference is minimised. In practice, however, much longer codes might be employed, in spite of their slightly inferior performance, because of the many advantages the have. Finally, the thermal noise does not seem to affect the system performance, since, even with a signal-to-noise ratio of unity, its contribution to the total interference is equivalent to adding just one more user to the system.

From the analysis of the multiple-cell CDMA environment, we can conclude that it is the forward and not the reverse link that limits the system capacity. Note, however, that voice activity detection was not considered in the analysis of the forward link, and with this modification the forward link might have better performance. The total system capacity was found inferior compared with previous results [8], but the results obtained here were shown to be more accurate, since they were verified by two different methods for both links. Still, the CDMA capacity is much higher compared with conventional analog and even digital systems.

The measure of system capacity that was used in the analysis of the multiple-cell CDMA system was the maximum number of users per cell that can be serviced at one time for a given service quality. In practice, however, the measure of economic usefulness of a multi-user communication system is the peak load that can be supported with a given quality, as measured by the *blocking probability*, i.e. the probability that an incoming user will find all channels busy. Considering a random traffic arrival process and a random service time, the *Erlang capacity* can be evaluated, which is the average traffic load that results in a maximum tolerable blocking probability. Such an approach is used in [11] and it would be very useful to apply this approach in our analysis of the forward and reverse links and find a more accurate estimate of the CDMA Erlang capacity.

Some more accurate estimates of the CDMA capacity can also be found, recognising that, due to imperfections in the power control mechanism, the required bit-energy-to-interference-density ratio at the receiver is not constant, but rather a log-normally distributed random variable [11]. Also, cell membership is determined in practice not by minimum distance, but rather by minimum attenuation. Using these two facts the calculations can be made very accurate, although they become rather complicated. The rough estimates obtained in this report, however, already show that the CDMA capacity is much superior to the capacity of other systems.

Finally, the system capacity can be further enhanced by using directive antennas and applying superresolution beamforming techniques, which have been introduced quite recently. This way high resolution capabilities can be provided, resulting in spatial isolation of users. Thus the total other-user interference is reduced and the system capacity is proportionally increased. Concluding, cellular CDMA systems can provide a next generation technology for truly portable communications in the most economic and efficient manner providing for a graceful evolution into the future generations of wireless technologies.

7. REFERENCES

[1] R. Pickholtz, D. Schilling and L. Milstein, "Theory of Spread Spectrum Communications - A Tutorial", *IEEE Trans. Comm.*, Vol. COM-30, May 1982, pp. 855-884.

[2] R. Scholtz, "The Origins of Spread Spectrum Communications", *IEEE Trans. Comm.*, Vol. COM-30, May 1982, pp. 822-854.

[3] R. Pickholtz, L. Milstein and D. Schilling, "Spread Spectrum for Mobile Communications", *IEEE Trans. Veh. Tech.*, Vol. 40, May 1991, pp. 313-322.

[4] W. Lee, "Overview of Cellular CDMA", *IEEE Trans. Veh. Tech.*, Vol. 40, May 1991, pp. 291-302.

[5] A. Viterbi, "When not to Spread Spectrum - A Sequel", *IEEE Com. Mag.*, Vol. 23, April 1985, pp. 12-17.

[6] K. Gilhousen, I. Jacobs, R. Padovani, A. Viterbi and L. Weaver, "Increased Capacity Using CDMA for Mobile Satellite Communications", *IEEE Journal Select. Areas Comm.*, Vol. 8, May 1990, pp. 503-514.

[7] W. Lee, *Mobile Communications Design Fundamentals*, J. Wiley & Sons, 1993.

[8] K. Gilhousen, I. Jacobs, R. Padovani, A. Viterbi, L. Weaver and C. Wheatley, "On the Capacity of a Cellular CDMA System", *IEEE Trans. Veh. Tech.*, Vol. 40, No. 2, May 1991, pp. 303-312.

[9] A. Salmasi, "An Overview of CDMA Applied to the Design of Personal Communications Networks", from *Third Generation Wireless Information Networks*, edited by S. Nanda, D. Goodman, Kluwer Academic Publishers, 1992, pp.277-298.

[10] J. Holmes, *Coherent Spread Spectrum Systems*, J. Wiley & Sons, NY, 1982.

[11] A. M. Viterbi and A. J. Viterbi, "Erlang Capacity of a Power Controlled CDMA System", *IEEE Journal Select. Areas Comm.*, Vol. 11, No. 6, August 1993, pp. 892-899.

[12] D. Sarwate and M. Pursley, "Cross-Correlation Properties of PN and Related Sequences", *Proc. IEEE*, Vol. 68, No. 5, May 1980, pp. 593-619.

[13] M. Pursley, "Performance Evaluation for Phase-Coded Spread Spectrum Multiple Access Communication - Part I: System Analysis", *IEEE Trans. Comm.*, Vol. COM-25, August 1977, pp. 795-799.

[14] L. Welch, "Lower Bounds on the Maximum Cross-Correlation of Signals", *IEEE Trans. Information Theory*, Vol. IT-20, May 1974, pp. 397-399.

[15] M. Pursley, "Performance Evaluation for Phase-Coded Spread Spectrum Multiple Access Communication - Part II: Code Sequence Analysis", *IEEE Trans. Comm.*, Vol. COM-25, August 1977, pp. 800-803.

[16] R. Ziemer and R. Peterson, *Introduction to Digital Communications*, Macmillan, 1992

[17] W. Peterson and E. Weldon, *Error Correcting Codes*, 2nd ed., MIT Press, 1972.

[18] R. Gold, "Optimal Binary Sequences for Spread Spectrum Multiplexing", *IEEE Trans. Information Theory*, Vol. IT-13, October 1967, pp. 619-621.

[19] R. Gold, "Maximal Recursive Sequences with 3-Valued Recursive Cross-Correlation Functions", *IEEE Trans. Information Theory*, January 1968, pp. 154-156.

[20] M. Simon, J. Omura, R. Scholtz and B. Levitt, *Spread Spectrum Communications*, Vol. I, Computer Science Press, 1985.

[21] K. Yao, "Error Probability of Asynchronous Spread Spectrum Multiple Access Communication Systems", *IEEE Trans. Comm.*, Vol. COM-25, August 1977, pp. 803-809.

[22] N. Nazari and R. Ziemer, "Computationally Efficient Bounds for the Performance of Direct Sequence Spread Spectrum Multiple Access Communication Systems in Jamming Environments", *IEEE Trans. Comm.*, Vol. COM-36, May 1988, pp. 577-586.

[23] S. Haykin, *Digital Communications*, J. Wiley & Sons, 1988.

[24] M. Pursley, "Spread Spectrum Multiple Access Communications", from *Multi-User Communication Systems*, edited by G. Longo, Springer-Verlag, Berlin, 1981, pp. 139-199.

[25] K. Karkkainen, M. Laukkanan and H. Tamanen, "Performance of Asynchronous DS-CDMA System with Long and Short Spreading Codes", *Electronics Letters*, Vol. 30, No. 13, 23 June 1994, pp. 1035-1036.

[26] S. Lim, "Analysis of the Performance of Spread Spectrum Systems in CDMA Environment", MEng. Thesis, Imperial College of Science, Technology and Medicine, June 1994.

[27] A. J. Viterbi, A. M. Viterbi and E. Zehavi, "Other-Cell Interference in Cellular Power Controlled CDMA", *IEEE Trans. Comm.*, Vol. 42, No. 2/3/4, February-April 1994, pp. 1501-1504.

[28] G. Forsythe, M. Malcolm, C. Moler, *Computer Methods for Mathematical Computations*, 1977, chap. 5.

[29] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, 3rd ed., McGraw-Hill, 1991, p. 122.

[30] W. Lee, *Mobile Cellular Telecommunications Systems*, McGraw-Hill, 1989.

[31] W. Lee, "Spectrum Efficiency in Cellular", *IEEE Trans. Veh. Tech.*, Vol. 38, No. 2, May 1989, pp. 69-75.

APPENDIX A

Pseudo-Random Sequences Simulation

All the calculations involved in the experiments of this report were implemented using MATLAB. The programs that were used for this purpose, either script files or functions, are included in this as well as the following appendices.

PRIMINIT.M

This file contains primitive polynomials of degree up to 80.
The coefficients are given in octal representation.

```
P2=['7'];

P3=['13'];

P4=['23'];

P5=['45';'75';'67'];

P6=['103';'147';'155'];

P7=['211';'217';'235';'367';'277';'325';'203';'313';'345'];

P8=['435';'551';'747';'453';'545';'537';'703';'543'];

P9=['1021';'1131';'1461';'1423';'1055';'1167';'1541'; ...
    '1333';'1605';'1751';'1743';'1617';'1553';'1157'];

P10=['2011';'2415';'3771';'2157';'3515';'2773';'2033'; ...
     '2443';'2461';'3023';'3543';'2745';'2431';'3177'];

P11=['4005';'4445';'4215';'4055';'6015';'7413';'4143'; ...
     '4563';'4053';'5023';'5623';'4577';'6233';'6673';'7335'];

P12=['10123';'15647';'16533';'16047';'11015';'14127'; ...
     '17673';'13565';'15341';'15053';'15621';'15321'; ...
     '11417';'13505'];

P13=['20033';'23261';'24623';'23517';'30741';'21643'; ...
     '30171';'21277';'27777';'35051';'34723';'34047'; ...
     '32535';'31425';'33343'];

P14=['42103';'43333';'51761';'40503';'77141';'62677'; ...
     '44103';'45145';'76303';'64457';'57231';'64167'; ...
     '60153';'55753';'67517'];

P15=['100003';'102043';'110013';'102067';'104307';'100317'; ...
     '177775';'103451';'110075';'102061';'114725';'103251'; ...
     '100021';'100201'];

P16=['210013';'234313';'233303';'307107';'307527';'306357'; ...
     '201735';'272201';'242413';'270155';'302157';'210205'; ...
     '305667';'236107'];

P17=['400011';'400017';'400431';'525251';'410117';'400731'; ...
     '411335';'444257';'600013';'403555';'525327';'411077'; ...
     '400041';'400101'];
```

```
P18=['1000201';'1000247';'1002241';'1002441';'1100045'; ...
     '1000407';'1003011';'1020121';'1101005';'1000077'; ...
     '1001361';'1001567';'1001727';'1002777'];

P19=['2000047';'2000641';'2001441';'2000107';'2000077'; ...
     '2000157';'2000175';'2000257';'2000677';'2000737'; ...
     '2001557';'2001637';'2005775';'2006677'];

P20=['4000011';'4001051';'4004515';'6000031';'4442235'];

P21=['10000005';'10040205';'10020045';'10040315';'10000635'; ...
     '10103075';'10050335';'10002135';'17000075'];

P22=['20000003';'20001043';'22222223';'25200127';'20401207'; ...
     '20430607';'20070217'];

P23=['40000041';'40404041';'40000063';'40010061';'50000241'; ...
     '40220151';'40006341';'40405463';'40103271';'41224445'; ...
     '40435651'];

P24=['100000207';'125245661';'113763063'];

P25=['200000011';'200000017';'204000051';'200010031'; ...
     '200402017';'252001251';'201014171';'204204057'; ...
     '200005535';'200014731'];

P26=['400000107';'430216473';'402365755';'426225667'; ...
     '510664323';'473167545';'411335571'];

P27=['1000000047';'1001007071';'1020024171';'1102210617'; ...
     '1250025757';'1257242631';'1020560103';'1112225171'; ...
     '1035530241'];

P28=['2000000011';'2104210431';'2000025051';'2020006031'; ...
     '2002502115';'2001601071'];

P29=['4000000005';'4004004005';'4000010205';'4010000045'; ...
     '4400000045';'4002200115';'4001040115';'4004204435'; ...
     '4100060435';'4040003075';'4004064275'];

P30=['10040000007';'10104264207';'10115131333';'11362212703'; ...
     '10343244533'];

P31=['20000000011';'20000000017';'20000020411';'21042104211'; ...
     '20010010017';'20005000251';'20004100071';'20202040217'; ...
     '20000200435';'20060140231';'21042107357'];

P32=['40020000007';'40460216667';'40035532523';'42003247143'; ...
     '4176042760 7'];

P33=['100000020001';'100020024001';'104000420001'; ...
     '100020224401';'111100021111';'100000031463'; ...
     '104020466001';'100502430041';'100601431001'];

P34=['201000000007';'201472024107';'377000007527'; ...
     '225213433257';'227712240037';'251132516577'; ...
     '211636220473';'200000140003'];

P35=['400000000005'];

P36=['1000000004001'];

P37=['2000000012005'];

P38=['4000000000143'];

P39=['10000000000021'];

P40=['20000012000005'];

P61=['200000000000000000047'];

P89=['4000000000000000000000000000151'];

for k=2:89
```

```
  K=num2str(k);
  if exist(['P',K])==1,
    eval(['global P',K])
%   eval(['P',K,'=oct2coef(P',K,');'])
  end
end

clear k K
```

PRIM.M

This function calculates a primitive polynomial's coefficients in binary form.

```
function pol=prim(m,n,combine)

if nargin>1,
  if any(size(m)~=size(n)) | nargin>2,
    [m,n]=meshgrid(m,n);
  end
  m=m(:);
  n=n(:);
  K=length(m);
  pol=zeros(K,max(m));
  for k=1:K
    M=num2str(m(k));
    N=num2str(n(k));
    eval(['global P',M])
    eval(['p=P',M,'(',N,',:);'])
    pol(k,1:m(k))=oct2coef(p);
  end
else
  pol=[];
  for k=1:length(m)
    M=num2str(m(k));
    eval(['global P',M])
    eval(['p=P',M,';'])
    pol=rows(pol,oct2coef(p));
  end
end
```

OCT2COEF.M

Converts polynomial coefficients from octal to binary form.

```
function bin=oct2coef(oct)

[m1,m2]=size(oct);
oct=oct';
oct=str2num(oct(:));
bin=zeros(3*m2,m1);
bin(:)=dec2bin(oct,3)';
bin=fliplr((bin(1:3*m2-1,:))');
bin=[bin; zeros(1,3*m2-1)];
bin=bin(1:m1,1:max(find(sum(bin))));
```

MSEQ.M

Implements a linear feedback shift register to gererate m-sequences.

```
function [seq,fs,per,sm]=mseq(p,n,is)

[m1,m2]=size(p);
if exist('n')~=1, n=2^m2-1; end
if exist('is')~=1, is=[ones(m1,1), zeros(m1,m2-1)]; end
seq=zeros(m1,n);
sm=zeros(m1,m2*n);
per=zeros(m1,1);
s=is;
for k=1:n
  seq(:,k)=s(:,m2);
```

```
  if nargout>3, sm(:,(1:m2)+(k-1)*m2)=s; end
  s=[rem((sum((s.*p)'))',2) s(:,1:m2-1)];
  if nargout>2,
    f=find(all((s==is)') & per'==0);
    per(f)=k*ones(size(f));
  end
end
fs=s;
```

GOLD.M

Implements a combination of two shift registers to generate Gold sequences.

```
function [seq,fs1,fs2,per,sm1,sm2]=gold(p,N,n,is1,is2)

N=min(N,2^(size(p(1,:),2)));
p1=ones(N,1)*p(1,:);
p2=ones(N,1)*p(2,:);
[m1,m2]=size(p1);
if exist('n')~=1, n=2^m2-1; end
if isempty(n), n=2^m2-1; end
if exist('is1')~=1, is1=[]; end
if isstr(is1),
  [is1,is2]=goldinit(p(1,:),m1);
  for i=1:m1,
    phase=1+floor(((2^m2-1)-1)*rand);
    [X,s1]=mseq(p1(i,:),phase,is1(i,:));
    [X,s2]=mseq(p2(i,:),phase,is2(i,:));
    is1(i,:)=s1;
    is2(i,:)=s2;
  end
elseif nargin<5,
  [is1,is2]=goldinit(p(1,:),m1);
end
seq=zeros(m1,n);
if nargout>4,
  sm1=zeros(m1,m2*n);
  sm2=zeros(m1,m2*n);
end
per=zeros(m1,1);
s1=is1;
s2=is2;
for k=1:n
  seq(:,k)=rem(s1(:,m2)+s2(:,m2),2);
  if nargout>4,
    sm1(:,(1:m2)+(k-1)*m2)=s1;
    sm2(:,(1:m2)+(k-1)*m2)=s2;
  end
  s1=[rem((sum((s1.*p1)'))',2) s1(:,1:m2-1)];
  s2=[rem((sum((s2.*p2)'))',2) s2(:,1:m2-1)];
  if nargout>3,
    f=find(all(([s1 s2]==[is1 is2])') & per'==0);
    per(f)=k*ones(size(f));
  end
end
fs1=s1;
fs2=s2;
```

GOLDINIT.M

Calculates initial conditions of shift registers for Gold sequences.

```
function [is1,is2]=goldinit(p,N)

L=length(p);
p=[1 p];
g=p;
k=L+1:-2:1;
g(k)=zeros(size(k));
is1=zeros(N,L);
for i=1:L
  is1(:,L-i+1)=ones(N,1)*g(i);
```

```
  g=[rem(g(i)*p+g,2) 0];
  p=[0 p];
end
is2=dec2bin((0:N-1)',L);
```

BGOLD.M

Finds balanced Gold codes.

```
function [seq,fs1,fs2,per,sm1,sm2]=bgold(p,N,n,is)

N=min(N,2^(size(p(1,:),2)-1));
if exist('n')~=1, n=[]; end
if exist('is')~=1, is=[]; end
a=2;
k=[];
while length(k)<N,
  a=a+1;
  if a>3, a, end
  if nargout>4,
    [seq,fs1,fs2,per,sm1,sm2]=gold(p,a*N,n,is);
  else
    [seq,fs1,fs2,per]=gold(p,a*N,n,is);
  end
  k=find(mean(seq')==mean(seq(1,:)));
end
k=k(1:N);
seq=seq(k,:);
fs1=fs1(k,:);
fs2=fs2(k,:);
per=per(k,:);
if nargout>4,
  sm1=sm1(k,:);
  sm2=sm2(k,:);
end
```

FGOLD.M

Finds appropriate (preferred) polynomials for Gold codes and saves results in M-files.

```
function [seq,fs1,fs2,per]=fgold(deg,N,n,is)

quote=setstr(39);
DEG=num2str(deg);
eval(['ready=(exist(',quote,'code',DEG,'.mat',quote,')==2);'])
if ready & ~isstr(deg),
  eval(['load code',DEG])
else
  if isstr(deg), DEG=deg; deg=str2num(deg);
  else DEG=num2str(deg);
  end
  prefpol=[ 5  1  3 ;
            6  1  2 ;
            7  1  5 ;
            9  1  8 ;
           10  1  5 ;
           11  1 15 ;
           13  1 15 ;
           14  1 15 ];
  k=find(prefpol(:,1)==deg);
  p=prim(deg,prefpol(k,[2 3]));
  if exist('N')~=1, N=128; end
  if isempty(N), N=128; end
  if exist('n')~=1, n=[]; end
  if exist('is')~=1, is=[]; end
  [seq,fs1,fs2,per]=bgold(p,N,n,is);
  eval(['save code',DEG,' seq fs1 fs2 per'])
end
```

CORR.M

Finds auto- or cross-correlation function of PN sequences.

```
function [c,olag]=corr(s,lag,M,shift,even)

[m1,m2]=size(s);
if exist('lag')~=1, lag=-(m2+2):(m2+2); end
if isempty(lag), lag=-(m2+2):(m2+2); end
if exist('M')~=1, M=m2; end
if isempty(M), M=m2; end
if exist('shift')~=1, shift=0; end
if isempty(shift), shift=0; end
if exist('even')~=1, even=1; end
if M==m2,
  if even, [c,olag]=evencorr(s,lag);
  else [c,olag]=oddcorr(s,lag);
  end
else
  s=1-2*s;
  x=s(1,:);
  if m1==2, y=s(2,:); else y=x; end
  lag=lag(:);
  c=zeros(1,length(lag));
  M=min(M,m2);
  for m=1:length(lag)
    kx=mod((1:M)-shift-1,m2)+1;
    ky=mod((1:M)+lag(m)-shift-1,m2)+1;
    c(m)=sum(x(kx).*y(ky));
  end
  c=c/M;
  olag=lag';
end
```

EVENCORR.M

Finds even cross-correlation function of PN sequences.

```
function [c,olag]=evencorr(s,lag)

s=1-2*s;
[m1,m2]=size(s);
x=s(1,:);
if m1==2, y=s(2,:); else y=x; end
if exist('lag')~=1, lag=-(m2+2):(m2+2); end
lag=lag(:);
c=[0 xcorr(x,y,'coeff')];
m=length(c)/2;
c=c(1:m)+c(m+1:2*m);
c=c(mod(lag,m2)+1);
olag=lag';
```

ODDCORR.M

Finds odd cross-correlation function of PN sequences.

```
function [c,olag]=oddcorr(s,lag)

s=1-2*s;
[m1,m2]=size(s);
x=s(1,:);
if m1==2, y=s(2,:); else y=x; end
if exist('lag')~=1, lag=-(m2+2):(m2+2); end
lag=lag(:);
c=[0 xcorr(x,y,'coeff')];
m=length(c)/2;
c=c(1:m)-c(m+1:2*m);
c=c(mod(lag,m2)+1);
olag=lag';
```

APPENDIX B

Single-Cell BPSK DS-CDMA System Simulation

BPSKTRAN.M

Implements DS-BPSK transmitter.

```
function [t,d,pn,dp,s]=bpsktran(data,seq,fo,theta,fs,k,PG,P)

[d1,d2]=size(data);
[s1,s2]=size(seq);
To=floor(fs/fo);
Tc=k*To;
t=((1:d2*PG*Tc)-1)/fs;
d=rect(data,PG*Tc);
seq=repeat(seq,ceil(d2*PG/s2));
seq=seq(1:d2*PG);
pn=rect(seq,Tc);
dp=pn.*d;
s=dp.*(sqrt(2*P)*cos(2*pi*fo*t+theta));
```

BPSKREC.M

Implements DS-BPSK receiver.

```
function [data,out,rc,car,rp,pn]= ...
        bpskrec(r,seq,fo,theta,fs,k,PG)

[r1,r2]=size(r);
[s1,s2]=size(seq);
To=floor(fs/fo);
Tc=k*To;
bits=floor(r2/(PG*Tc));
t=((1:r2)-1)/fs;
seq=repeat(seq,ceil(bits*PG/s2));
seq=seq(1:bits*PG);
pn=rect(seq,Tc);
rp=r.*pn;
car=cos(2*pi*fo*t+theta);
rc=rp.*car;
out=mean(reshape(rc',PG*Tc,bits*s1));
out=reshape(out,bits,s1)';
data=out<0;
```

BPSK.M

Implements DS-BPSK CDMA system.

```
function [ber,rec,Sout,Nout]= ...
        bpsk(ib,tran,seq,del,fo,theta,fs,k,PG,P,SNR,fil,graph)

% Main Parameters
if nargin<13, graph=[]; end
[M,bits]=size(tran);
To=floor(fs/fo);
Tc=k*To;
Tbit=PG*Tc;
del(1)=Tbit;
fc=fo/k;
```

```
% Transmiter & Receiver Filter Design
[Bt,At,delt]=txfil(fo,fc,fs,fil);
%[Br,Ar,delr]=rxfil(fc,PG,fs,fil);

% Ideal Output (No Noise or Interference)
[t,d,pn,dp,s1]=bpsktran(tran(1,:),seq(1,:),fo,theta(1),fs,k,PG,P);
sfil1=filterd(Bt,At,s1,0,fil);
rfil1=fliplr(filterd(Bt,At,fliplr(sfil1),0,fil));
[rec1,Sout]=bpskrec(rfil1,seq(1,:),fo,theta(1),fs,k,PG);

% Initialisation
s=zeros(1,bits*Tbit);
ber=zeros(M,1);
rec=zeros(M,bits);
Nout=zeros(M,bits);

% Main Loop
for i=1:M,

  % Transmiter
  [t,d,pn,dp,si]=bpsktran([ib(i) tran(i,:)], ...
       seq(i,:),fo,theta(i),fs,k,PG,P);
  si=si(del(i)+(1:bits*Tbit));
  s_old=s;
  s=s+si;

  % Transmiter Filter
  sfil=filterd(Bt,At,s,0,fil);

  % Thermal Noise
  n=randn(size(s));
  n=n/std(n)*sqrt(P)*10^(-SNR/20);
  r=sfil+n;

  % Receiver Filter
  rfil=fliplr(filterd(Bt,At,fliplr(r),0,fil));

  % Receiver
  [rec(i,:),Nout(i,:),rc,car,rp,pn]= ...
       bpskrec(rfil,seq(1,:),fo,theta(1),fs,k,PG);
  Nout(i,:)=Nout(i,:)-Sout;

  % Bit Error Rate
  ber(i)=sum(sum(rec(i,:)~=tran(1,:)))/bits;
  diary on
  disp([i ber(i)])
  diary off

  % Plots
  if graph,
    figure(1)
      w=(1:Tc*10)+Tc*0;
      subplot(511)
      plot(t(w),s_old(w))
      title('Direct Sequence BPSK CDMA System (Transmitter)')
      ylabel('Old s(t)')
      grid
      subplot(512)
      plot(t(w),si(w))
      ylabel(['s',num2str(i),'(t)'])
      grid
      subplot(513)
      plot(t(w),s(w))
      ylabel('New s(t)')
      grid
      subplot(514)
      plot(t(w),sfil(w))
      ylabel('sfil(t)')
      grid
      subplot(515)
      plot(t(w),r(w))
      xlabel('t (msec)')
      ylabel('r(t)')
      grid
    figure(2)
```

```
      subplot(511)
      plot(t(w),rfil(w))
      title('Direct Sequence BPSK CDMA System (Receiver)')
      ylabel('rfil(t)')
      grid
      subplot(512)
      plot(t(w),pn(w))
      ylabel('pn(t)')
      V=axis;
      V([3 4])=[-1.2 1.2];
      axis(V);
      grid
      subplot(513)
      plot(t(w),rp(w))
      ylabel('rp(t)')
      grid
      subplot(514)
      plot(t(w),car(w))
      ylabel('carrier')
      grid
      subplot(515)
      plot(t(w),rc(w))
      xlabel('t (msec)')
      ylabel('rc(t)')
      grid
    if bits>1,
      figure(3)
      w=1:10:Tbit*min(bits,5);
      plot(t(w),rc(w))
      title('Direct Sequence BPSK CDMA System (Receiver)')
      xlabel('t (msec)')
      ylabel('rc(t)')
      grid
    end
    figure(1)
    pause
  end
end
```

BPSKTEST.M

Tests the DS-BPSK CDMA system and saves simulation results.

```
% Main Parameters
M=64;
N=500;
bits=20;
code=7;
fo=20;
fs=20*fo;
k=2;
P=1/2;
SNR=0;
fil=0;
graph=0;

% Constants
M=min(M,2^(code-1));
eval(['load code',num2str(code)])
PG=length(seq);
To=floor(fs/fo);
Tc=k*To;
Tbit=PG*Tc;
savename='bpsk';
if fil, savename=[savename,'f']; end

% Initialisation
if exist(['r',savename,'.mat'])==2,
  eval(['load r',savename])
else
  K=0;
  SumBER=zeros(M,1);
  SumS=0;
  SumN=zeros(M,1);
```

```
  SumN2=zeros(M,1);
end

eval(['diary o',savename])
diary off

% Main Loop
for i=1:N
  diary on
  disp(['Iteration ',num2str(i)])
  disp(['Total Number of Bits: ',num2str(K)])
  diary off
  K=K+bits;

  % Random Delays & Carrier Phases of Users
  del=floor(Tbit*rand(M,1))+1;
  del(1)=Tbit;
  theta=2*pi*rand(M,1);

  % Random Codes
  [X,C]=sort(rand(size(seq,1),1));
  C=C(1:M);

  % Random Inputs
  ib=round(rand(M,1));
  tran=round(rand(M,bits));

  % BPSK Simulation
  [BERi,rec,Si,Ni]=bpsk(ib,tran,seq(C,:), ...
        del,fo,theta,fs,k,PG,P,SNR,fil,graph);

  % Accumulation of Results
  SumBER=SumBER+BERi*bits;
  SumS=SumS+sum(abs(Si)')';
  SumN=SumN+sum(Ni')';
  SumN2=SumN2+sum((Ni.^2)')';

  eval(['save r',savename,' SumBER SumS SumN SumN2 K PG SNR'])
end
```

REPEAT.M

Produces a large matrix that contains many copies of a smaller one.

```
function s=repeat(seq,n)

if exist('n')~=1, n=100; end
[m1,m2]=size(seq);
s=seq(:)*ones(1,n);
s=s(:);
s=reshape(s,m1,n*m2);
```

RECT.M

Produces a rectangular waveform from a binary sequence.

```
function s=rect(seq,n)

if exist('n')~=1, n=100; end
[m1,m2]=size(seq);
seq=seq';
s=(1-(seq(:)*2)*ones(1,n))';
s=s(:);
s=reshape(s,n*m2,m1)';
```

TXFIL.M

Finds transmitter filter coefficients.

```
function [B,A,delay]=txfil(fo,fc,fs,fil)
```

```
if fil,
  F1=(fo-fc/2)/fs;
  F2=(fo+fc/2)/fs;
  [B,A]=ellip(6,0.1,60,2*[F1 F2]);
  delay=61;
else
  B=[];
  A=[];
  delay=[];
end
```

RXFIL.M

Finds receiver filter coefficients.

```
function [B,A,delay]=rxfil(fc,PG,fs,fil)

if fil,
  F=(fc/PG)/fs;
  [B,A]=ellip(4,0.1,60,2*F);
  [f,mag,maglog,phase,delay]=response(B,A,2,[0 fc/PG]/fs,0);
  k=find(f<(fc/(2*PG))/fs);
% delay=round(min(delay(k)));
  delay=2050;
else
  B=[];
  A=[];
  delay=[];
end
```

FILTERD.M

Filters a signal and introduces a delay.

```
function sfil=filterd(B,A,s,del,fil)

if fil,
  sfil=filter(B,A,[s zeros(1,del)]);
  sfil=sfil(del+1:length(sfil));
else
  sfil=s;
end
```

XBPSKER.M

Loads and illustrates the results of the BPSK system simulation.

```
function [Nu,BER,SNIRexp,SNIRth,SNIRth2,PEexp,PEth,PEth2]=Xbpsker(fil)

% Load Results
if nargin<1, fil=0; end
savename='bpsk';
if fil, savename=[savename,'f']; end
eval(['load r',savename])

% Determine Nu Axis Range
BER=(SumBER/K)';
%Nu=max(find(BER<1e-5)):length(SumBER');
Nu=20:length(SumBER');
BER=BER(Nu);

% Bit-Error-Rate Curve
SNIRexp=((SumS/K).^2)./((SumN2(Nu)-(SumN(Nu)).^2/K)/(K-1))';
SNIRth=1./((Nu-1)/(3*PG)+10^(-SNR/10)/(2*PG));
SNIRth2=2*PG./((Nu-1)+10^(-SNR/10));
PEexp=tail(sqrt(SNIRexp));
PEth=tail(sqrt(SNIRth));
PEth2=tail(sqrt(SNIRth2));
figure(1)
```

```
subplot
semilogy(Nu,PEexp,'y',Nu,PEth,'m--',Nu,PEth2,'g--',Nu,BER,'c')
grid
title('Bit Error Rate for BPSK-DS-CDMA System')
xlabel('No. of users')
ylabel('BER')

% BER vs. EUE Curve
EUEth=SNIRth2/2;
figure(2)
subplot
semilogy(num2db(EUEth),BER,'y',num2db(EUEth),PEth,'m--', ...
        num2db(EUEth),PEth2,'g--')
grid
title('BER vs. Eb/Io for BPSK-DS-CDMA System')
xlabel('Eb/Io (dB)')
ylabel('BER')

% Signal to Noise+Interference Ratio Curve
%figure(3)
%subplot
%semilogy(Nu,SNIRexp,'y',Nu,SNIRth,'m--',Nu,SNIRth2,'g--')
%axis([min(Nu) max(Nu) 0.6*min(SNIRth2) 1.2*max(SNIRth)])
%grid
%title('Signal to Noise+Interference Ratio for BPSK-DS-CDMA System')
%xlabel('No. of users')
%ylabel('SNIR')

figure(1)
```

APPENDIX C

Single-Cell QPSK DS-CDMA System Simulation

QPSKTRAN.M

Implements DS-QPSK transmitter.

```
function [t,dc,ds,pn,dpc,dps,s]=qpsktran(data,seq,fo,theta,fs,k,PG,P)

[d1,d2]=size(data);
[s1,s2]=size(seq);
To=floor(fs/fo);
Tc=k*To;
symbols=d2/2;
t=((1:symbols*PG*Tc)-1)/fs;
dc=rect(data(:,1:2:d2-1),PG*Tc);
ds=rect(data(:,2:2:d2),PG*Tc);
seq=repeat(seq,ceil(symbols*PG/s2));
seq=seq(1:symbols*PG);
pn=rect(seq,Tc);
dpc=pn.*dc;
dps=pn.*ds;
sc=dpc.*(sqrt(P)*cos(2*pi*fo*t+theta));
ss=dps.*(sqrt(P)*sin(2*pi*fo*t+theta));
s=sc+ss;
```

QPSKREC.M

Implements DS-QPSK receiver.

```
function [data,outc,outs,rcc,rcs,carc,cars,rp,pn]= ...
        qpskrec(r,seq,fo,theta,fs,k,PG)

[r1,r2]=size(r);
[s1,s2]=size(seq);
To=floor(fs/fo);
Tc=k*To;
symbols=floor(r2/(PG*Tc));
t=((1:r2)-1)/fs;
seq=repeat(seq,ceil(symbols*PG/s2));
seq=seq(1:symbols*PG);
pn=rect(seq,Tc);
rp=r.*pn;
carc=cos(2*pi*fo*t+theta);
cars=sin(2*pi*fo*t+theta);
rcc=rp.*carc;
rcs=rp.*cars;
outc=mean(reshape(rcc',PG*Tc,symbols*s1));
outc=reshape(outc,symbols,s1)';
outs=mean(reshape(rcs',PG*Tc,symbols*s1));
outs=reshape(outs,symbols,s1)';
bits=2*symbols;
data(:,1:2:bits-1)=outc<0;
data(:,2:2:bits)=outs<0;
```

QPSK.M

Implements DS-QPSK CDMA system.

```
function [ber,rec,Sc,Ss,Nc,Ns]= ...
        qpsk(ib,tran,seq,del,fo,theta,fs,k,PG,P,SNR,fil,graph)

% Main Parameters
if nargin<13, graph=[]; end
[M,bits]=size(tran);
symbols=bits/2;
To=floor(fs/fo);
Tc=k*To;
Tsymb=PG*Tc;
del(1)=Tsymb;
fc=fo/k;

% Transmiter & Receiver Filter Design
[Bt,At,delt]=txfil(fo,fc,fs,fil);
%[Br,Ar,delr]=rxfil(fc,PG,fs,fil);

% Ideal Output (No Noise or Interference)
[t,dc,ds,pn,dpc,dps,s1]= ...
      qpsktran(tran(1,:),seq(1,:),fo,theta(1),fs,k,PG,P);
sfil1=filterd(Bt,At,s1,0,fil);
rfil1=fliplr(filterd(Bt,At,fliplr(sfil1),0,fil));
[rec1,Sc,Ss]=qpskrec(rfil1,seq(1,:),fo,theta(1),fs,k,PG);

% Initialisation
s=zeros(1,symbols*Tsymb);
ber=zeros(M,1);
rec=zeros(M,bits);
Nc=zeros(M,symbols);
Ns=zeros(M,symbols);

% Main Loop
for i=1:M,

  % Transmiter
  [t,dc,ds,pn,dpc,dps,si]=qpsktran([ib(i,:) tran(i,:)], ...
       seq(i,:),fo,theta(i),fs,k,PG,P);
  si=si(del(i)+(1:symbols*Tsymb));
  s_old=s;
  s=s+si;

  % Transmiter Filter
  sfil=filterd(Bt,At,s,0,fil);

  % Thermal Noise
  n=randn(size(s));
  n=n/std(n)*sqrt(P)*10^(-SNR/20);
  r=sfil+n;

  % Receiver Filter
  rfil=fliplr(filterd(Bt,At,fliplr(r),0,fil));

  % Receiver
  [rec(i,:),Nc(i,:),Ns(i,:),rcc,rcs,carc,cars,rp,pn]= ...
       qpskrec(rfil,seq(1,:),fo,theta(1),fs,k,PG);
  Nc(i,:)=Nc(i,:)-Sc;
  Ns(i,:)=Ns(i,:)-Ss;

  % Bit Error Rate
  ber(i)=sum(sum(rec(i,:)~=tran(1,:)))/bits;
  diary on
  disp([i ber(i)])
  diary off

  % Plots
  if graph,
    figure(1)
      w=(1:Tc*10)+Tc*0;
      subplot(511)
      plot(t(w),s_old(w))
      title('Direct Sequence BPSK CDMA System (Transmitter)')
      ylabel('Old s(t)')
      grid
      subplot(512)
      plot(t(w),si(w))
      ylabel(['s',num2str(i),'(t)'])
```

```
      grid
      subplot(513)
      plot(t(w),s(w))
      ylabel('New s(t)')
      grid
      subplot(514)
      plot(t(w),sfil(w))
      ylabel('sfil(t)')
      grid
      subplot(515)
      plot(t(w),r(w))
      xlabel('t (msec)')
      ylabel('r(t)')
      grid
    figure(2)
      subplot(511)
      plot(t(w),rfil(w))
      title('Direct Sequence BPSK CDMA System (Receiver)')
      ylabel('rfil(t)')
      grid
      subplot(512)
      plot(t(w),pn(w))
      ylabel('pn(t)')
      V=axis;
      V([3 4])=[-1.2 1.2];
      axis(V);
      grid
      subplot(513)
      plot(t(w),rp(w))
      ylabel('rp(t)')
      grid
      subplot(514)
      plot(t(w),carc(w))
      ylabel('cosine')
      grid
      subplot(515)
      plot(t(w),rcc(w))
      xlabel('t (msec)')
      ylabel('rcc(t)')
      grid
    if symbols>1,
      figure(3)
      w=1:Tsymb*min(symbols,5);
      plot(t(w),rcc(w))
      title('Direct Sequence BPSK CDMA System (Receiver)')
      ylabel('rcc(t)')
      grid
    end
    figure(1)
    pause
  end
end
```

QPSKTEST.M

Tests the DS-QPSK CDMA system and saves simulation results.

```
% Main Parameters
M=64;
N=500;
symbols=20;
code=7;
fo=20;
fs=20*fo;
k=2;
P=1/2;
SNR=0;
fil=1;
graph=0;

% Constants
bits=2*symbols;
M=min(M,2^(code-1));
eval(['load code',num2str(code)])
```

```
PG=length(seq);
To=floor(fs/fo);
Tc=k*To;
Tsymb=PG*Tc;
savename='qpsk';
if fil, savename=[savename,'f']; end

% Initialisation
if exist(['r',savename,'.mat')==2,
  eval(['load r',savename])
else
  K=0;
  SumBER=zeros(M,1);
  SumSc=0;
  SumNc=zeros(M,1);
  SumNc2=zeros(M,1);
  SumSs=0;
  SumNs=zeros(M,1);
  SumNs2=zeros(M,1);
end

eval(['diary o',savename])
diary off

% Main Loop
for i=1:N
  diary on
  disp(['Iteration ',num2str(i)])
  disp(['Total Number of Bits: ',num2str(K)])
  diary off
  K=K+bits;

  % Random Delays & Carrier Phases of Users
  del=floor(Tsymb*rand(M,1))+1;
  del(1)=Tsymb;
  theta=2*pi*rand(M,1);

  % Random Codes
  [X,C]=sort(rand(size(seq,1),1));
  C=C(1:M);

  % Random Inputs
  ib=round(rand(M,2));
  tran=round(rand(M,bits));

  % BPSK Simulation
  [BERi,rec,Sci,Ssi,Nci,Nsi]=qpsk(ib,tran,seq(C,:), ...
        del,fo,theta,fs,k,PG,P,SNR,fil,graph);

  % Accumulation of Results
  SumBER=SumBER+BERi*bits;
  SumSc=sum([SumSc abs(Sci)]')';
  SumNc=sum([SumNc Nci]')';
  SumNc2=sum([SumNc2 Nci.^2]')';
  SumSs=sum([SumSs abs(Ssi)]')';
  SumNs=sum([SumNs Nsi]')';
  SumNs2=sum([SumNs2 Nsi.^2]')';

eval(['save r',savename, ...
      ' SumBER SumSc SumNc SumNc2 SumSs SumNs SumNs2 K PG SNR'])
end
```

XQPSKER.M

Loads and illustrates the results of the QPSK system simulation.

```
function [Nu,BER,SNIRcexp,SNIRsexp,SNIRth,SNIRth2, ...
          PEcexp,PEsexp,PEth,PEth2]=Xqpsker(fil)

% Load Results
if nargin<1, fil=0; end
savename='qpsk';
if fil, savename=[savename,'f']; end
eval(['load r',savename])
```

```
% Determine Nu Axis Range
BER=(SumBER/K)';
%Nu=max(find(BER<1e-5)):length(SumBER');
Nu=10:length(SumBER');
BER=BER(Nu);
L=K/2;

% Bit-Error-Rate Curve
SNIRcexp=((SumSc/L).^2)./((SumNc2(Nu)-(SumNc(Nu)).^2/L)/(L-1))';
SNIRsexp=((SumSs/L).^2)./((SumNs2(Nu)-(SumNs(Nu)).^2/L)/(L-1))';
SNIRth=1./(2*(Nu-1)/(3*PG)+10^(-SNR/10)/(2*PG));
SNIRth2=2*PG./(2*(Nu-1)+10^(-SNR/10));
PEexp=(1-tail(-sqrt(SNIRcexp)).*tail(-sqrt(SNIRsexp)))/2;
PEth=tail(sqrt(SNIRth))-0.5*tail(sqrt(SNIRth)).^2;
PEth2=tail(sqrt(SNIRth2))-0.5*tail(sqrt(SNIRth2)).^2;
figure(1)
subplot
semilogy(Nu,PEexp,'y',Nu,PEth,'m--',Nu,PEth2,'g--',Nu,BER,'c')
grid
title('Bit Error Rate for QPSK-DS-CDMA System')
xlabel('No. of users')
ylabel('BER')

% BER vs. SNIR Curve
EUEth=SNIRth2/2;
figure(2)
subplot
semilogy(num2db(EUEth),BER,'y',num2db(EUEth),PEth,'m--', ...
        num2db(EUEth),PEth2,'g--')
grid
title('BER vs. Eb/Io for QPSK-DS-CDMA System')
xlabel('Eb/Io (dB)')
ylabel('BER')

% Signal to Noise+Interference Ratio Curve
%figure(3)
%subplot
%semilogy(Nu,SNIRcexp,'y',Nu,SNIRsexp,'y:',Nu,SNIRth,'m--',Nu,SNIRth2,'g--')
%axis([min(Nu) max(Nu) 0.6*min(SNIRth2) 1.2*max(SNIRth)])
%grid
%title('Signal to Noise+Interference Ratio for QPSK-DS-CDMA System')
%xlabel('No. of users')
%ylabel('SNIR')

figure(1)
```

APPENDIX D

Multiple-Cell CDMA System Simulation: Reverse Link

TRAN.M

Co-ordinate transformation.

```
function [X,Y]=tran(x,y)
  X=x-y/2;
  Y=y*sqrt(3)/2;
end
```

INVTRAN.M

Inverse co-ordinate transformation.

```
function [x,y]=invtran(X,Y)
  a=sqrt(3);
  x=X+Y/a;
  y=Y*2/a;
end
```

CELL.M

Determines the co-ordinates of the nearest cell centre.

```
function [xm,ym]=cell(x,y);

  if max(size(x))>1 & max(size(y))==1, y=y*ones(size(x)); end
  if max(size(y))>1 & max(size(x))==1, x=x*ones(size(y)); end
  k=find(x==0 & y==0);
  x(k)=x(k)+1.5;
  xm=floor(x);
  ym=floor(y);
  a=mod(xm+ym,3);
  xf=x-xm;
  yf=y-ym;
  k=find(a==1 | (a==2 & xf>=yf));
  xm(k)=xm(k)+1;
  k=find(a==1 | (a==2 & xf<yf));
  ym(k)=ym(k)+1;
  k=find(xm==0 & ym==0);
  if k~=[],
    r=sqrt(x.^2+y.^2-x.*y);
    x(k)=x(k)*1.5 ./ r(k);
    y(k)=y(k)*1.5 ./ r(k);
    [xm(k),ym(k)]=cell(x(k),y(k));
  end

end
```

FMEAN.M

Integrand that corresponds to the mean of interference-to-signal ratio.

```
function f=fmean(x,y,R)
```

```
if nargin<3, R=1000; end
if min(size(x))==1 & min(size(y))==1, [x,y]=meshgrid(x,y); end
[xa,xb]=size(x);
b=log(10)/10;
sigma=8;
s=sqrt(2)*sigma;
e=exp((sigma*b)^2);
[xm,ym]=cell(x,y);
rm=sqrt((x-xm).^2+(y-ym).^2-(x-xm).*(y-ym));
r0=sqrt(x.^2+y.^2-x.*y);
rat=max(rm,1/R)./max(r0,1/R);
f=tail(40*log10(rat)/s+s*b) .* rat.^4 * e;

end
```

FMEAN1.M

Performs first integration of function FMEAN.

```
function f=fmean1(y)

diary on
f=int8('fmean',0,100,1e-6,0,y)
diary off

end
```

IMEAN.M

Performs second integration of function FMEAN and saves results.

```
diary omean
diary off
a=3/8;
[I,cnt]=int8('fmean1',0,100,1e-6,0);
E=a*I;
save rmean
diary on
disp('END')
diary off
```

FVAR.M

Integrand that corresponds to the variance of interference-to-signal ratio.

```
function f=fvar(x,y,R)

if nargin<3, R=1000; end
if min(size(x))==1 & min(size(y))==1, [x,y]=meshgrid(x,y); end
[xa,xb]=size(x);
a=3/8;
b=log(10)/10;
sigma=8;
s=sqrt(2)*sigma;
e=exp(2*(sigma*b)^2);
[xm,ym]=cell(x,y);
rm=sqrt((x-xm).^2+(y-ym).^2-(x-xm).*(y-ym));
r0=sqrt(x.^2+y.^2-x.*y);
rat=max(rm,1/R)./max(r0,1/R);
A=40*log10(rat)/s+s*b;
f=e * rat.^8 .* (e*tail(A+s*b) - a*(tail(A)).^2);

end
```

FVAR1.M

Performs first integration of function FVAR.

```
function f=fvar1(y)

diary on
f=int8('fvar',0,50,1e-6,0,y)
diary off

end
```

IVAR.M

Performs second integration of function FVAR and saves results.

```
diary ovar
diary off
a=3/8;
[I,cnt]=int8('fvar1',0,50,1e-6,0);
V=a*I;
save rvar
diary on
disp('END')
diary off
```

INT8.M

Numerical integration. Based on built-in MATLAB function QUAD8, with some changes so that it can handle vector as well as scalar functions.

```
function [Q,cnt] = int8(funfcn,a,b,tol,trace,p1,p2,p3,p4,p5,p6,p7,p8,p9)

%QUAD8  Numerical evaluation of an integral, higher order method.
%       Q = QUAD8('F',A,B) approximates the integral of F(X) from A
%       to B to within a relative error of 1e-3.  'F' is a string
%       containing the name of the function.  The function must return
%       a vector of output values if given a vector of input values.
%       Q = QUAD8(F,A,B,TOL) integrates to a relative error of TOL.
%       Q = Inf is returned if an excessive recursion level is reached,
%       indicating a possibly singular integral.
%       Q = QUAD8(F,A,B,TOL,TRACE) integrates to a relative error of TOL and
%       traces the function evaluations with a point plot of the integrand.
%
%       QUAD8 uses an adaptive recursive Newton Cotes 8 panel rule.
%       Q = QUAD8('F',A,B,TOL,TRACE,P1,P2,...) allows coefficients P1, P2, ...
%       to be passed directly to function F:   G = F(X,P1,P2,...).
%       To use default values for TOL or TRACE, you may pass in the empty matrix ([]).
%       See also QUAD.

%       Cleve Moler, 5-08-88.
%       Copyright (c) 1984-94 by The MathWorks, Inc.

% [Q,cnt] = quad8(F,a,b,tol) also returns a function evaluation count.

if nargin < 4, tol = 1.e-3; trace = 0; end
if nargin < 5, trace = 0; end
if isempty(tol), tol = 1.e-3; end
if isempty(trace), trace = 0; end
% QUAD8 usually does better than the default 1e-3.
h = b - a;

% Top level initialization, Newton-Cotes weights
w = [3956 23552 -3712 41984 -18160 41984 -3712 23552 3956]/14175;
x = a + (0:8)*(b-a)/8;
%y = feval(funfcn,x);
% set up function call
args = '(x';
args1 = [];
for n = 1:nargin-5
   args1 = [args1,',p',int2str(n)];
end
args1 = [args1,')'];
args = [args args1];
y = eval([funfcn,args]);
```

```
yllow = min([min(real(y)) min(imag(y))]);
ylhi = max([max(real(y)) max(imag(y))]);
lims = [min(x) max(x) yllow ylhi];
ind = find(~finite(lims));
if ~isempty(ind)
    [mind,nind] = size(ind);
    lims(ind) = 1.e30*(-ones(mind,nind) .^ rem(ind,2));
end
if trace
    axis(lims);
% doesn't take care of complex case
    plot([a b],[real(y(1,1)) real(y(1,9))],'.'), hold on
    if any(imag(y))
        plot([a b],[imag(y(1,1)) imag(y(1,9))],'+')
    end
end
lev = 1;

% Adaptive, recursive Newton-Cotes 8 panel quadrature
if any(any(imag(y)))
    Q0 = 1e30;
else
    Q0 = inf;
end
[Q,cnt] = eval(['int8stp(funfcn,a,b,tol,lev,w,x,y,Q0,trace',args1]);
cnt = cnt + 9;
if trace
    hold off
    axis('auto');
end
```

INT8STP.M

Used by INT8. This function is also based on the built-in MATLAB function QUAD8STP.

```
function [Q,cnt] = int8stp(FunFcn,a,b,tol,lev,w,x0,f0,Q0,trace,p1,p2, ...
                           p3,p4,p5,p6,p7,p8,p9)
%QUAD8STP Recursive function used by QUAD8.
%       [Q,cnt] = quad8stp(F,a,b,tol,lev,w,f,Q0) tries to approximate
%       the integral of f(x) from a to b to within a relative error of tol.
%       F is a string containing the name of f.  The remaining arguments are
%       generated by quad8 or by the recursion.  lev is the recursion level.
%       w is the weights in the 8 panel Newton Cotes formula.
%       x0 is a vector of 9 equally spaced abscissa is the interval.
%       f0 is a vector of the 9 function values at x.
%       Q0 is an approximate value of the integral.
%       See also QUAD8 and QUAD.

%       Cleve Moler, 5-08-88.
%       Copyright (c) 1984-94 by The MathWorks, Inc.

LEVMAX = 10;
args1 = [];
for n = 1:nargin-10
   args1 = [args1,',p',int2str(n)];
end
args1 = [args1,')'];

% Evaluate function at midpoints of left and right half intervals.
x = zeros(1,17);
[fa,fb]=size(f0);
f = zeros(fa,17);
x(1:2:17) = x0;
f(:,1:2:17) = f0;
x(2:2:16) = (x0(1:8) + x0(2:9))/2;
%f(2:2:16) = feval(FunFcn,x(2:2:16));
f(:,2:2:16) = eval([FunFcn,'(x(2:2:16)',args1]);
if trace
    plot(x(2:2:16),f(1,2:2:16),'.');
    if any(any(imag(f)))
        plot(x(2:2:16),imag(f(1,2:2:16)),'+');
    end
```

```
end
cnt = 8;

% Integrate over half intervals.
h = (b-a)/16;
Q1 = h*w*f(:,1:9).';
Q2 = h*w*f(:,9:17).';
Q = Q1 + Q2;

% Recursively refine approximations.
if any(abs(Q - Q0) > tol*abs(Q)) & lev <= LEVMAX
   c = (a+b)/2;
   [Q1,cnt1] = eval(['int8stp(FunFcn,a,c,tol/2,lev+1,w,x(1:9), ...
                     f(:,1:9),Q1,trace',args1]);
   [Q2,cnt2] = eval(['int8stp(FunFcn,c,b,tol/2,lev+1,w,x(9:17), ...
                     f(:,9:17),Q2,trace',args1]);
%  [Q1,cnt1] = int8stp(FunFcn,a,c,tol/2,lev+1,w,x(1:9),f(1:9),Q1,trace);
%  [Q2,cnt2] = int8stp(FunFcn,c,b,tol/2,lev+1,w,x(9:17),f(9:17),Q2,trace);
   Q = Q1 + Q2;
   cnt = cnt + cnt1 + cnt2;
end
```

PDFI.M

Monte-Carlo simulation for reverse link.

```
function [x,pdf,l,f]=pdfi(d,xmax,M,bins)

% Main Parameters
R=1000;
sigma=8;
s=sqrt(2)*sigma;
a=3/8;

% Define Grid
x=0:d:xmax-d;
y=0:d:xmax-d;
[x,y]=meshgrid(x,y);

% Find Cell Centers
[xm,ym]=cell(x,y);

% Distances from Cell Centers & from Origin
rm=sqrt((x-xm).^2+(y-ym).^2-(x-xm).*(y-ym));
r0=sqrt(x.^2+y.^2-x.*y);

% Ratio of the 2 Distances
rat4=(max(rm,1/R)./max(r0,1/R)).^4;
rat4=rat4(:)';
N=length(rat4);

% Initialisation
f=zeros(1,M);

% Main Loop
for j=1:M

  % Voice activity RV
  psi=(rand(1,N)<a);

  % Shadowing RV
  shad=(10^(s/10)).^randn(1,N);

  % Integrand
  f(j)=sum(psi .* rat4 .* shad .* (rat4 .* shad<=1));

  % Display Current Step (every 10 steps)
  if mod(j,10)==0,
    diary on
    disp(j),
    diary off
  end

end
```

```
% Find Histogram
l=length(f);
[n,x]=hist(f,bins);
pdf=n/(l*(x(2)-x(1)));
```

RUNPDFI.M

Used to pass appropriate parameters to PDFI and save simulation results.

```
% Runpdfi.m

% Main Parameters
bins=40;
d=.1;
xmax=20;
M=10000;

% Output to file 'opdfi'
diary opdfi
diary off

% Run Simulation
[x,pdf,l,fnew]=pdfi(d,xmax,M,bins);

% Save Results
if exist('rpdfi.mat')==2,
  load rpdfi
  f=[f fnew];
  l=length(f);
  [n,x]=hist(f,bins);
  pdf=n/(l*(x(2)-x(1)));
else
  f=fnew;
end
save rpdfi x pdf l f d
```

CAP.M

Finds reverse link performance as a function of the number of users.
Used to evaluate reverse link capacity.

```
function C=cap(Ns,m,s2,ld);

% Define Load
if nargin<4, ld=1; end
ld=max(ld,eps);

% Main Parameters
a=3/8;
W=1.25e6;
R=8e3;
EUE=db2num(7);
SNR=db2num(-1);
d=(W/R)/EUE-1/SNR;

% Calculate Capacity
Ns=Ns(:)';
L=length(Ns);
C=zeros(1,L);
for i=1:L
  n=Ns(i);
  k=0:n-1;
  t=tail((d-k-m*n*ld)./sqrt(s2*n*ld));
  C(i)=sum(comb(n-1,k) .* a.^k .* (1-a).^(n-1-k) .* t);
end
```

COMB.M

COMB(N,M) finds the number of combinations of M elements out of N.

```
function c=comb(n,m)

m=m(:)';
L=length(m);
c=zeros(1,L);
for i=1:L
  if m(i)>n/2, c(i)=prd(m(i)+1:n)/prd(1:n-m(i));
  else         c(i)=prd(n-m(i)+1:n)/prd(1:m(i));
  end
end
```

PRD.M

Finds the product of the elements of a matrix.

```
function p=prd(X);

p=prod(X);
f=find(isempty(p));
p(f)=ones(size(f));
```

APPENDIX E

Multiple-Cell CDMA System Simulation: Forward Link

RING.M

Contains co-ordinates of the 18 nearest (to the origin) cell centres.

```
function [cx,cy]=ring

cx=[0 1 2 1 -1 -2 -1 2 3 4 3 2 0 -2 -3 -4 -3 -2 0];
cy=[0 2 1 -1 -2 -1 1 4 3 2 0 -2 -3 -4 -3 -2 0 2 3];
```

FI.M

Monte-Carlo simulation for forward link (Chernoff upper bound technique).

```
function [x,pdf,l]=fi(d,M,bins)

f=[];
sigma=8;
[cx,cy]=ring;
N=length(cx);
for x=0:d:1-d
  for y=0:d:x/2
    r2=((x-cx).^2+(y-cy).^2-(x-cx).*(y-cy))';
    S=(r2.^(-2) * ones(1,M)) .* (10^(sigma/10)).^randn(N,M);
    [s1,k]=max(S);
    k=(0:M-1)*N + k;
    S(k)=zeros(size(k));
    f=[f sum(S)./s1];
  end

  diary on
  disp(x)
  diary off

end
l=length(f);
[n,x]=hist(f,bins);
pdf=n/(l*(x(2)-x(1)));
```

SFI.M

Monte-Carlo simulation for forward link (Gaussian approximation technique).

```
function [x,pdf,l,f]=sfi(d,M,bins)

% Main Parameters
sigma=8;

% Cell Centers of 2 Rings
[cx,cy]=ring;
N=length(cx);

% Define Grid
x=0:d:1-d;
y=0:d:1-d;
```

```
L=length(x)^2;
[x,y]=meshgrid(x,y);
[x,cx]=meshgrid(x,cx);
[y,cy]=meshgrid(y,cy);

% Distances from Cell Centers
r4=((x-cx).^2+(y-cy).^2-(x-cx).*(y-cy)).^(-2);

% Initialisation
f=zeros(1,M);

% Main Loop
for j=1:M

  % Received Powers
  s=r4 .* (10^(sigma/10)).^randn(N,L);

  % Maximum Received Power
  [S,k]=max(s);
  k=(0:L-1)*N + k;

  % Ratio of Sum of All Other Powers to the Maximum
  s(k)=zeros(size(k));
  f(j)=sum(sum(s)./S);

  % Display Current Step (every 10 steps)
  if mod(j,10)==0,
    diary on
    disp(j),
    diary off
  end

end

% Find Histogram
l=length(f);
[n,x]=hist(f,bins);
pdf=n/(l*(x(2)-x(1)));
```

RUNSFI.M

Used to pass appropriate parameters to SFI and save results.

```
% Main Parameters
bins=40;
d=.05;
M=2;

% Output to file 'osfi'
diary osfi
diary off

% Run Simulation
[x,pdf,l,fnew]=sfi(d,M,bins);

% Save Results
if exist('rsfi.mat')==2,
  load rsfi
  f=[f fnew];
  l=length(f);
  [n,x]=hist(f,bins);
  pdf=n/(l*(x(2)-x(1)));
else
  f=fnew;
end

save rsfi x pdf l f d
```

FCHERN.M

Function whose minimum provides Chernoff bound estimate.

```
function E=fchern(s,n,f,p,type)

[sa,sb]=size(s);
E=zeros(sa,sb);
s=s(:)';
a=3/8;
d=0.8*1250/8/10^0.5;
A=exp(f*s) .* (p*ones(1,length(s)));
if type==1,
  E(:)=((1-a)+a*sum(A)).^n .* exp(-s*d);
else
  E(:)=(sum(A)).^n .* exp(-s*d);
end
```

CHERN.M

Calculates minimum of FCHERN.

```
function [C,Sm]=chern(Ns,type,graph)

load rfi
m=1:floor(0.7*length(x));
p=(pdf(m)*(x(2)-x(1)))';
f=1+(x(m))';
Ns=Ns(:)';
L=length(Ns);
C=zeros(1,L);
Sm=C;
s=0:.01:2;
for i=1:L
  disp(Ns(i))
  E=fchern(s,Ns(i),f,p,type);
  k=max(find(E<1));
  lim=s(k);
  if exist('graph')==1, plot(s(1:k),E(1:k)), end
  Sm(i)=fmin('fchern',0,lim,[0,1e-4],Ns(i),f,p,type);
  C(i)=fchern(Sm(i),Ns(i),f,p,type);
end
```

CAP.M

Finds forward link performance as a function of the number of users. Used to evaluate forward link capacity.

```
function C=cap(Ns,type)

if type<3,
  C=chern(Ns,type);
else

  % Main Parameters
  b=0.8;
  W=1.25e6;
  R=8e3;
  EUE=db2num(5);
  d=b*(W/R)/EUE;

  % Mean & Variance of Interference
  m=1.549167;
  s2=0.368528316;

  % Calculate Capacity
  C=tail((d-m*Ns)./sqrt(s2*Ns));

end
```

APPENDIX F

Auxiliary Functions

TAIL.M

Gaussian upper tail function.

```
function y=tail(x)

if x==[],
  y=[];
else
  y=erfc(x/sqrt(2))/2;
end
```

NUM2DB.M

Converts a number from linear scale to decibels.

```
function y=num2db(x)

y=10*log10(x);
```

DB2NUM

Converts a number from decibels to linear scale.

```
function y=db2num(x)

y=10.^(x/10);
```

MOD.M

Modulo function. Based on built-in MATLAB function REM, but works also for negative numbers.

```
function [m,n]=mod(x,y)

x=floor(x);
y=floor(y);
n=floor(x ./ y);
m=x - n .* y;
```

DEC2BIN.M

Converts an integer from decimal to binary.

```
function bin=dec2bin(dec,n)

if exist('n')~=1, n=-Inf; end
dec=round(dec(:));
bin=rem(dec,2);
dec=(dec-bin)/2;
while (any(dec) & n==-Inf) | size(bin,2)<n,
  b=rem(dec,2);
```

```
  dec=(dec-b)/2;
  bin=[b bin];
end
```

ROWS.M

Constucts a large matrix from many different vectors as its rows.

```
function M=rows(m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12)

M=m1;
for k=2:nargin
  K=num2str(k);
  [Mx,My]=size(M);
  eval(['[mx,my]=size(m',K,');'])
  if My>my,
    eval(['m',K,'=[m',K,' zeros(mx,My-my)];'])
  elseif My<my,
    M=[M zeros(Mx,my-My)];
  end
  eval(['M=[M; m',K,'];'])
end
```