

Data transmission analysis of Wide Band CDMA Communication System

Mohammad Asad Rehman Chaudhry*, Zakia Iqbal **, Noor M Sheikh, Yang Chenyang*,
Muhammad Inam, Ali Ahmad, Mustajaab Daar, Shajee Haider

**Beijing University of Avoinics and Astronautics, Beijing ,China*

*Department of Electrical Engineering,
University of Engineering & Technology Lahore, Pakistan*

***Khwarzimi Institue of Computer Sciences,
University of Engineering & Technology Lahore, Pakistan*

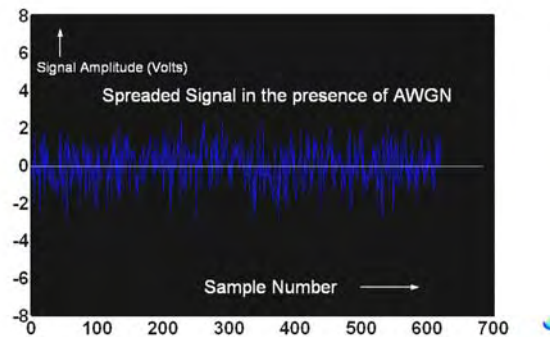
Abstract

The goal for the next generation of mobile communications is to efficiently integrate a wide variety of communication services like high speed data, video and multimedia traffic as well as voice signals. Out of this, video conferencing through mobile communications will be the greatest achievement. The technology which seems to tackle the challenge to make these services available is one of the 3G systems defined by IMT-2000, the Wide Band CDMA. To combine the WCDMA air interface with the present fixed network of Global System of Mobile communications (GSM) is undoubtedly the greatest challenge for the communication engineers and the scientists of the day. One of the main purposes of this paper is to inspire the young engineers to take this challenge. A WCDMA system has been implemented in Matlab. All kinds of data like image, sound and text have been transmitted through a channel modeled to be having multipath fading and AWGN. The data is received by means of the Rake receiver which is the biggest advantage of spread spectrum communications as is WCDMA. Hoping this simulation to be a valuable fun and paving way for further work on WCDMA.

Out of four 3G systems defined by IMT-2000, CDMA appears to be preferred technology for wireless communications owing to its multidimensional advantages as security, increased capacity, reluctance to interference and above all the multipath diversity. Much higher bandwidth than required is really painful but the frequency reuse factor being unity is fascinating where all the users, transmitting their data over the same carrier, yet achieving great shielding from the multiuser interference and multipath fading.

1. Introduction

What CDMA is? In a nut shell, every user being assigned a PN code which is nothing other than a high rate sequence of 1's and -1's occupying white noise like properties such as an impulsive response when correlated with itself and a sharp decay on correlating with its delayed version. A group of such codes which are orthogonal to each other, is really the key to the success of CDMA. Once the data from all users spreaded after multiplication with their individual codes, summed up and thrown onto a noisy, full of interference, channel, the receiver is to despread the data with a coherent copy of the same code for a particular user, which on multiplying with the data from other users produces a low amplitude noise which on integrating, produces a zero due to its constant wiggling about a zero level. Pasted below is an event from the simulation results which shows the noise like characteristics of the spreaded signal and that's why jammers have to cover miles to hack a spread spectrum signal.



Here comes the mathematics;

$$\text{Multi-user base-band data} = r(t) = \sum_{i=1}^N [S_i(t) * C_i(t) + n_i(t)] \quad (N \text{ is number of users})$$

$$\begin{aligned}
 r_1(t) &= C_1(t) * r(t) = C_1(S_1 * C_1 + S_2 * C_2 + S_3 * C_3 + \dots \\
 &+ S_N * C_N + n_1 + n_2 + n_3 + \dots + n_N) \\
 &= S_1 * C_1 + S_2 * C_1 * C_2 + S_3 * C_1 * C_3 + \dots + S_N * C_1 * C_N \\
 &+ n_1 * C_1 + n_2 * C_1 + \dots + n_N * C_1
 \end{aligned}$$

Due to orthogonality

$$C_i * C_j = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{otherwise} \end{cases}$$

Now integrating over the symbol period and assuming the signal 'S' to have single value over the whole period i.e; 1 or -1

$$\begin{aligned}
 \text{Received data for user1} &= \frac{1}{T} \int_0^T r_1(t) dt = \frac{S_1}{T} \int_0^T C_1 * C_1 dt \\
 &= S_1 = \text{signal transmitted for user1}
 \end{aligned}$$

How is CDMA secured for example against jamming? Just by spreading the signal components, we can fight well against the noise just as an army of 100 men, if spread over the field, is less likely to be destroyed by a bomber as compared to when all 100 men are combined together over a single place.

Why CDMA can accommodate large no. of users? Suppose N users in a cell, transmitting same amount of input power P, the interference experienced by one from all others is $I = (N-1)P$. The interference per unit of bandwidth used by the spreaded signal (which is W) is $I_0 = I/W$. As proved by Viterbi [2], interference from other users behaves just like AWGN, the signal to noise ratio $E_b/I_0 = P/(I_0R)$, where R being the data rate. Thus from combination of these equations and assuming N to be very large, the capacity of a CDMA system is found to be $N = (W/R) / (E_b/I_0)$. Thus E_b/I_0 being fixed, the capacity simply is a function of W/R which is our spreading factor.

How is CDMA reluctant to interference? For a single user i.e; $N=1$ and we find that $E_b/I_0 = (WP/IR)$, again W/R being spreading factor G_p and E_b/I_0 being fixed, we see $1/P$ is proportional to G_p which means that higher interference and lesser amount of input power is affordable given that we spread the information BW by a factor G_p .

"Multipath is our friend" is just proved by the implementation of "rake Receiver" where every rake finger accepts the multipath signal and extracts a particular component from it which is in phase with the underlying PN code used in that finger. The results from all fingers are put together to optimize the signal to noise ratio such that multipath signal is converted into multiplied signal.

2. Our Simulation

Image, voice and text data transmission has been simulated in Matlab. To reduce the simulation time, the long time voice and large sized images are avoided. Text message length is limited to 50 characters.

2.1 Image Transmission

Images of format like JPEG, TIF, BMP, CUR, HDF, ICO, PCX, PNG, XWD can be handled in Matlab by a matrix with three dimensions in case of colored image and when a single dimension of the matrix is used, the image becomes black & white. The main aspect of image transmission simulation has been the efforts to reduce the size of image in order to transmit lesser number of bits. The Algorithm employed to transmit images is given below.

Input: images selected by users

Output: The received images

1. Read an image selected by a user from the root directory; when read, we get a matrix.
2. Convert the matrix if of 3D, into 1D. Repeat above two steps for all users in the multi-user environment and get a group of data which is in the form of color values of the pixels.
3. Encode each value of the color into 8 bits and hence a junk of binary data is formed. We can reshape this data into a single matrix whose rows correspond to the data of each user.
4. Now jump to the "Internal Data Processing" unit of this Simulation.
5. Once the binary data is received and converted back into pixel color values, we can display the image from each user as the received image.
6. The simulation can be repeated by varying the values of SNR, interference from other users, spreading factor etc.

2.2 Voice Transmission

Matlab handles voice with the help of a vector containing the samples of voice within the range [-1,1]. Sampling frequency, according to Nyquist must have a value greater than 10 KHz, although results at 5 KHz \rightarrow 10 KHz are also acceptable. Simulation time has been a major problem in voice simulation since a voice of 1sec may require transmission of 10000 samples with each sample encoded into 8 bits. To reduce this time, voice has been compressed in two ways

1. by mu-law compressor
2. by very rough approximation which is acceptable as the response of human ear to the voice is logarithmic and even if a lot of samples are omitted and then the remaining voice played at a frequency which is less than the original sampling frequency by the same factor as the factor of compression, the voice is just bearable. Just for example, if out of 10000 samples of voice, if half are omitted and remaining are played at 5000 Hz, the resulting voice is acceptable. Moreover, taking into account the fact that the voice of

noble and reasonable people very less often changes abruptly, the lost samples have been tried to recover by linearly averaging the two corners samples to get the middle sample. The algorithm for sound transmission goes like this:

Input: voices of different users

Output: received voices

1. Record a sound from an external device, when recorded we get the vector having samples of the voice sampled at the user's assigned value.
2. Drop half or more of the samples thus doing lossy compression to reduce the size of data.
3. Convert the sampled voice into mu-law compressed voice hence getting levels of voice b/w 0 → 255
4. Encode each sample value into 8 bits.
5. Repeat above steps for all users, finally getting a matrix whose rows contain the data for each user.
6. Now jump to the "Internal Data Processing" unit of this simulation.
7. When the whole data is received via Rake receiver, estimate the lost samples by linear estimation.
8. Now play back the sounds of all users. Again this simulation can be repeated for different values of SNR, interference level and the spreading factor.

2.3 Text Transmission

Just as we enjoy the SMS over self phones, text messages transmission has been simulated. Although the text data occupies very less space as compared to the image and voice, the length of text messages has been limited to 50 characters. Every text character has its decimal value and in a mathematical environment such as provided by the Matlab, that value can be used for identifying the character. The algorithm for the text transmission simulation is as follows:

Input: Text messages of different users to be transmitted

Output: Received text messages

1. Get text message as a string from all users.
2. Encode each character of the messages into 8 bit ASCII code. And finally get a matrix whose rows contain the binary data for each user.
3. Now jump to the "Internal Data Processing" unit of this simulation.
4. Once the data simulated and received, convert each 8-bit code back into the character format
5. And finally display the message of each user.
6. The Simulation can be repeated for various values of SNR, multi-user interference and the spreading factor.

2.4 Internal Data Processing Unit

Data in every format is to come finally in binary form that is cluster of bits and the binary data

from its transmission to reception resides in this unit of simulation which in fact contains all those sub units which a modern WCDMA communication system may have. Following are the different units used in the simulation of WCDMA system.

2.4.1 Frame Maker. Data from each user may consist of some thousands of bits. Transmitting the data frame by frame is more efficient than the transmission of whole junk of data. Frame maker cuts frames of length assigned by the user, from the data vector. However, using frame of 8 bits is the best case as far as the run time of simulation is concerned.

2.4.2 Convolutional Encoder. To reduce the bit error rate, the error control encoding is an integral part of a wireless communication system. Coding reduces E_b/N_0 required to achieve a specified BER under a specified modulation scheme. An encoding gain of about 4 to 6 db is achievable through convolutional encoding using the Binary Phase Shifting Key (BPSK).

What is a convolutional encoder? In simple words, it is a block that converts each block of 'k' input bits into a block of 'n' output bits each of which depends upon a particular number of bits that has been passed through the encoder before that particular bit. This mutual dependence is actually key to the all around success of the convolutional codes. A convolutional encoder is recognized by some parameters like:

k=number of bits that are pumped into the encoder tail at once i.e; after each clock cycle

n=number of code bits associated with k-bit input block

K=the constraint length which is the number of k-tuple stages in the encoder shift register

$r=k/n$ =code rate is simply the reciprocal of the excess band width used due to encoding

A, $k=1, n=2, K=3$ and hence rate $\frac{1}{2}$, convolutional encoder has been used. So every single bit of the input data is converted into 2 code bits each of which depends upon 2 previous bits of the data present in the 3-bit shift register at the moment when that particular bit is inserted. The algorithm goes like this:

Input: Data vector A, connection matrix C, k

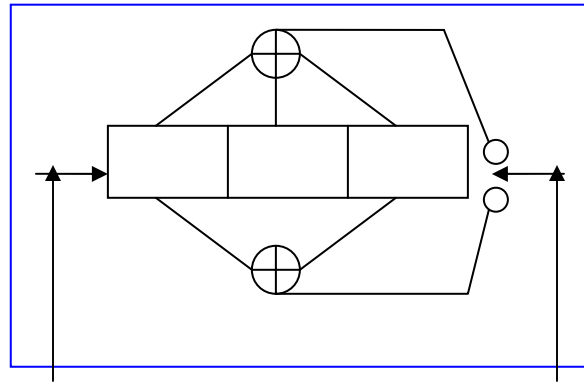
Output: The convolutionally encoded data

1. Make zero padding if the length of input vector is not divisible by k.
2. Add flush bits to the input vector equal to (K-1) times the 'k' in number. Flush bits are required to clear the contents of shift register before a new input vector is passed through this algorithm.
3. Generate a matrix 'S' whose columns contain the state of the shift register after each clock cycle. It can be done by passing the input bits one by one to the tail of the shift register. The size of this matrix will be

$(K*k)$ by $((\text{length}[A] + \text{no. of padded zeros})/k + \text{no. of flush bits i.e; } (K-1)*k)$.

4. Find out the product $= C*S$ with all additions being modulo 2 and reshape this product matrix into one dimensional vector which will be the code vector

The decoding of convolutional codes has been done by the Viterbi decoding algorithm which will be discussed in incoming sections.



input k=1, K=3, Convolutional Encoder output

2.4.3 Base Modulator / Demodulator. Now the Base Modulation is always required to get the desired spectral characteristics in the base band signal according to the properties of the channel. Although in any simulation like this, base modulation is not a very important task but has been done for convenience. It is really simple to do in matlab

Input: Binary data in the form of 1's and 0's.

Output: Base modulated data.

Simply check every bit, if it is 0 replace it with -1.

The procedure for demodulation is just opposite of it.

2.4.4 Walsh Modulator. The Walsh sequences are assigned to each channel of the system for the duration of a call. These sequences are orthogonal to each other ensuring that user signals are also orthogonal. The multiple access interference (MAI) can be theoretically zero if the signals are mutually orthogonal [1]. In short, a Walsh sequence is the identity of a channel and the multiplexing of the channels is accomplished with the help of Walsh sequences such that all channels are distinguished due to their orthogonality. The Walsh sequences can be created by the rows of Hadamard matrices of order 2^n where $n=1, 2, 3, \dots$ Next is the algorithm

Input: N =order of Hadamard matrix, u =no. of users

Output: last 'u' rows of the N by N Hadamard matrix

Base: $H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

1. For $x=1$ to $N/2$

$H_{2x} = [H_x \ H_x; H_x \ -1*H_x]$

2. Return last 'u' rows of the Hadamard matrix.

2.4.5 PN Code Generator. Actually, the letter 'C' in CDMA is both for PN code and the Walsh code as far as a forward channel is concerned i.e; the channel from base station to mobile station. The PN code is the identity of a base station. In other words, Walsh sequence eliminate the multiuser interference whereas the PN code eliminates the multicell

interference. Gold sequences have been used as PN code due to their extreme closeness to the white noise as far as the auto-correlation function is concerned. Following algorithm has been used.

Input: C_1, C_2 : the connection matrices for shift registers, b =no. of base stations

Output: 'b' number of PN sequences

1. First generate two maximum length shift register sequences according to the shift register connections to the modulo2 adders given by C_1 and C_2 . This will take total $(2^{\text{length}[C_1]} - 1)$ number of shifts.

2. Now create $(2^{\text{length}[C_1]} - 1)$ number of PN codes by shifting one sequence each time and adding it to the other in modulo2 addition.

3. Push each sequence to a row of $(2^{\text{length}[C_1]} - 1)$ by $(2^{\text{length}[C_1]} - 1)$ matrix.

4. Return first (or last) 'b' rows of the above matrix.

2.4.6 Pass Band Modulator/Demodulator. After the data from each user is orthogonally multiplexed and the resulting base band signal spreaded by the base station identifying PN code, the spreaded signal modulates a sinusoidal carrier

The BPSK has been used to avoid two streams of data as in case of QPSK. However, the pass band modulation / demodulation is not an essential part of a computer simulation simply because the purpose of modulation is to avoid larger antenna sizes and it is really the real time application in which it matters deeply.

2.4.7 Multipath Channel Model. In the urban areas, where the abundance of mobile users is present usually, the signal from transmitting antenna reaches the receiving antenna, in addition to the LOS component, from various paths with different amount of attenuation and delay depending upon the path length and the geographical conditions of that particular region e.g. the heights of the buildings and

the weather conditions. But whatever the case may be we cannot avoid multipath. It is the WCDMA which optimizes multipath fading (the attenuation of the received signal with respect to a threshold) through Rake Reception. The model is implemented as follows:

Input: Transmitted Signal

Output: The multipath signal

1. Take the input signal and produce given number of its replicas.
2. Introduce a different delay in each replica e.g. delay of one, two or more chips hence inviting the frequency selective fading.
3. Now attenuate each replica by a different factor which is obtained from the power delay profile of a particular area.
4. Add AWGN and the interference to every version of the transmitted signal
5. Introduce Doppler Shift to every part.
6. Combine all replicas to get the multipath signal

2.4.8 Rake Receiver. Now comes some fascinating feature of WCDMA, the Rake Reception. The Rake concept uses a special wide band signal of band width W such that $W \gg 1/T_m$ where T_m is the maximum delay spread of the multipath channel. Due to spreading with PN code, the autocorrelation function of the signal achieves the characteristics desired by the Rake concept i.e; having a width of the order $1/W$ due to which an isolation of the multipath components is possible. The basic Rake receiver structure is a tapped delay line that attempts to collect the signal energy from all the signal paths that fall within the span of the delay line and to combine them optimally [1].

Now as the output of a multipath channel can be modeled by the sum of attenuated and delayed versions of the signal;

$$Z(t) = \sum_n \alpha_n z(t - \tau_n)$$

Here $\{\alpha_n\}$ are the attenuation factors and $\{\tau_n\}$ represent the path delays

In complex envelope form, we have

$$\underline{Z}(t) = \sum_n \alpha_n \underline{z}(t - \tau_n) \exp(-j2\pi f_0 \tau_n)$$

Now, if we cross correlate this signal with the local delayed code

$$\underline{Z}_k = \underline{Z}(t - k/W)$$

then any component with the path delay within $\pm 1/W$ of $\tau = k/W$, if exists will be captured due to its closeness to the Z_k and all other components

will produce zero cross correlation due to their pseudorandom character. Thus the cross-correlation of received signal with local code is:

$$\underline{Z}^* \underline{Z}_k = \sum_n \alpha_n \underline{z}^*(t - \tau_n) \exp(-j2\pi f_0 \tau_n) * Z(t - k/W)$$

which after some algebraic manipulations can be written as

$$\underline{Z}^* \underline{Z}_k = \sum_n \alpha_n \exp(-j\omega_0 \tau_n) \underline{R}_z(\tau_n - k/W)$$

Which has non-zero values only when $|\tau_n - k/W| \leq 1/W$ or in words, for a component which is aligned with the underlying code as far as the path delay is concerned.

So, in this fashion we can use different versions of the local codes and can extract almost all the components from the multipath signal hence maximizing the signal to noise ratio.

The Algorithm for implementing Rake receiver is just a mirror image of the multipath channel model. All components of the multipath signal are received through the different fingers of the rake combiner and they are integrated over the symbol interval to get the actual symbol transmitted.

2.4.8 The Viterbi Decoder. No doubt, the war against the interference, noise and jamming signals is inherent to the WCDMA communications but it was the glory of Viterbi who devised a way to decode the convolutionally encoded data by his well known algorithm called "Viterbi decoding Algorithm", and gave birth to the tremendously high encoding gain which we can get from the convolutional encoding of the data

Latter, it was proved by J.K. Omura in 1969 that the Vitebi algorithm is indeed a maximum likelihood decoding rule [1]. The first step in this algorithm is to develop the trellis diagram for the convolutional encoder used. Now, we know only the recieved sequence from which the transmitted sequence is to be found. The transmitted sequence is really the combination of the branch words obtained while traversing a branch in the trellis from given state for the given input. The essence of the algorithm is the comparison between the received sequence at time $t=t_i$. When two paths enter the same state at $t=t_i$, the one having minimum distance with respect to the received sequence is chosen and that path is called the surviving path. So, our goal is to penetrate deep and deep into the trellis to get ultimately a single path at a minimum hamming distance from the received sequence. The combination of the branch words obtained while

walking along this path gives the received sequence. A coding delay is certainly there as to decide on bit corresponding to a branch word, we have to wait until only one path emerges from a state node to which that bit is inputted. Following algorithm has been used to perform viterbi decoding in Matlab.

Input: Encoded sequence 'E' with some flipped bits.

Output: Decoded sequence with some errors recovered.

1. First of all, develop the trellis diagram. This can be visualized by finding out the following matrices:

State transition matrix: containing the next state for a given previous state and the given input.

The input matrix: containing the inputs given at a particular state of shift register.

The output matrix: containing the branch output in decimal number for given input and given state of the shift register. The indices of these matrices actually contain the real information e.g. the element of state transition matrix at index (0,1) tells that at state '00' if we input a '1', the next state will be equal in decimal to the value of the element.

2. For each level of the trellis determine the surviving paths by comparing the state-metrics (hamming distances) until we reach the depth of trellis and one final path is achieved.

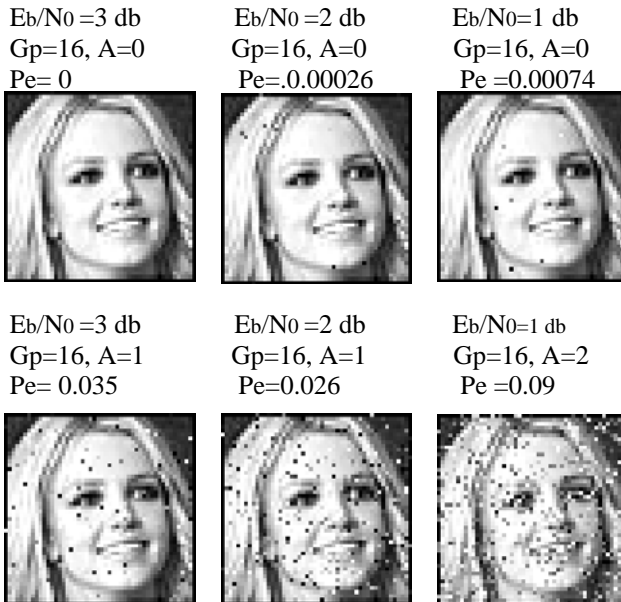
3. The decoded sequence is obtained by determining all the bits inputted to a given node in order to traverse the last surviving path.

3. Results of Simulation

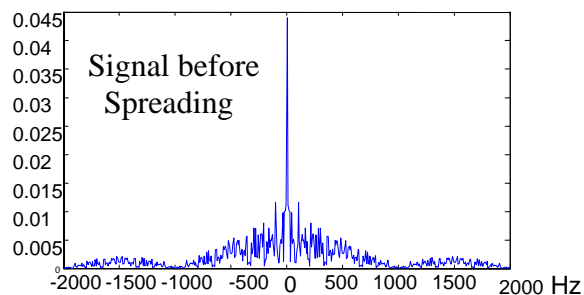
The beauty of WCDMA system is the enlarged capacity at very low level of SNR i.e; E_b/N_0 and still a tremendous fight against the multiuser interference, multipath fading and the always present thermal noise of white nature. In very simple words, with the help of convolutional encoding and viterbi decoding, a probability of error of less than 10^{-3} have been achieved at even 2-3 db with a processing gain of say 16.

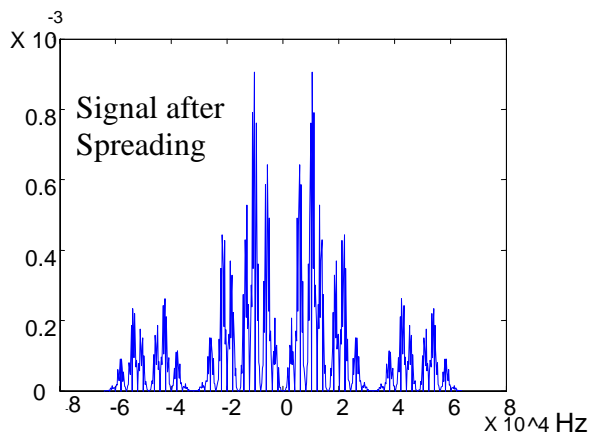
The results from image transmission simulation have been pasted to express the effects of AWGN, multicell interference and multiuser interference on an image like jpeg. Due to noise and interference, some of the pixels are disturbed and the image contains dots in it which simply correspond to corrupt pixels. The multicell interference is assumed to be sinusoidal. A sinusoidal interference of say 2 volts is about 10 times in amplitude to the amplitude of the signal with $E_b/N_0 = 2$ db and spreaded by a factor of 16.

The BER vs E_b/N_0 curves, the spirit of this simulation, have been plotted both with and without encoding to manifest the effect of coding gain at a particular operating point.



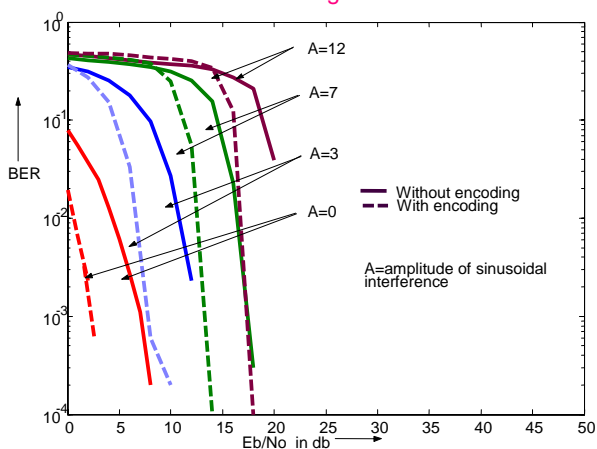
Hundreds of such experiments can be done by varying the values of the spreading factor, the SNR, the multicell interference, the multiuser interference. Only, a few of them can be shown in this limited length paper. Whatever, the values of the parameters, we get a result for BER lying on any of the following error curves. The magnitude spectra of both the signal before spreading and after spreading are also shown. The bandwidth used by the spreaded signal is given by: $W = R * G_p$, $R =$ data rate, $G_p =$ Spreading factor





The spectrums shown are for the base band signal and the dc null is obvious in the spread signal being in bipolar signaling scheme used in this simulation.

Error curves for a CDMA comm. system in the presence of AWGN and sinusoidal interference, with and without Encoding



4. References

Most of the work done in this project is related to Matlab so the readers must have some knowledge of Matlab to get full advantage from the simulation.

[1] CDMA System Engineering Handbook by Jhong

Sam Lee Leonard E. Miller.

[2] CDMA Principles of Spread Spectrum

Communication by A.J. Viterbi

[3] CDMA RF System Engineering by Samuel

C.Yang.

[4] Mobile Fading Channels by Matthias Patzold.

[5] Modern Digital and Analog Communications by

B.P.Latthi.

[6] Digital Communications by Bernard Sklar .

[7] WCDMA for UMTS by Harri Holma & Antti

Toskala.

[8] Contemporary Communication Systems using

Matlab by John G. Proakis & Masoud Salehi

[10] Tero Ojanpera and Ramjee Prasad, "An Overview

of Air Interface Multiple Access for IMT-2000/UMTS,

" *IEEE Communications Magazine*, vol. 36, pp. 88-95,

September 1998.

[11] Erik Dahlman, Bjorn Gudmundson, Matts Nilsson,

and Johan Skold, "UMTS/IMT-2000 Basedon Wide

band CDMA, " *IEEE Communications Magazine*, vol.

36, pp. 70-80, September 1998.