# Direct Sequence Spreading

Gene W. Marsh

## I. A General Description of Direct Sequence Spreading

### A. The standard view of a communication system

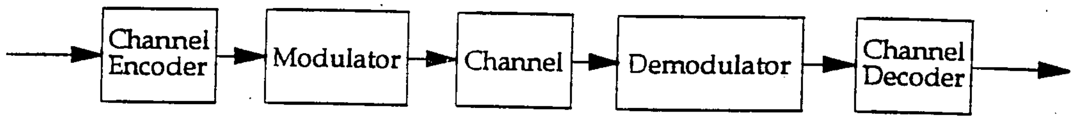


**Fig. 1.** A block diagram for a Standard Communication System.

### B. A PN-Spread Communication System

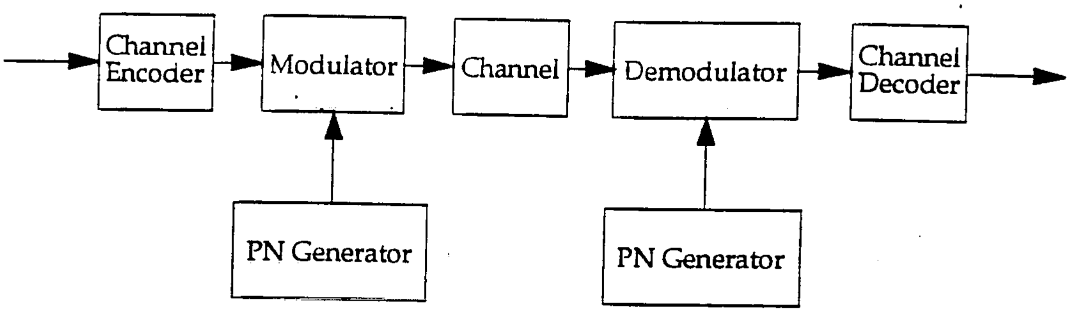


**Fig. 2.** A block diagram for a PN-spread Communication System.

1. To approach channel capacity, it is desirable to make the signal more noiselike.Therefore, we introduce a random number generator, to make the modulation appear random

### C. A closer look at the modulator/demodulator.[1]

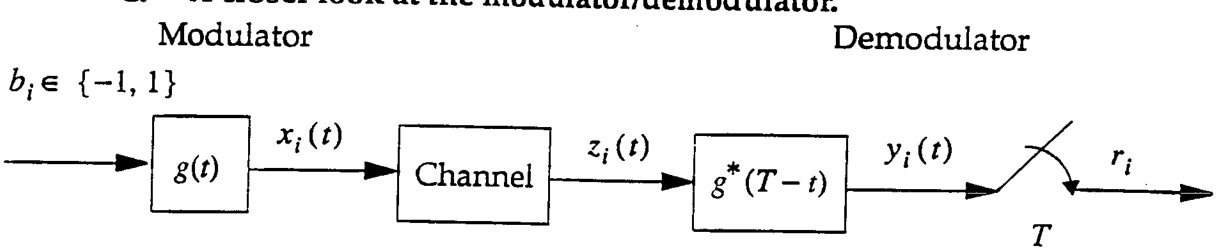


**Fig. 3.** The Standard Model

1. In the standard modulator pair shown in Figure 3, a bit determines whether the transmit filter or its inverse is emitted every $T$ seconds.

1. If you like your bits to be from the set $\{0, 1\}$, then you can replace $b_i$ and $c_i$ with $2b_i - 1$ and $2c_i - 1$, below.

The waveform then passes through the channel, and is corrupted by noise.

The resulting signal is passed through a matched filter, and sampled every $T$ seconds.

a) $T$ is known as the **bit time** and $R = \frac{1}{T}$ is known as the **bit rate.**

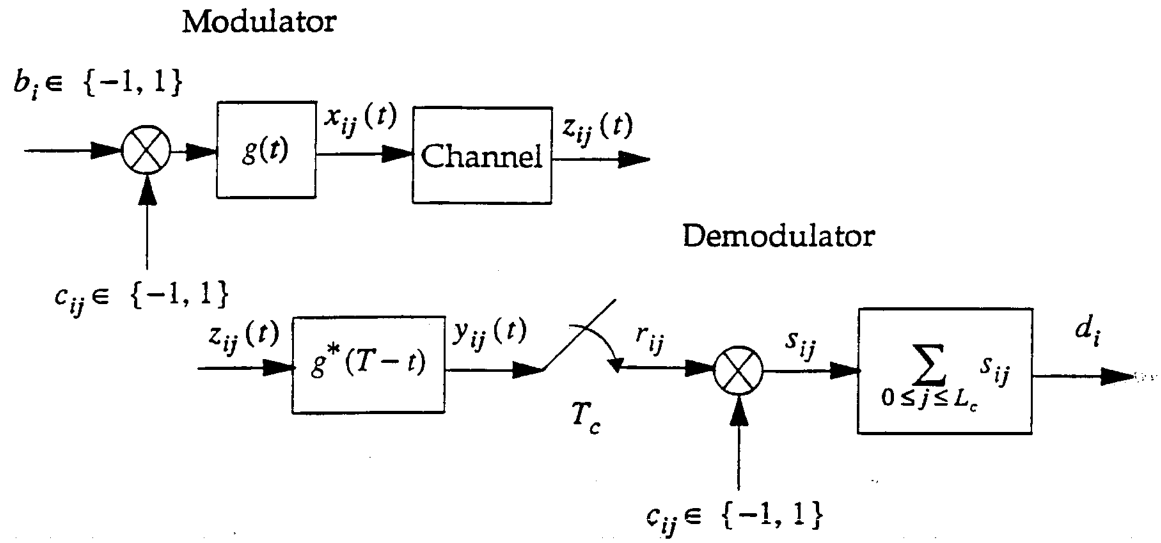


**Fig. 4.** The CDMA Model

2. In the CDMA system shown in Figure 3, a slightly different thing happens.

   A bit still enters the system every $T$ seconds, but it is now multiplied by a faster moving, random sequence every $T_c$.

   The result is sent through the channel, sampled, and the sampled signal is multiplied by the corresponding random bit. The result is then summed.

   a) Clearly things that move faster in time are wider in frequency. Hence the name "spreading".
   b) Each of the short transmitted symbols is known as a **chip**.
   c) The time $T_c$ is then known as a **chip time**, while $R_c = \frac{1}{T_c}$ is known as the **chip rate** or **spreading rate.**
   d) The spreading bits $c_{ij}$ are assumed to be i.i.d Bernoulli random variables over $\{-1, 1\}$ with parameter $\frac{1}{2}$.



L-2

e) Clearly, $L_c = \frac{T}{T_c} = \frac{R_c}{R}$ is the number of chips per bit. This is often referred to as the **bandwidth expansion factor.**

# II. Performance issues

## A. Does this little stunt cost us anything?

1. Let us examine the performance of each system assuming the channel simply corrupts the signal using zero mean, white, additive Gaussian noise $n(t)$ with variance $\sigma^2 = \frac{N_0}{2}$, which is independent of the data symbols.

a) In the standard system, we have

$$y_i(t) = b_i g(t) * g^*(T-t) + n(t) * g^*(T-t)$$

After sampling, we find that $E[d_i | b_i] = 2b_i E_b$ and

$$\sigma^2(d_i | b_i) = \frac{N_0}{2}\int_{-\infty}^{\infty}|G(f)|^2 df = \frac{N_0}{2}E_b \text{ where } E_b = \int_0^T |g(t)|^2 dt.$$

If we assume that there is no ISI, then successive samples of the noise are independent. The threshold for decisions is placed at 0. Assume zeros and ones are equally likely.

We can then calculate the probability of error as follows:

$$P(e | b_i = -1) = P(d_i > 0 | b_i = -1) = Q\left(\frac{E_b}{\sqrt{\frac{N_0}{2}E_b}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

$$P(e | b_i = 1) = P(d_i < 0 | b_i = 1) = 1 - Q\left(\frac{-E_b}{\sqrt{\frac{N_0}{2}E_b}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

$$P(e) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad (1)$$

b) In the CDMA system, for each chip, we have

$$y_{ij}(t) = b_i c_{ij} g(t) * g^*(T-t) + n(t) * g^*(T-t)$$



L-3

The $r_{ij}$ are then independent, with a Gaussian distribution such that $E[r_{ij}| b_i, c_j] = b_i c_{ij} E_c$ and

$$\sigma^2(r_{ij}| b_i, c_{ij}) = \frac{N_0}{2}\int_{-\infty}^{\infty} |G(f)|^2 df = \frac{N_0}{2}E_c \text{ where } E_c = \int_0^T |g(t)|^2 dt.$$

After the multiplier, the $s_{ij}$ clearly have mean $E[s_{ij}| b_i, c_{ij}] = b_i E_c$ and variance $\sigma^2(s_{ij}| b_i, c_{ij}) = \sigma^2(r_{ij}| b_i, c_{ij})$. In particular, we should note that the $s_{ij}$ do not depend on the $c_{ij}$ at all.

Now, if we sum the $L_c$ chips corresponding to a particular bit, we find that $d_i$ is again Normal, with conditional mean and variance given by $E[d_i| b_i] = b_i L_c E_c$ and $\sigma^2(d_i| b_i) = \frac{N_0}{2}L_c E_c$. The probability of error is therefore still

$$P(\mathrm{e}) = Q\left(\sqrt{\frac{2L_c E_c}{N_0}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad (2)$$

where $E_b = L_c E_c = \frac{R_c}{R}E_c$.

c) *Therefore, spreading has not cost us anything, except some complexity in the modulator/demodulator.*
   (1) In general, this is only true for direct spreading in coherent communications systems.
   (2) It should surprise no one that this is true, since, if you look carefully at our picture, all we have done is modify the transmit filter, and then made a matched filter for the receiver.

# III. Pseudo-Noise (PN) Sequences

## A. M-Sequences (Maximal Length Shift Register Sequences)

1. From Galois field theory, we have the notion of an M-sequence.
   a) Let us consider Galois Field 2. This is composed of the elements 0 and 1 with addition defined by "exclusive-or" and multiplication defined by "and".
   b) Let $n$ be any positive integer. Then we can define a Galois Field with $2^n$ elements by considering all possible polynomials of degree $n-1$ or less. Addition is defined by polynomial addition



L-4

modulo $x^n + 1$, and multiplication is polynomial multiplication modulo $x^n$.

c) For any such $GF(2^n)$, it is possible to find an element, $\alpha$, such that, if $\beta \neq 0 \in GF(2^n)$, then $\beta = \alpha^k$ for some $0 \leq k < 2^n$. Thus, you can cycle through all of the elements of the field by multiplying repeated multiplication or division by $\alpha$. These elements are the primitive elements of the field, and are represented by the primitive polynomials.

d) There are tables of primitive polynomials.

e) This sequence of $n$-bit numbers can be used to generate a sequence of $2^n - 1$ bits with useful properties (to be discussed later).

f) For any primitive polynomial, there are two ways to generate this bit sequence. I will illustrate this by example. Let $n = 5$. The primitive polynomial for $GF(32)$ is $p(x) = x^5 + x^2 + 1$.

(1) The "xor into the middle" method is shown in Figure 5. In

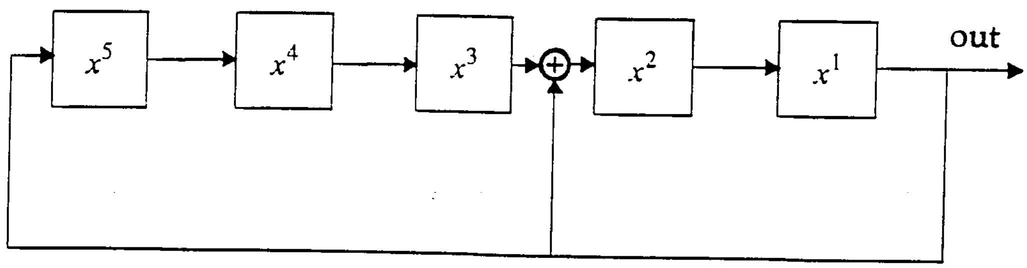


**Fig. 5.** A Galois Configuration

general, this is the easiest way to implement an M-sequence in software. The sequence produced by this machine is shown in Table 1.

Table 1: Sequence Generated by the Galois Configuration[a]

| State (Hex) | Output Bit | State (Hex) | Output Bit | State (Hex) | Output Bit | State (Hex) | Output Bit |
|---|---|---|---|---|---|---|---|
| 1f | 1 | 14 | 0 | 12 | 0 | 15 | 1 |
| 1d | 1 | a | 0 | 9 | 1 | 18 | 0 |
| 1c | 0 | 5 | 1 | 16 | 0 | c | 0 |
| e | 0 | 10 | 0 | b | 1 | 6 | 0 |
| 7 | 1 | 8 | 0 | 17 | 1 | 3 | 1 |
| 11 | 1 | 4 | 0 | 19 | 1 | 13 | 1 |
| 1a | 0 | 2 | 0 | 1e | 0 | 1b | 1 |
| d | 1 | 1 | 1 | f | 1 | 1f | 1 |

a. States are listed sequentially down the columns

(2) The "shift around" method is shown in Figure 6. This is the preferred method for implementing an M-sequence in hardware. The sequence produced by this implementation is shown in Table 2. Note that the output sequence shown here is the reverse of the one in Table 1.

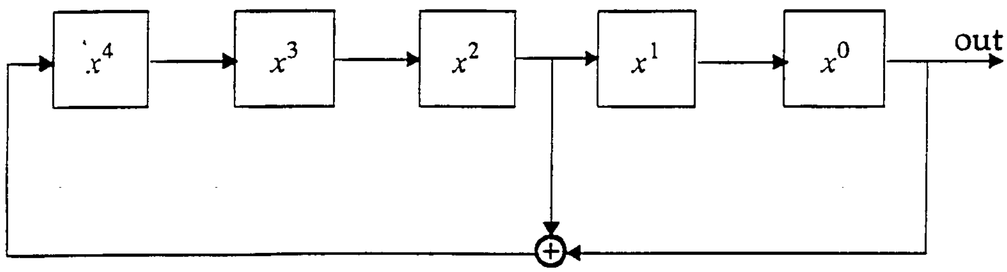


**Fig. 6.** A Fibonacci Configuration

Table 2: Sequence Generated by the Fibonacci Configuration[a]

| State (Hex) | Output Bit | State (Hex) | Output Bit | State (Hex) | Output Bit | State (Hex) | Output Bit |
|---|---|---|---|---|---|---|---|
| 1f | 1 | 1b | 1 | 2 | 0 | 1a | 0 |
| f | 1 | 1d | 1 | 1 | 1 | d | 1 |
| 7 | 1 | e | 0 | 10 | 0 | 6 | 0 |
| 3 | 1 | 17 | 1 | 8 | 0 | 13 | 1 |
| 11 | 1 | b | 1 | 4 | 0 | 19 | 1 |
| 18 | 0 | 15 | 1 | 12 | 0 | 1c | 0 |
| c | 0 | a | 0 | 9 | 1 | 1e | 0 |
| 16 | 0 | 5 | 1 | 14 | 0 | 1f | 1 |

a. States are listed sequentially down the columns.

2. Some nice properties of M-sequences.
   a) Even numbers of ones and zeros are produced.
   b) The sequence produced is relatively uncorrelated with shifts of itself.
   c) It is possible to generate the output bit in a different way.
      (1) Associate with each bit in the register a 0 or a one i.e., create a mask for the register.
      (2) To compute a bit, "and" the mask with the register. The parity of the result is the next output bit. Note: Do not destroy the contents of the register, as you need it to compute the next state.
      (3) The sequences generated in this way are shifted versions of the original sequence. Indeed, all shifts of the sequence can be generated in this fashion.
   d) Given a starting time and a clock rate, there are fast algorithms for computing the current state of the shift register.
3. Thus, M-sequences work well for generating our spreading bits. Indeed, this is the most common way to do it.

## B. Acquisition of a Spread Spectrum Signal

1. Suppose we are generating our spreading codes using M-sequences. How do we get the sequences synchronized in the transmitter and the

receiver?

a) Technically, this is known in the industry as acquiring the signal.

b) At Qualcomm, we referred to this as making the jump to hyperspace.

2. This is the hardest problem in any spread spectrum system.

3. Here is how it is done.

a) First, tie the state of the shift register to absolute time. For instance, you can declare it to have the all one's state at midnight on January 1, 1992, and that you will clock it at a 1 MHz rate.

b) Now, suppose someone is sending you a signal, and you know what mask he is using. You need a clock. Check it, and determine what the state of the shift register is at this point in time, and initialize your shift register to that value.

c) Now, you need to accumulate a sufficient number of chips. If you accumulate $N$ chips, the average value of the signal will be $NE_c$, and the variance will be $NE_c\frac{N_0}{2}$.

d) After accumulating $N$ chips, assume that your clock is off by one register position, update it accordingly, and accumulate $N$ new chips at this new time hypothesis.

e) In the end, if you have an $n$-bit shift register, you should accumulate $2^n - 1$ different hypothesis, one for each time offset. The hypothesis with the highest absolute value is declared to be the correct offset, the shift register is initialized to that value, and demodulation can begin.

f) Using the statistics of the signal, it is possible to compute the probability that you select the wrong time offset. Buy choosing a large enough value of $N$, it is possible to make this acceptably small.

g) Just in case, after declaring a hypothesis to be correct, we wait for a bit to see if the tracking loops take hold. This is a good way of detecting a bad acquisition.

# IV. Applications of Spread Spectrum

## A. Low probability of Intercept (LPI) Communications

1. If we simply take the original waveform and scale it down in time by $L_c$, then we decrease its energy by a factor of $L_c$.

$$E_c = \int_{-\infty}^{\infty} |g(L_c t)|^2 dt = \frac{1}{L_c}\int_{-\infty}^{\infty} |g(t')|^2 dt' = \frac{1}{L_c}E_b$$

After despreading, the energy has increased by a factor of $L_c$, as we saw above. Therefore, we have spread the same energy over a a wider spectrum. This makes the signal harder to distinguish from the noise background. It is used in aircraft communications so make the signal hard to detect. It would be a shame for a stealth bomber to be given away by a radio transmission.

a) This is the root of my earlier comment about making the signal more noise like.

2. This same thing can be seen by looking at the signal in time, where we have divided the signal energy up over a sequence of smaller chips.
3. In most spread systems, the signal is buried in noise.

**B. Code Division Multiple Access (CDMA)**

1. Consider a system where we have 2 signals. These signals use the same code bits, but one uses a time shifted version of the other, i.e. the second user is shifted in time by $n$ chips. Let $k = \left(i + \left\lfloor \frac{j+n}{L_c} \right\rfloor\right)$ and $l = (j+n) \bmod L_c$. The undesired user adds some interference. We are going to assume that interference is Gaussian. We then have

$$y_{ij}(t) = b_i c_{ij} g(t) * g^*(T-t) + b_k c_{kl} g(t) * g^*(T-t) + n(t) * g^*(T-t) \quad (3)$$

This will in turn give:

$$E[r_{ij} | b_i, c_j] = b_i c_{ij} E_c$$

$$\sigma^2(r_{ij} | b_i, c_{ij}) = \frac{N_0}{2}E_c + E_c^2$$

$$E[s_{ij} | b_i, c_{ij}] = b_i E_c$$

$$\sigma^2(s_{ij} | b_i, c_{ij}) = \sigma^2(r_{ij} | b_i, c_{ij})$$

$$E[d_i | b_i] = b_i L_c E_c$$

$$\sigma^2(d_i | b_i) = \frac{N_0}{2}L_c E_c + L_c E_c^2$$

$$P(e) = Q\left(\sqrt{\frac{(L_c E_c)^2}{\frac{N_0}{2} L_c E_c + L_c E_c^2}}\right) = Q\left(\sqrt{\frac{E_b}{\frac{N_0}{2} + E_c}}\right) \quad (4)$$

a) Note that we assumed that the two users were lined up in time. In general, this is not true, but it is the worst case. In that sense, the above is an upper bound on what happens.

b) We also assumed that the other user's interference was Gaussian. For a long enough spreading sequence, this is not unreasonable.

2. Other user looks like white noise to the current bit, contributing an energy equal to their chip energy. Thus, his interference is controlled.

3. It is possible to demodulate both users. We can use this to make a multiple access scheme called Code Division Multiple Access (CDMA).

   a) Let each user use either

      (1) a unique spreading sequence, different from and uncorrelated with all other users;

      (2) the same spreading pattern, delayed by some number of chips from his fellows.

   b) Every user in the system can demodulate his own signal. The other users simply raise the noise floor. For $N$ users, we can bound the probability of error as

$$P(e) \le Q\left(\sqrt{\frac{E_b}{\frac{N_0}{2} + (N-1)E_c}}\right) \quad (5)$$

   c) It is theoretically possible to approach Shannon capacity using this approach.

      (1) If I remember right, "theoretically" is the correct word, because the demodulator needs to demodulate each signal, and subtract off its effect before it demodulates the next signal.

## C. Anti-Jam (AJ) Communications

1. Jamming is the transmission of a signal in order to degrade a communication system.

2. Design of a communication system which is robust against jamming can be viewed as a game, where the transmitter tries to minimize the

bit error rate, and the jammer tries to maximize the bit error rate.

3. Often, the jammer is power limited. In this case, what the transmitter wants to do is to force him to spread his power over the widest possible bandwidth, so that he wastes as much of his power as possible.
4. Now, consider a channel with no noise, over which we send a BPSK signal. Suppose the jammer transmits a single loud tone at the carrier frequency. In a normal system, we can imagine the tone being loud enough that it would dominate our own signal, and we would demodulate that rather than the received signal. However, passing it through the despreader "modulates" that signal, making it look like wide band noise. Our own signal, on the other hand, becomes despread, making it look like a tone. The spread system performs better than the standard system in such a situation.
5. Direct sequence spreading is not the best or most common technique used for AJ modems. Most of them use non-coherent FSK and hopping rather than spreading.

## D. Position Location

1. To do time tracking for the system above, one takes 2 samples, one at $\frac{T}{4}$ and one at $\frac{3T}{4}$. Call them $s_{ij}^{e}$ and $s_{ij}^{l}$. The time tracking metric is then computed as $TM = \left(\sum_{0 \le j < L_c} s_{ij}^{e}\right)^2 - \left(\sum_{0 \le j < L_c} s_{ij}^{l}\right)^2$. If we are sampling too late, then this metric will be positive, and if we are sampling too early, then this metric will be negative.
2. If we are off by a fraction of a chip, this metric drops appreciably. Therefore, it is possible to do time tracking down to a fraction of a chip.
   a) For example, in CDMA cellular, we track to $\frac{1}{8}$ of a chip. We could do better, but this is deemed sufficient.
3. If the spreading sequence is tied to absolute time (more on this below), it is possible to use this feature to measure the time for a signal to travel from source to destination. Given several independent transmitters, one can do triangulation.
4. GPS works on a scheme similar to this.

## E. Multi-path Mitigation

1. First, realize that we only really need concern ourselves with fading on the chip level.

a) The wider your signal bandwidth, the less serious the effect of fading. This is because fades only have a certain bandwidth in the frequency domain. If your signal is wider than that bandwidth, it is unlikely that a fade will occur which will crush all of the signal energy.

b) The accepted way to combat fading is with diversity. In this case, direct sequence spreading gives you time diversity. Since a fade lasts for only a finite amount of time, it is possible that only a fraction of the chips that make up one bit will be faded.

2. Consider a case where we have 2 signals arriving with different delays, and suppose that the difference in the delays, $\tau$ is such that $T_c < \tau$. Let $\tau = nT_c + \tau_c$, $k = \left(i + \left\lfloor \frac{j+n}{L_c} \right\rfloor\right)$ and $l = (j+n) \bmod L_c$. We then have

$$y_{ij}(t) = b_i c_{ij} g(t) * g^*(T-t) + b_k c_{kl} g(t+\tau) * g^*(T-t) + n(t) * g^*(T-t) \qquad (6)$$

This will in turn give:

$$E[r_{ij} | b_i, c_j] = b_i c_{ij} E_c$$

$$\sigma^2(r_{ij} | b_i, c_{ij}) = \frac{N_0}{2} E_c + E'^2$$

$$E[s_{ij} | b_i, c_{ij}] = b_i E_c$$

$$\sigma^2(s_{ij} | b_i, c_{ij}) = \sigma^2(r_{ij} | b_i, c_{ij})$$

$$E[d_i | b_i] = b_i L_c E_c$$

$$\sigma^2(d_i | b_i) = \frac{N_0}{2} L_c E_c + L_c E'^2$$

$$P(e) = Q\left(\sqrt{\frac{(L_c E_c)^2}{\frac{N_0}{2} L_c E_c + L_c E'^2}}\right)$$

Now, we know that $E' \le E_c$, so that

$$P(\mathrm{e}) \leq Q\left(\sqrt{\frac{L_c E_c}{\frac{N_0}{2} + E_c}}\right) = Q\left(\sqrt{\frac{E_b}{\frac{N_0}{2} + E_c}}\right) \tag{7}$$

3. What is the significance of this?
   a) Any signal arriving at a time offset greater than one chip looks like white noise to the current bit. Thus, ISI is controlled.
   b) It is possible to demodulate the late path as well. This helps improve our performance. Thus, the ISI can help you. This is a benefit a narrowband system cannot provide.

# V. CDMA Cellular

## A. Capacity of the CDMA System[2]

1. The capacity of the system is determined by the capacity of the mobile-to-cell link.
   a) This is principally because the mobiles must use smaller transmitters, and cannot synchronize their signals.
2. Let us begin by realizing that the capacity of the system is controlled by the signal to noise ratio needed to achieve an acceptable link.
   a) Since we have a wideband system, it is possible to use powerful coding techniques with little penalty. Because of this, the CDMA system only requires $\left.\left(\frac{E_b}{N_0}\right)\right|_{\text{desired}} = 7\text{ dB}$.
3. Assume the following:
   a) The system is interference limited, i.e. that the noise from other users in our own cell is much greater than the background thermal noise.
   b) Assume that we only care about users in our own cell.
   c) All signals are power controlled so that the reach the cell with equal power.[3]
4. Under these assumptions, we see from (5) that the signal-to-noise ratio is just $\frac{E_b}{(N-1)E_c} = \frac{R_c}{R(N-1)}$.

2. This is a "back-of-the-envelope" approximation. It is relatively easy to understand, and gives approximate results. Actual capacity is determined through simulation and testing.
3. This is crucial for achieving good capacity in the CDMA system.

a) We need this to be equal to $\left.\left(\frac{E_b}{N_0}\right)\right|_{\text{desired}}$.

b) Solving gives $N \approx \frac{R_c}{R} \cdot \frac{1}{\left(\frac{E_b}{N_0}\right)_{\text{desired}}}$.

5. The assumption that we are limited only by the users in our own cell is too generous. We can mitigate this by appropriate scaling.
   a) With an omnidirectional antenna, the number of users is decreased by about a factor of $F = 0.6$.
   b) In the actual system, the cells use sectorized antennas. Each sector has a field of view of 120°. Due to overlap between the sectors, this only buys back a factor of $G = 2.55$.
   c) Note that both of these numbers are empirical.
6. In a CDMA system, it is possible to have a variable rate transmission, so that one does not transmit when there is no data.
   a) Speech does not occur 100% of the time in a conversation.
   b) The amount of dead time varies with the language spoken. For English, the empirical number is $d = 0.4$.
   c) By using a variable rate vocoder, and not transmitting during the silent times, we get to take advantage of this, and our signal to noise ratio increases appropriately.
7. Therefore, the total capacity of the system is approximately

$$N \approx \frac{R_c}{R} \cdot \frac{1}{\left.\left(\frac{E_b}{N_0}\right)\right|_{\text{desired}}} \cdot \frac{1}{d} \cdot F \cdot G \tag{8}$$

8. For the CDMA system, $R = 9600$ Hz and $R_c = 1.2288$ MHz.
9. Using the numbers above, this gives us about 98 CDMA channels in a 1.25MHz bandwidth.
10. After blocking is considered, this yields about a factor of 20 increase in the number of calls per cell.
    a) This estimate varies considerably depending on who is doing it, as most of the data is empirical.

### B. Power Control and the Near/Far Problem

1. One of the principle assumptions made in the capacity calculation

above was that all mobiles reach the cell at the same power level. This assumption is absolutely essential to operation of the system.

2. As mobiles are spread out over the entire cell, this is somewhat problematical, to say the least.
3. In order to combat this, we had to institute a method of power control. Very simply, it works as follows.
   a) the mobile and the cell both monitor their average received signal-to-noise ratio (SNR). The mobile communicates his received SNR to the cell.
   b) The cell adjusts his output to provide an acceptable SNR to the cell.
   c) If the SNR received at the cell is too low, the cell tells the mobile to increase his power.
   d) If all cells controlling the mobile tell him to increase his power, he does. Otherwise, he decreases his power.
   e) In this way, as a mobile leaves one cell, he will begin to pick up another. Both cells will tell him to increase his power, so that both can hear.
   f) As he moves into the new cell, it will tell him to decrease his output, and he will. Eventually, the old cell can no longer hear him.
   g) This is the basic mechanism for a handoff, but the reality is much more complicated.

### C. A Final Note

1. There is a lot more that can be said about CDMA cellular. Due to time pressure, I will stop here. [4] provides a good general description of what CDMA cellular is and how it works.

## VI. References

[1] John G. Proakis. *Digital Communications.* McGraw-Hill Book Company, New York, 1983.

[2] Marvin K. Simon, Jim K. Omura, Robert A. Sholtz, Barry K. Levitt. *Spread Spectrum Communications, Volume 1.* Computer Science Press, Rockville, Maryland, 1985.

[3] Richard E. Blahut. *Theory and Practice of Error Control Codes.* Addison-Wesley Publishing Company, Reading, Massachusetts, 1983

[4] *An Overview of the Application of Code Division Multiple Access (CDMA) to Digital Cellular Systems and Personal Cellular Networks.* Document Number EX60-10010. Qualcomm, Inc., San Diego, CA, May 21, 1992.