MASTER'S THESIS

Physical-layer Performance Evaluation of a High-performance LAN- LAN Interconnection via Satellite

by R.S. Maust
Advisor: E.L. Walker

CSHCN M.S. 93-1
(ISR M.S. 93-25)

Physical-layer Performance Evaluation of a High-performance LAN-LAN Interconnection via Satellite

Reid S. Maust

Department of Electrical and Computer Engineering

West Virginia University

December 6, 1993

Dr. Ernest L. Walker, Research Advisor

This thesis is submitted in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering at West Virginia University

Contents

List of Figures

List of Tables

ABSTRACT

With the proliferation of computer networks comes the need for better internetworking techniques for more efficient communication. A Ka-band satellite link between two high-performance local-area networks (LANs) is described. The LANs are assumed to use the High-performance Parallel Interface (HIPPI), which is a parallel point-to-point interface affording data rate near 1 Gbit/s. In order to maintain the parallel structure, the data lines share the satellite channel by code-division multiple access (CDMA). Two CDMA coding schemes are examined, namely Gold codes and Barker codes. Since HIPPI is a synchronous system, a modification to each of the CDMA codes is described to improve performance. For each of the CDMA codes, two modulation schemes, multiple-amplitude minimum-shift-keying (MAMSK) and continuous-phase frequency-shift-keying (CPFSK) are modeled. In order to obtain acceptable bit-error-rate (BER) performance, equalization of the channel to compensate for the satellite nonlinearities is described. The satellite channel is simulated for each of the four possibilities of CDMA code and modulation scheme in order to predict BER performance. It is found that MAMSK outperforms CPFSK for a low signal-to-noise ratio, but CPFSK outperforms MAMSK for a high signal-to-noise ratio.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Ernest L. Walker, for his valuable advice throughout this project. I would also like to thank my other committee members, Dr. Stuart K. Tewksbury and Dr. Craig S. Sims, for their comments concerning this thesis. Finally, I would like to thank Dr. Jon C. Freeman from NASA's Lewis Research Center for providing data used in the modeling of the ACTS satellite channel.

Chapter 1

INTRODUCTION

The possibility of using a wideband satellite to interconnect two high-performance local-area networks (LANs) is investigated. The satellite channel is simulated to compare the performance of various combinations of modulation and channel-coding techniques. The LANs are assumed to use the High-performance Parallel Interface (HIPPI), which is a parallel point-to-point interface affording a high data rate (800 Mb/s). In order to maintain the parallel structure of the data lines and avoid converting the parallel data lines into a serial bit-stream, the data lines share the satellite channel by code-division multiple access (CDMA), which permits all the data lines to transmit simultaneously and use the entire available bandwidth of the Advanced Communications Technology Satellite (ACTS), operating in its high-data-rate Microwave Switch Matrix (MSM) mode.

Two CDMA codes, Gold codes and Barker codes, are examined. Since HIPPI is a synchronous system, a modification to each of the CDMA codes is described to improve performance by reducing the mutual interference among the data lines. For each of the CDMA codes, two modulation schemes, multiple-amplitude minimum-shift-keying (MAMSK) and continuous-phase frequency-shift-keying (CPFSK) are examined. Since both Gold and Barker codes require a channel which is at least approximately linear, equalization of the channel to compensate for the satellite nonlinearities is described.

The 900-MHz ACTS satellite channel is simulated with a variety of signal-to-noise ratios for each of the four possibilities of CDMA code and modulation scheme in order to predict BER performance. It is found that MAMSK outperforms CPFSK for a low signal-to-noise ratio, but CPFSK outperforms MAMSK for a high signal-to-noise ratio.

The remainder of this thesis is divided into six additional chapters and an appendix. Chapter 2 is a brief overview of some literature pertinent to this investigation. Chapter 3 is the statement of the problem. Chapter 4, the Theoretical Approach, expands on Chapters 2 and 3 by giving the details of the pertinent theory. Chapter 5 is the description of the simulation model used. Chapter 6 discusses the results of the simulation. Chapter 7 lists the conclusions drawn from the simulation results. Finally, the appendix provides source-code listings for the simulation model, which was written in the language of the MathWorks' *MatLab* software.

Chapter 2

LITERATURE REVIEW

This literature review gives an overview of the internetworking connection being investigated. The physical-layer issues concerning this interconnection are then outlined. A more detailed analysis is given in the chapter describing the Theoretical Approach.

2.1 SYSTEM OVERVIEW

With the networking of high-performance workstations comes the need for high-speed communication in order to transfer large amounts of data quickly. Large, fast data transfers are necessary for such tasks as copying a large database or transmitting graphics or animation in real time. In the networking configuration considered here, two LANs of workstations are connected via a Ka-band (30 GHz uplink, 20 GHz downlink) satellite, as shown in Figure 2.1. The LANs are assumed to use the High-performance Parallel Interface (HIPPI), which is a parallel point-to-point interface affording a high data rate (800 Mb/s) [1]. The satellite to be used is NASA's Advanced Communications Technology Satellite (ACTS), which supports an RF bandwidth of 900 MHz [2]. Although HIPPI currently provides for a fiber-optic link [1], to the knowledge of the author, no attempt to provide a HIPPI-compatible satellite link has been previously reported.

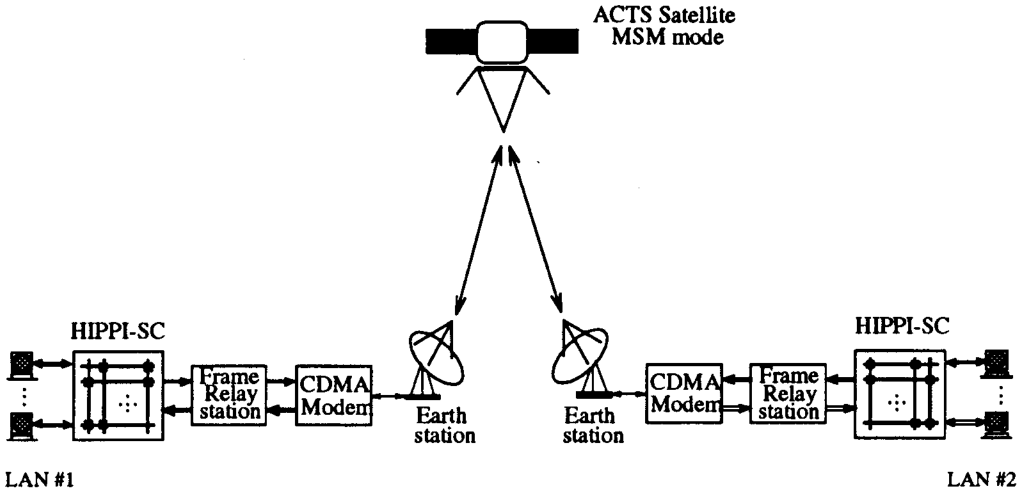


Figure 2.1: Block Diagram of Network Configuration

2.2 HIPPI OVERVIEW

HIPPI (the High-performance Parallel Interface) is a parallel point-to-point interface which has an option to support either 800 Mbits/s or 1600 Mbits/s. The clock rate is 25 MHz for both options, which differ only in that they use either 32 or 64 parallel lines to transmit data, corresponding to a data rate of 800 Mb/s or 1600 Mb/s, respectively [1]. This investigation assumes the 800 Mbits/s option. In addition to the 32 data lines, the clock line and several control lines are also transmitted in parallel to the data. A description of the control lines is given in the tutorial paper [1].

2.3 ACTS OVERVIEW

NASA's ACTS (Advanced Communications Technology Satellite) is a wideband satellite supporting a bandwidth of 900 MHz [2]. ACTS is designed to demonstrate new satellite communication technology. Notably, ACTS operates in the relatively unused Ka frequency band (30 GHz uplink, 20 GHz downlink), where rain attenuation is much more severe than at more commonly used lower frequencies. To compensate for this attenuation, ACTS provides two beacons which can be used to measure the attenuation in the link [2], which can be used by the receiver for compensation. ACTS is simulated in its MSM (Microwave Switch Matrix) mode, in which there is no onboard processing of data; data are only amplified and repeated.

Despite its advantages over more conventional satellites, ACTS, like conventional satellites, introduces nonlinearities into the channel, since it uses a traveling-wave tube amplifier (TWTA). The nonlinearities of interest here are AM-AM (amplitude modulation-amplitude modulation), in which the amplification varies with the input envelope amplitude, and AM-PM (amplitude modulation-phase modulation), in which phase distortion varies according to the input envelope amplitude. Actual experimental data from NASA [3] are used in the modeling of the ACTS AM-AM distortion.

2.4 PHYSICAL LAYER INTERNETWORKING ISSUES

To the knowledge of the author, no other work has been published concerning the interconnection of two HIPPI LANs via a Ka-band satellite. This internetworking raises the issues of propagation over the channel, modulation and coding, and of synchronization and error handling. The major propagation issues are free-space loss, noise, rain attenuation, and nonlinear distortion. Two modulation schemes, multiple-amplitude minimum-shift keying (MAMSK) and continuous-phase phase-shift keying (CPFSK), and two coding schemes, Gold and Barker, are considered. Since this investigation is focused on the physical-layer issues of the interconnection, the medium-access-control-layer issues of synchronization and error handling are beyond the scope of this investigation.

The propagation issues of free-space loss, noise, rain attenuation, and nonlinear distortion affect the performance of the communications link. The free-space loss, which depends only on the carrier frequency and the distance between the transmitting and receiving antennas [4], is constant in this investigation, since the distance and carrier frequencies of both the uplink and downlink are constant. The satellite channel noise is additive, white, Gaussian noise [4]. In the Ka frequency band, rain attenuation is more severe than at lower frequencies and cannot be neglected [5]. Nonlinear distortion results from the ACTS satellite's traveling-wave-tube amplifier, which causes both amplitude and phase distortion [3, 6].

The combination of modulation and coding used must use bandwidth efficiently and must have as little amplitude variation as possible. The modulation must be able to transmit 800 Mb/s data within the 900 MHz ACTS bandwidth [2]. Since the ACTS nonlinearities depend on the amplitude of the transmitted envelope [3, 7], keeping a constant-amplitude envelope would hold the ACTS nonlinearities to a constant value, facilitating the cancellation of the nonlinear effects. Although keeping the envelope amplitude constant is not a requirement, increasing the amount of variation in the amplitude requires more sophisticated equalization of the nonlinearities. The coding used in this investigation permits the HIPPI lines to share the frequency band using a spread-spectrum technique known as code-division multiple-access (CDMA) [7]. Two CDMA codes with little cross-talk, Gold codes [7] and Barker codes [8], are examined. The amount of bandwidth required depends on both the modulation and the coding employed. For either Gold or Barker codes, both the MAMSK [9] and CPFSK [7] modulation schemes can transmit the data within the allotted bandwidth and have amplitudes that, while not constant, have small enough variation for equalization still to be feasible.

Chapter 3

PROBLEM STATEMENT

The problem to be solved is that of demonstrating the feasibility of interconnecting two HIPPI LANs over a wideband ACTS satellite channel, from a physical-layer point of view. It is necessary to design appropriate techniques for modulation and coding, compensation for channel nonlinearities and rain attenuation, and measuring the performance of the channel.

The modulation and coding techniques are constrained to fit within the available bandwidth of the ACTS satellite. Two types of modulation are to be examined, MAMSK [9] and CPFSK [7]. For each modulation technique, two CDMA codes, Gold [7] and Barker [8], are to be compared.

An appropriate equalization technique must be developed to compensate for the effects of the channel nonlinearities [3], which are caused by the amplifier onboard ACTS. In addition, equalization is also required to compensate for rain attenuation, which is severe in the frequency band used by ACTS [5].

The performance measure to be used is the bit-error-rate of the channel, which is to be examined for all four possibilities of CDMA code and modulation technique and under various levels of noise in the channel. In order to predict the bit-error-rate of the channel under these conditions, a simulation model of the channel is to be employed.

Chapter 4

THEORETICAL APPROACH

A block diagram giving an overview of the system is shown in Figure 4.1. The theoretical representation of the satellite channel is discussed in Sections 4.1 through 4.3, which describe propagation, the ACTS channel, and the equalization of the ACTS nonlinearities, respectively.

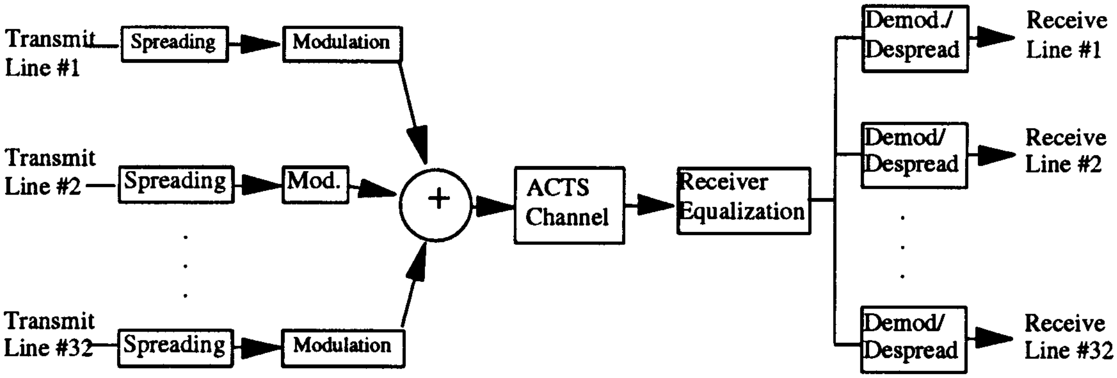


Figure 4.1: Block Diagram of Overall Interconnection System

Section 4.4 discusses the two modulation techniques examined here. The generation of the modulating waveforms and the modulation techniques' performance in the nonlinear channel are both described. The demodulation techniques employed here are discussed in Section 4.5.

Section 4.6 discusses the underlying theory behind code-division multiple-access and describes how to generate two CDMA codes, Gold and Barker.

Finally, Section 4.7 ties the previous sections together in describing the modulation and coding issues specifically relating to the HIPPI architecture.

4.1 SATELLITE PROPAGATION

Two important transmission impediments for any radio link are Gaussian noise and free-space loss (i.e., the attenuation due only to the distance between the transmitter and the receiver). In addition to these factors, rain attenuation is a severe degradation for the Ka-band carriers (30 GHz uplink, 20 GHz downlink) of interest here. The free-space loss is given by [4]:

$$L_{FS} = 20 \log \left(\frac{4\pi d}{\lambda} \right) \text{ dB} \tag{4.1}$$

where d is the distance between the transmitting and receiving antennas and λ is the wavelength corresponding to the carrier frequency.

The channel's rain attenuation is given in [5] as

$$L_{rain} = a(f) \cdot R^{b(f)} \frac{L}{\sin \theta} \text{ dB} \tag{4.2}$$

where, for the frequencies of interest here,

$$a(f) \approx (4.21 \times 10^{-5})(f)^{2.42}, \quad 2.9 \leq f \leq 54 \text{ GHz} \tag{4.3}$$

$$b(f) \approx \begin{cases} 1.41(f)^{-0.0779} & 8.5 \leq f \leq 25 \text{ GHz} \\ 2.63(f)^{-0.272} & 25 \leq f \leq 164 \text{ GHz} \end{cases} \tag{4.4}$$

$$L \approx 4 \text{ km} \tag{4.5}$$

and $\theta =$ the elevation angle of the satellite.

4.2 THE ACTS CHANNEL

NASA's Advanced Communications Technology Satellite (ACTS) is an experimental wideband satellite used to study the feasibility of communication on the Ka frequency band (30 GHz uplink, 20

GHz downlink) [2]. Using the Ka band avoids the congestion of lower-frequency bands, but it also exposes the transmissions to more rain attenuation than at lower frequencies. The bandwidth of ACTS is 900 MHz [10]. A straight-line approximation of the single-sided ACTS frequency response is given in Figure 4.2 [3].

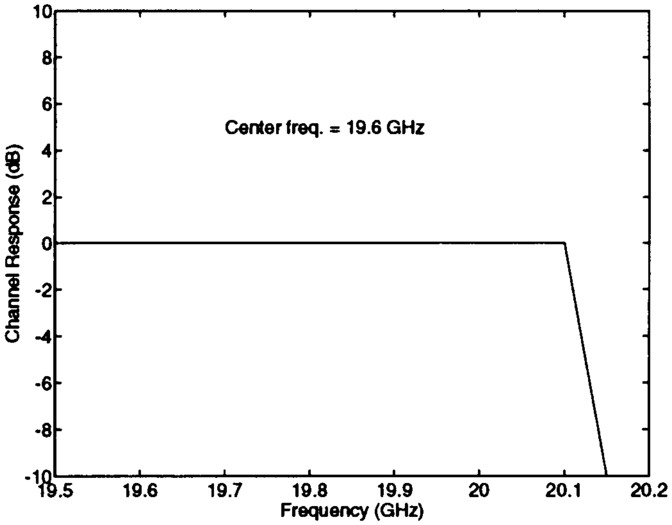


Figure 4.2: ACTS Frequency Response

The ACTS satellite is modeled as a noiseless frequency down-converter corrupted by nonlinearities. The ACTS nonlinearities result from the effects of the traveling-wave tube amplifier (TWTA) onboard ACTS. The TWTA causes both amplitude saturation (AM-AM Distortion) and phase distortion (AM-PM Distortion) of the transmitted signal. The measured saturation curve (output vs. input power, in decibel milliwatts) for ACTS is shown in Figure 4.3, from NASA [3].

As shown in Figure 4.4, the phase nonlinearity is approximated as a linear function of input amplitude, expressed in decibels, which is typical for a TWTA [6].

4.3 EQUALIZATION

Equalization to compensate for the ACTS nonlinearities, is performed in two steps: amplitude correction and phase correction. To perform amplitude equalization, a third-order polynomial ap-

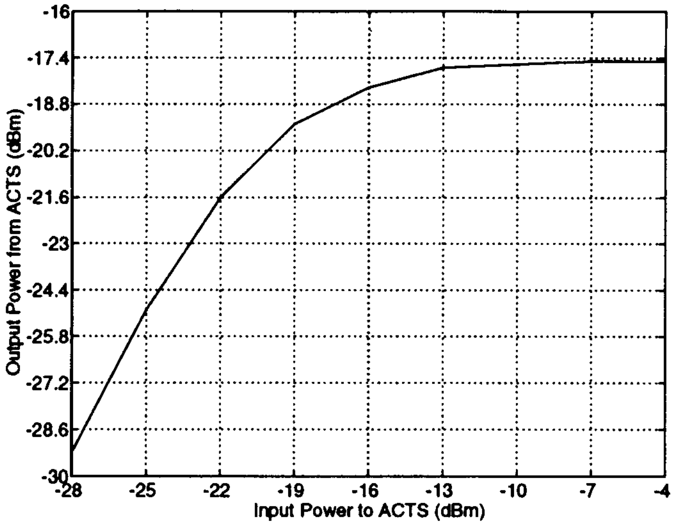


Figure 4.3: ACTS AM-AM Distortion

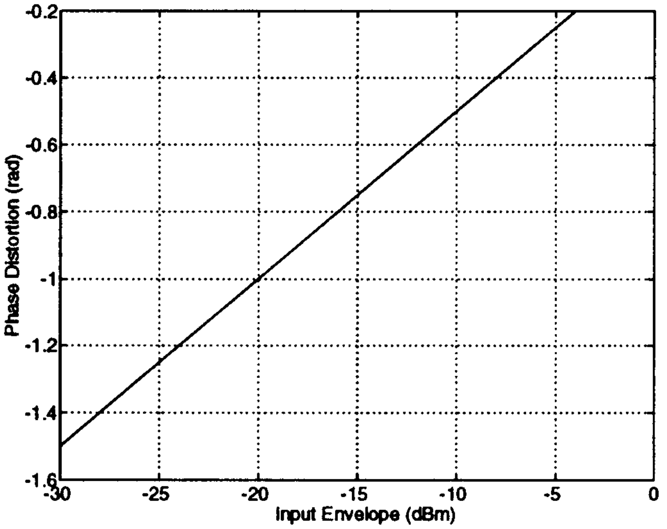


Figure 4.4: ACTS AM-PM Distortion (Approximated)

proximation of the inverse of the AM-AM distortion curve (Figure 4.3) is found. In Figure 4.5, both the ideal inverse of the distortion curve (dotted line) and the cubic polynomial approximation of the inverse (solid line) are shown. Note that the ideal equalizer (dotted line) is not physically realizable, since there is more than one equalized amplitude corresponding to the received amplitude of -18 dBm. On the other hand, the polynomial approximation is realizable, since each received amplitude corresponds to only one equalized amplitude. The equalizing polynomial is evaluated at the value of the received envelope amplitude, resulting in the corrected amplitude.

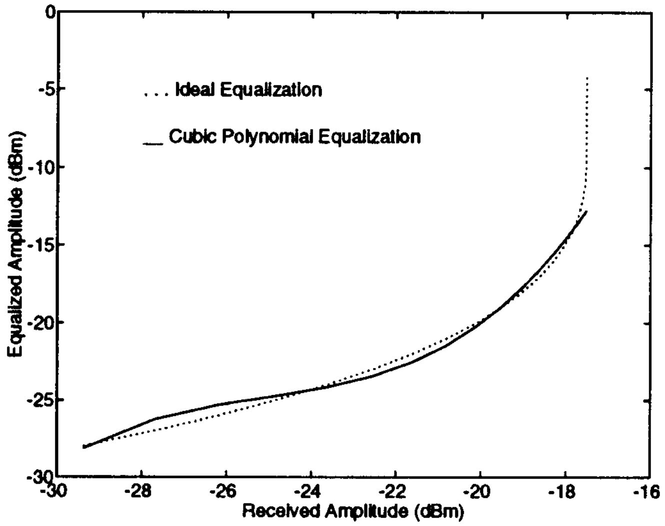


Figure 4.5: ACTS Amplitude Equalization

The performance of the amplitude equalization in the absence of noise is shown in Figure 4.6. The solid line represents the equalized amplitude using the polynomial described above. The dashed line shows the curve for perfect equalization (i.e., the equalized envelope exactly equals the transmitted envelope). For received amplitudes less than 10 dBm, the two curves are indistinguishable. For amplitudes larger than 10 dBm, the satellite limiting is so severe that the equalization cannot overcome it.

The inverse of the AM-PM distortion curve (Figure 4.4) a straight line. The AM-PM inverse function is evaluated at the value of the *corrected* amplitude, resulting in the phase correction needed. The corrected amplitude and phase are the amplitude and phase of the receiver's estimate of the

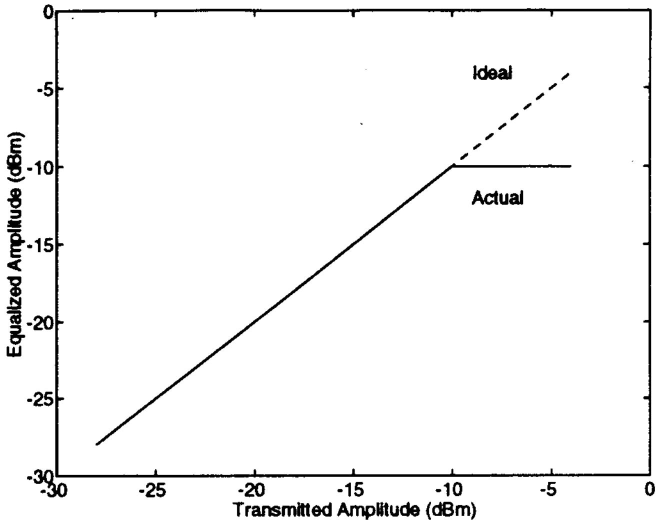


Figure 4.6: ACTS Amplitude Equalization Performance in the Absence of Noise

transmitted envelope.

4.4 MODULATION

Two modulation schemes, MAMSK and CPFSK, are compared in this investigation. The next two sections describe the modulation schemes. The performance of both modulation schemes in a nonlinear channel is then described in Section 4.4.3. Both modulation schemes are assumed to operate on two input bits at a time for each data line. Following common practice in modulation, a binary digit "1" is represented as the value +1, but a binary digit "0" is represented as the value -1. The input bit-pairs are Gray coded and mapped to one of the four symbols -3, -1, +1, or +3, as shown in Table 4.1.

The modulation schemes only differ in how the four input symbol values (-3, -1, +1, and +3) are mapped onto the radio carrier.

Since approximately 99% of the signal power is contained within the transmission bandwidth of ACTS for both CPFSK and MAMSK modulation, the effects of intersymbol interference are neglected (following [9]) to simplify the simulation.

Bit pair		Symbol
-1	-1	-3
-1	1	-1
1	1	+1
1	-1	+3

Table 4.1: Gray Coding of Bit-pairs

4.4.1 CPFSK MODULATION

One modulation technique examined here is a variant of continuous-phase frequency-shift-keying (CPFSK) with modulation index 0.25 [7], which was chosen because of its bandwidth efficiency. Approximately 99% of the signal power is concentrated in a two-sided bandwidth of 0.6 times the data rate [6]. A block diagram of this CPFSK modulator is shown in Figure 4.7. The angle generator block in Figure 4.7 converts the data value into a complex envelope with amplitude 1 and an angle according to Table 4.2.

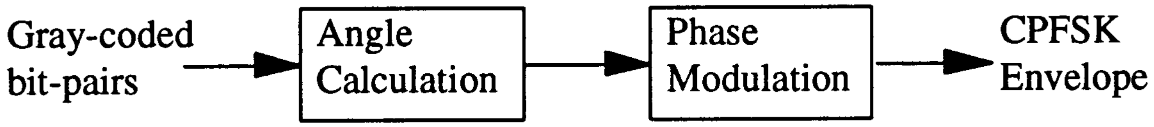


Figure 4.7: CPFSK Modulator Block Diagram

This differs slightly from ordinary CPFSK in that ordinary CPFSK maps the input values to a phase *difference*, instead of an absolute phase. For example, an ordinary CPFSK modulator would interpret a value of +1 as an instruction to add 45 degrees to whatever the current phase happened to be. On the other hand, the modulator used here would interpret a value of +1 as an instruction to change the phase to 45 degrees, regardless of its previous value. This modification was necessary to facilitate correlation detection of the spread-spectrum codes. The complex-envelope constellation

Input symbol	Angle (degrees)
-3	-45
-1	-135
+1	45
+3	135

Table 4.2: Determination of CPFSK Angle

for a CPFSK transmission is shown in Figure 4.8.

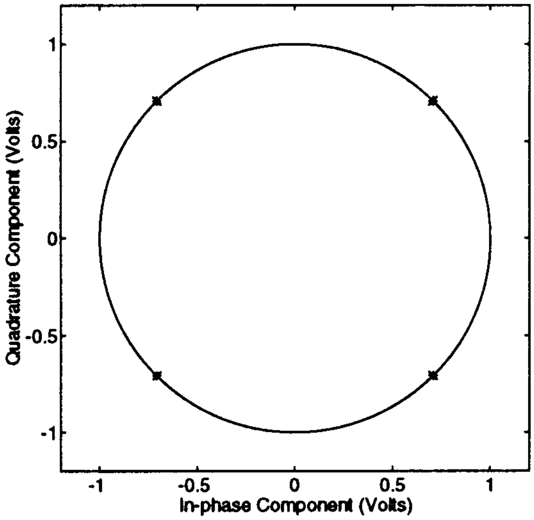


Figure 4.8: CPFSK Envelope

Because vector addition is commutative and associative, the data lines can be separated into two disjoint sets for computational ease: those transmitting either +1 or -1 (envelope angles of 45 or -135 degrees) and those transmitting either +3 or -3 (envelope angles of 135 or -45 degrees). The input values in each of these sets are spread separately using either Barker or Gold codes. Note that, since the spreading codes multiply the value by either +1 or -1, all vectors within each of these sets remain parallel after spreading. Thus, the amplitude of the sum of all vectors within one of these sets is simply the scalar sum of the data values. This sum, which is scalar, is then multiplied by a unit vector with angle of either 45 or 135 degrees, corresponding to the first and second groups defined above. The vector sum of the second set (transmitting +3 or -3) must be divided by 3 to normalize the amplitude, since 3 and -3 are coded to vectors with magnitude 1, not 3. The receiver will multiply this vector sum by 3 to undo this normalization. The two vector sums are then added, using vector addition, to find the envelope value at the given time. This envelope vector can take on a finite number of values, with both of its components constrained to be integer multiples of $\sqrt{2}$.

4.4.2 MAMSK MODULATION

Another modulation scheme examined here is multiple-amplitude minimum-shift keying (MAMSK). Unlike standard MSK, more than two bits can be transmitted per symbol by using multiple amplitude levels on each of the quadrature carriers. In this model, 16-MAMSK is used to transmit four bits per symbol, reducing the required bandwidth to half that of standard MSK [9]. The MAMSK modulator functions exactly like an standard MSK modulator, except that input symbols convey amplitude as well as phase information. Unlike standard MSK, where the input symbols to each of the quadrature carriers can be either +1 or -1, 16-MAMSK's input symbols to each carrier represent two bits of information and can be either +3, +1, -1, or -3. The block diagram for this MAMSK modulator is shown in Figure 4.9.

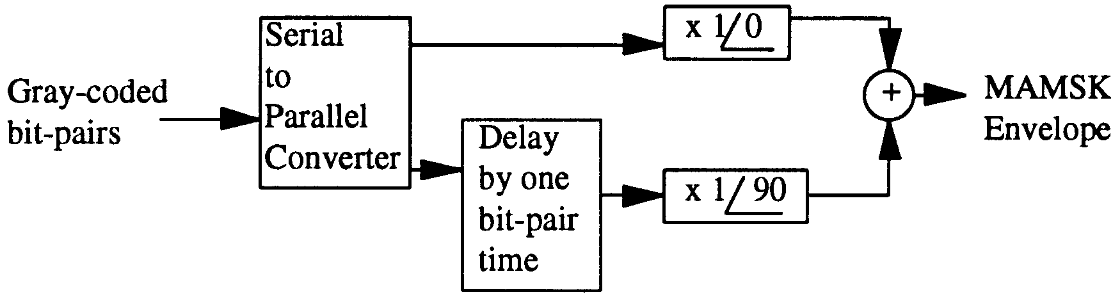


Figure 4.9: MAMSK Modulator Block Diagram

Because of the linearity of the MSK modulator, the amplitude of the complex envelope at the optimal sample instant is proportional to that of the input word, resulting in a multiple-amplitude complex envelope for MAMSK. The MAMSK envelope is shown in Figure 4.10.

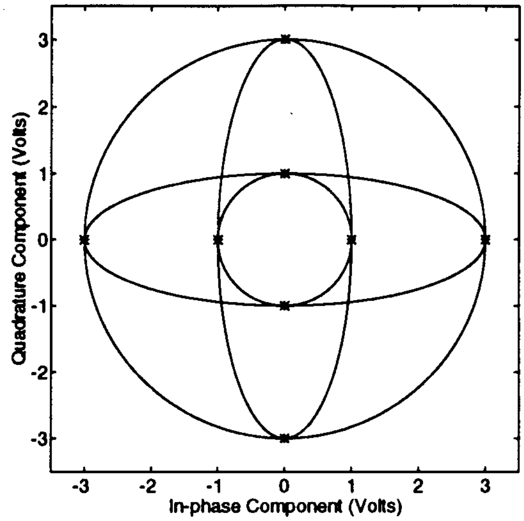


Figure 4.10: MAMSK Envelope

4.4.3 NONLINEAR-CHANNEL MODULATION PERFORMANCE

Since the ACTS nonlinearities (both amplitude and phase) are functions of the input envelope's amplitude, a signal with as small a deviation as possible in envelope amplitude is desired. Ideally, modulation having a constant-amplitude envelope should be used so that the nonlinearities' effects become constant and thus can be easily neutralized. However, even if each data line sharing the channel uses a constant-amplitude modulation scheme, the total transmitted envelope does not have a constant amplitude after the envelopes of all the data lines are added together (as a result of the CDMA users' transmitting simultaneously). Fortunately, it is observed that smaller amplitudes are much more likely than larger amplitudes for both MAMSK and CPFSK modulation.

CPFSK PERFORMANCE

Although CPFSK is a constant-amplitude modulation scheme, the overall transmitted envelope is not constant-amplitude. However, smaller amplitudes have a higher probability than larger amplitudes. For example, for 32 data lines, the maximum possible amplitude of the envelope would be 32 and would occur when all 32 lines were transmitting the same input symbol. For example, assume that all 32 lines were transmitting a +1, resulting in an total envelope with amplitude 32 at an angle of 45 degrees. The probability of this event, assuming independent and equally probable source symbols for each line, is

$$\begin{aligned}
\text{P(32 lines transmit +1)} &= [\text{P(one line transmits +1)}]^{32} && (4.6)\\
&= \left(\frac{1}{4}\right)^{32} && (4.7)\\
&= 2^{-64} && (4.8)\\
\text{P(32 lines transmit +1)} &= 54.21 \times 10^{-21} && (4.9)
\end{aligned}$$

Since the source symbols are independent, the probability of all 32 lines transmitting the same source symbol is 4 times this result, since there are 4 possible source symbols. Thus,

$$\text{P(32 lines the same)} = 2^{-62} = 216.8 \times 10^{-21} \qquad (4.10)$$

By comparison, the bit-error-rate of a satellite is typically on the order of 10^{-7} at best. Therefore, the probability of all the input lines transmitting the same value is negligible compared to the BER. It is asserted that all reasonably likely amplitudes lie below some threshold, which is significantly less than the maximum value of 32. A histogram of the amplitude values for a simulation run is given in Figure 6.4 in the Results chapter.

MAMSK PERFORMANCE

Since the MAMSK amplitude can assume a value of either 1 or 3, MAMSK is not a constant-envelope modulation. If all 32 data lines are transmitting either +3 or -3, the overall envelope has its maximum amplitude of 96 (32 × 3). Assuming independent, equally likely source symbols and proceeding as before,

$$\begin{aligned}
\text{P(32 lines transmit +3)} &= [\text{P(one line transmits +3)}]^{32} && (4.11)\\
&= \left(\frac{1}{4}\right)^{32} && (4.12)\\
&= 2^{-64} && (4.13)\\
\text{P(32 lines transmit +3)} &= 54.21 \times 10^{-21} && (4.14)
\end{aligned}$$

and the probability that either all 32 lines transmit +3 or all 32 lines transmit -3 is twice this result. Thus,

$$P(32 \text{ lines transmit } +3 \cup 32 \text{ lines transmit } -3) = 2^{-63} = 108.4 \times 10^{-21} \qquad (4.15)$$

Again, the probability that the transmitted envelope achieves its maximum value is negligible compared to the bit-error-rate, and it is asserted that all reasonably likely amplitudes lie below some threshold, which is significantly less than the maximum value of 96. A histogram of the amplitude values for a simulation run is given in Figure 6.3 in the Results chapter.

4.5 DEMODULATION

For both CPFSK and MAMSK, the correlation of the received envelope with each of two reference phases is used to estimate the transmitter's chips. The transmitters chips are then despread by each data line at the receiver to recover the data.

4.5.1 CPFSK DEMODULATION

For CPFSK modulation, the reference phases have amplitude of 1 and angles of 45 and 135 degrees. The block diagram in Figure 4.11 shows how to estimate the chips using these reference phases.

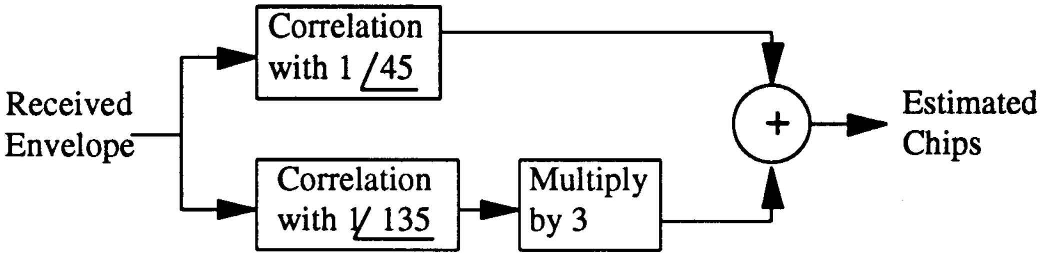


Figure 4.11: CPFSK Demodulator Block Diagram

In performing the CPFSK demodulation technique used here, it is required to calculate the sum of each of the two disjoint sets of input values defined in Section 4.4.1 (i.e, those data lines transmitting +1 or -1 and those data lines transmitting +3 or -3). Since the two reference vectors are orthogonal, they have no correlation with each other. Therefore, the sums of the two sets are calculated by performing correlation with each of the two reference vectors.

The result of the correlation with the 45-degree reference (corresponding to input values of +1 and -1) is the receiver's estimate of the sum of the first input set (all inputs transmitting either +1 or -1). The result of the correlation with the 135-degree reference (which corresponds to input values of +3 and -3) is tripled to give the receiver's estimate of the sum of the second input set (all inputs transmitting either +3 or -3), because the reference vector has an amplitude of 1, but the input values have a value of 3. These estimated sums are then added to form an estimate of the transmitted chips. The estimated chips are then despread by correlation with either the Barker or Gold codes and decoded with Table 4.3, resulting in an estimate of the input values.

4.5.2 MAMSK DEMODULATION

For MAMSK modulation, the reference phases have amplitude of 1 and angles of 0 and 90 degrees. In this case, the phases are offset in time by half of a symbol-time. The block diagram in Figure 4.12 shows how to estimate the chips using these reference phases.

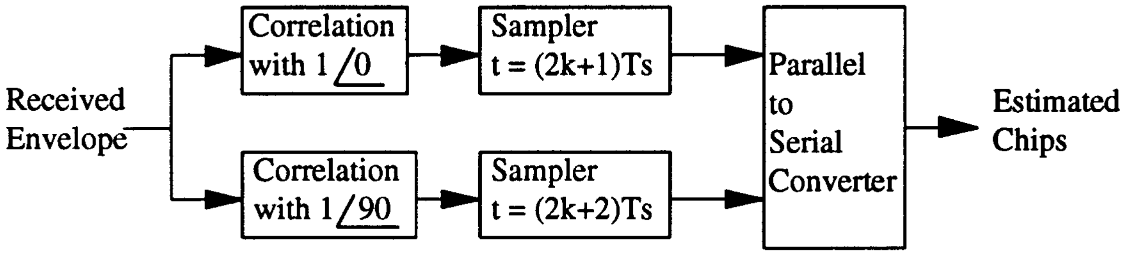

Received
Envelope
Correlation with $1\angle 0$
Sampler t = (2k+1)Ts
Correlation with $1\angle 90$
Sampler t = (2k+2)Ts
Parallel to Serial Converter
Estimated Chips

Figure 4.12: MAMSK Demodulator Block Diagram

The MAMSK receiver used here operates on the complex envelope of the received signal. As in standard MSK, the envelope values are correlated with the cosine and sine carriers (i.e., phasors at angles 0 and 90 degrees, respectively), giving the receiver's estimates of the transmitted chips. Note that the correlation values are alternately sampled, at sample times offset by half of a symbol-time. The chips are then despread using either Gold or a Barker codes. The resulting values are then decoded according to the decision regions in Table 4.3, resulting in the receiver's estimates of the transmitted data. The only difference here between MAMSK and standard MSK is in the decision regions of the received, despread value. Standard MSK must only decide whether a given value is a +1 or a -1. In contrast, MAMSK must decide between the four possibilities of -3, -1, +3, and +1.

4.6 CODE-DIVISION MULTIPLE ACCESS

Code-division multiple access (CDMA) is the use of spread-spectrum techniques to provide a means other than time-division or frequency-division multiple access for users to share the resources of the channel. All CDMA users are permitted to transmit simultaneously and use the entire available bandwidth [7]. Each user is assigned a unique spreading code, which is correlation-detected at its receiver. The spreading codes are chosen to have as small a cross-correlation as possible, since cross-correlation causes mutual interference between users.

In this investigation, the lines of the HIPPI data bus are regarded as the "users" sharing the bandwidth. The lines are multiplexed onto the satellite channel using CDMA, allowing the data to be synchronously transmitted across the channel in parallel, instead of converting the thirty-two parallel lines of data into a serial bit-stream.

A matrix equation for the cross-correlation of two binary sequences is defined in Section 4.6.1. Section 4.6.2 expands the matrix notation for the generation of the CDMA codes. Two CDMA codes, Barker codes and Gold codes, are described in Sections 4.6.3 and 4.6.4, respectively. In addition, a slight modification, to reduce mutual interference between data lines, is described for each code. In fact, for perfect synchronization, cross-talk is eliminated.

4.6.1 CROSS-CORRELATION FOR TWO BINARY SEQUENCES

Following the convention in [7], the cross-correlation between two length-N binary sequences $\{a_i\}$ and $\{b_i\}$ is defined as

$$R_{ab}(k) = \frac{1}{N}\sum_{n=0}^{N-1} a_n b_{n+k} \tag{4.16}$$

where the elements of the two sequences can take the values of +1 or -1 and where k is the time-shift between the start of the two codes. In this project, the CDMA users transmit synchronously; thus, the case of k=0 is of particular interest here. With k=0, Equation 4.16 can be rewritten in vector form as

$$R_{ab}(0) = \frac{1}{N} ab^T \tag{4.17}$$

where a and b are length-N row-vectors whose elements contain the elements a_i and b_i, respectively. The superscript T indicates the transposition of the vector b. Note that this multiplication is simply the inner product (or "dot product") of a and b.

The power in the vector notation in Equation 4.17 is that the equation can be generalized into a matrix equation which gives the cross-correlation of each sequence with all others. If the binary sequences form the rows of a matrix A, then the zero-time-shift auto-correlation matrix of A with itself is defined as

$$R = \frac{1}{N} AA^T \tag{4.18}$$

where the elements of A can take the values +1 or -1. In the matrix R, each element R_{ii} (i.e., an element along the main diagonal) represents the peak autocorrelation of a sequence with itself, which is +1 for the case of zero time-shift. An off-diagonal term R_{ij} represents the cross-correlation of sequence i with sequence j. Since this cross-correlation represents how much code i will interfere with code j in a CDMA scheme, it is desired to make the off-diagonal terms of R as small as possible. In the complete absence of cross-talk, the off-diagonal terms would equal zero, and R would become the identity matrix.

4.6.2 CODE GENERATION AND RECEPTION

Following the matrix notation introduced in Section 4.6.1, let the spreading codes have length N and form the rows of a spreading matrix, call it S, and let the Gray-coded input symbols (either -3, -1, +1, +3, from Table 4.1) for all the available data lines at a given time be the elements of a column vector x. Then, spreading the data corresponds to multiplying each data line's input value (which is a scalar) by the line's spreading code (i.e., its row of the spreading matrix S) to form the "chips" for the line, which is a series of N values which represent the single input symbol transmitted, where N is the spreading-code length. The chips for all the channels are then added together to form a row vector y, which is a series of N values transmitted over the channel. In matrix notation, this operation can be represented by the matrix multiplication

$$y = x^T S \tag{4.19}$$

For the moment, the effects of noise and channel nonlinearities will be neglected. To despread the transmitted waveform, the received sequence y is correlated with each of the spreading codes. Using the result of Equation 4.17 in Section 4.6.1, for the i^{th} data line, the correlation of the received waveform with the spreading code is

$$c_i = \frac{1}{N} S_i y^T \tag{4.20}$$

where S_i is the i^{th} row of S, which is the i^{th} spreading code. The N vector equations represented in Equation 4.20 can be written compactly in matrix form as

$$c = \frac{1}{N} S y^T \tag{4.21}$$

where c is a column vector representing the correlation values for each line. Using Equation 4.19, Equation 4.21 can be multiplied by the scalar N and be rewritten as

$$Nc = Sy^T \tag{4.22}$$

$$= S(x^T S)^T \tag{4.23}$$

$$Nc = SS^T x \tag{4.24}$$

Therefore,

$$c = \frac{1}{N} SS^T x \tag{4.25}$$

Thus, if SS^T, which is the correlation of the codes with each other, is a scalar multiple of the identity matrix, there is no crosstalk between the users, and, by Equation 4.25, c is proportional to x.

The effects of channel nonlinearities and noise are to perturb the received values from their ideal values. The receiver will estimate the ideal value according to the decision regions corresponding to the received values in Table 4.3.

Received Value	Estimated Symbol
< -2	-3
-2 to 0	-1
0 to +2	+1
> 2	+3

Table 4.3: Decision Regions at the Receiver

Thus, the receiver will make an error if the channel degradations are large enough to cause the received value to lie in the wrong decision region. For example, suppose a -3 is transmitted, but the channel corrupts the value by adding 2.5 to it. Therefore, the receiver receives a value of -0.5, which it decodes as a -1 using Table 4.3, resulting in an error. Thus, care must be taken to minimize the channel degradations.

4.6.3 BARKER CODES

A Barker sequence is a binary sequence of length N with the property that its autocorrelation function is no larger than 1/N in magnitude for all time-shifts other than zero (modulo N) [8]. Of course, the autocorrelation function is 1 for a time-shift of 0 (modulo N). This property will be shown to be useful for CDMA systems in which all users transmit synchronously [11], which is the case for a HIPPI bus, where the data lines are regarded as the CDMA users. The known Barker sequences are given in Table 4.4 [8], where only the sign of each element is shown (e.g., “+−” in the first row of the table corresponds to the length-2 Barker sequence [+1 −1]). Multiplying a Barker sequence by -1 or reversing the order of its elements will still produce a Barker sequence [8]. Note that the longest known Barker sequence has a length of 13.

Length	Sequence
2	++ and +−
3	++−
4	+++− and ++−+
5	+++−+
7	+++−−+−
11	+++−−−+−−+−
13	+++++−−++−+−+

Table 4.4: Barker Sequences

Following the technique in [11], each of the N cyclic shifts of a length-N Barker sequence is taken as a spreading code. For example, a Barker-4 matrix, whose rows contain the cyclic shifts of the Barker-4 sequence, is shown below:

$$B_4 = \begin{bmatrix} -1 & +1 & +1 & +1 \\ +1 & -1 & +1 & +1 \\ +1 & +1 & -1 & +1 \\ +1 & +1 & +1 & -1 \end{bmatrix} \tag{4.26}$$

Note that the order in which the rows appear in the matrix is unimportant, but each row is a cyclic shift of the Barker code + + +−.

Thus, from the autocorrelation property of the Barker sequence, each of these codes has a cross-correlation not exceeding 1/N in magnitude with all other codes, provided that all codes are transmitted synchronously. Thus, up to N CDMA users may share the channel using length-N Barker spreading codes.

CONCATENATION OF BARKER CODES

In order to permit more than 13 users, the codes must be concatenated. For the HIPPI system, a length 52 (4 × 13) sequence will suffice.

In order to form a size $(n \times r)$ Barker matrix from two Barker codes of length n and r $(r < n)$, first form the size n Barker matrix, B_n. Then, write the size r Barker matrix, but instead of using +1 and -1 as its elements, use $+B_n$ and $-B_n$ as submatrices of the larger matrix. For example, if B_{13} is the Barker-13 matrix, then a size-52 $(52 = 4 \times 13)$ matrix is formed from the Barker-4 matrix as follows:

$$B_4 = \begin{bmatrix} -1 & +1 & +1 & +1 \\ +1 & -1 & +1 & +1 \\ +1 & +1 & -1 & +1 \\ +1 & +1 & +1 & -1 \end{bmatrix} \tag{4.27}$$

$$B_{52} = \begin{bmatrix} -B_{13} & +B_{13} & +B_{13} & +B_{13} \\ +B_{13} & -B_{13} & +B_{13} & +B_{13} \\ +B_{13} & +B_{13} & -B_{13} & +B_{13} \\ +B_{13} & +B_{13} & +B_{13} & -B_{13} \end{bmatrix} \tag{4.28}$$

The correlation properties of this concatenated matrix are discussed in Section 4.6.3.

ELIMINATION OF BARKER-CODE CROSSTALK

Other than the length-2 and length-4 codes, Barker codes' correlation matrices have +1/N in all the off-diagonal positions. Thus, unlike Gold codes, Barker codes require -1/N to be added to all values of the correlation matrix if crosstalk is to be eliminated. Augmenting a Barker matrix, B, to form B_{aug} as before, the resulting correlation matrix is

$$R_{aug} = \frac{1}{N} B_{aug} B_{aug}^T \tag{4.29}$$

$$= \frac{1}{N} \begin{bmatrix} B & v \end{bmatrix} \begin{bmatrix} B^T \\ v^T \end{bmatrix} \tag{4.30}$$

$$R_{aug} = \frac{1}{N}(BB^T + vv^T) \tag{4.31}$$

Solving for a column vector v such that vv^T is a matrix of all -1's has no real solution. To find a real solution, we relax the condition that the receiver filter be exactly matched to the source. If we augment the source by a column vector of all +1's (p) and augment the receiver by a column vector of all -1's (n), we get

$$R_{aug} = \frac{1}{N} \begin{bmatrix} B & p \end{bmatrix} \begin{bmatrix} B^T \\ n^T \end{bmatrix} \tag{4.32}$$

$$R_{aug} = \frac{1}{N}(BB^T + pn^T) \tag{4.33}$$

where pn^T is a square matrix of all -1's.

Thus, the crosstalk has been removed. However, -1/N has been added to every element of the correlation matrix, including the autocorrelation values on the main diagonal. This slightly degrades the signal-to-noise ratio.

Another complication occurs when a larger Barker-code matrix is formed from smaller Barker matrices. For example, observe the correlation matrix of a Barker-52 code:

$$B_{52} = \begin{bmatrix} -B_{13} & +B_{13} & +B_{13} & +B_{13} \\ +B_{13} & -B_{13} & +B_{13} & +B_{13} \\ +B_{13} & +B_{13} & -B_{13} & +B_{13} \\ +B_{13} & +B_{13} & +B_{13} & -B_{13} \end{bmatrix} \tag{4.34}$$

$$52R_{B52} = B_{52}B_{52}^T \tag{4.35}$$

$$= \begin{bmatrix} -B_{13} & +B_{13} & +B_{13} & +B_{13} \\ +B_{13} & -B_{13} & +B_{13} & +B_{13} \\ +B_{13} & +B_{13} & -B_{13} & +B_{13} \\ +B_{13} & +B_{13} & +B_{13} & -B_{13} \end{bmatrix} \begin{bmatrix} -B_{13}^T & +B_{13}^T & +B_{13}^T & +B_{13}^T \\ +B_{13}^T & -B_{13}^T & +B_{13}^T & +B_{13}^T \\ +B_{13}^T & +B_{13}^T & -B_{13}^T & +B_{13}^T \\ +B_{13}^T & +B_{13}^T & +B_{13}^T & -B_{13}^T \end{bmatrix} \tag{4.36}$$

$$52R_{B52} = \begin{bmatrix} 4M_{B13} & 0 & 0 & 0 \\ 0 & 4M_{B13} & 0 & 0 \\ 0 & 0 & 4M_{B13} & 0 \\ 0 & 0 & 0 & 4M_{B13} \end{bmatrix} \tag{4.37}$$

where

$$M_{B13} = B_{13}B_{13}^T = 13R_{B13}$$

The correlation matrix $52R_{B52}$ is graphed in Figure 4.13.

There is some crosstalk in the Barker-52 code, since the matrix $4M_{13} = 52R_{B13}$ contains +4's in its off-diagonal terms. If the Barker-52 matrix were augmented with +2 at the transmitter and -2 at the receiver (the method described above for non-concatenated codes), the crosstalk would still exist, since this would subtract 4 from *every* element in the R_{B52} matrix. Thus, while the crosstalk would be eliminated from the M_{B13} submatrices along the main diagonal of R_{B52}, -4 would also be added to every element of every zero matrix located off the main diagonal of R_{B52}.

Instead, the inner Barker matrix (except Barker-2 and Barker-4, which have no crosstalk) must be augmented *before* building the larger Barker matrix. Here, the Barker-13 matrix should be augmented to form two 13x14 matrices (one for the transmitter and one for the receiver), which are then used to build the corresponding Barker-56 matrices.

At the transmitter, the Barker-13 matrix will be augmented by p, a vector of all +1's, in forming the B_{56T} matrix:

$$B_{14T} = \begin{bmatrix} B_{13} & p \end{bmatrix} \tag{4.38}$$

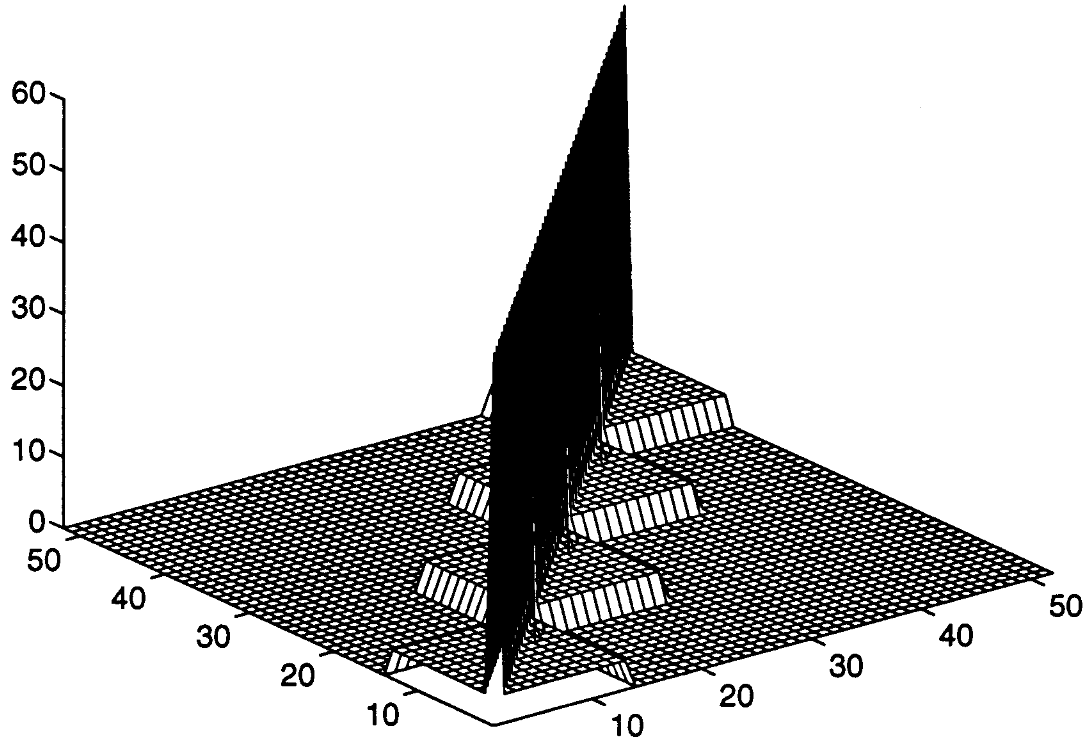


Figure 4.13: Barker-52 Code Correlation

$$B_{56T} = \begin{bmatrix} -B_{14T} & +B_{14T} & +B_{14T} & +B_{14T} \\ +B_{14T} & -B_{14T} & +B_{14T} & +B_{14T} \\ +B_{14T} & +B_{14T} & -B_{14T} & +B_{14T} \\ +B_{14T} & +B_{14T} & +B_{14T} & -B_{14T} \end{bmatrix} \tag{4.39}$$

At the receiver, the Barker-13 matrix will be augmented by n, a vector of all -1's, in forming the B_{56R} matrix:

$$B_{14R} = \begin{bmatrix} B_{13} & n \end{bmatrix} \tag{4.40}$$

$$B_{56R} = \begin{bmatrix} -B_{14R} & +B_{14R} & +B_{14R} & +B_{14R} \\ +B_{14R} & -B_{14R} & +B_{14R} & +B_{14R} \\ +B_{14R} & +B_{14R} & -B_{14R} & +B_{14R} \\ +B_{14R} & +B_{14R} & +B_{14R} & -B_{14R} \end{bmatrix} \tag{4.41}$$

The correlation matrix of the Barker-56 code is then

$$R_{B56} = \frac{1}{56} B_{56T} B_{56R}^T \tag{4.42}$$

$$= \frac{1}{56} \begin{bmatrix} 4M_{B14} & 0 & 0 & 0 \\ 0 & 4M_{B14} & 0 & 0 \\ 0 & 0 & 4M_{B14} & 0 \\ 0 & 0 & 0 & 4M_{B14} \end{bmatrix} \tag{4.43}$$

where

$$M_{B14} = B_{14T} B_{14R}^T \tag{4.44}$$

$$= \begin{bmatrix} B_{13} & p \end{bmatrix} \begin{bmatrix} B_{13}^T \\ n^T \end{bmatrix} \tag{4.45}$$

$$= B_{13} B_{13}^T + p n^T \tag{4.46}$$

$$M_{B14} = 12 I_{13} \tag{4.47}$$

where I_{13} is the 13x13 identity matrix.

Thus, all off-diagonal terms of the Barker-56 correlation matrix are zero. The Barker-56 correlation matrix is then

$$R_{B56} = \frac{48}{56} I_{52} \tag{4.48}$$

Therefore, since R_{B56} is proportional to the identity matrix, there is no crosstalk between the channels. However, note that the autocorrelation values along the main diagonal are 48/56 instead of 1. Thus, the correlation matrix is multiplied by 56/48 to normalize the autocorrelation values to 1. An example of two of the Barker codes is given in Figure 4.14, while the correlation matrix $56R_{B56}$ is graphed in Figure 4.15.

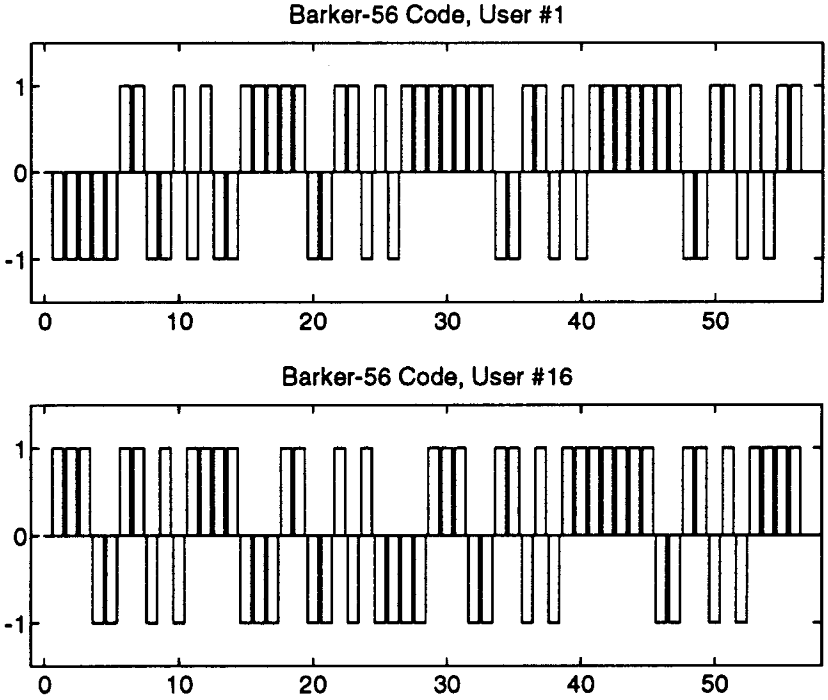


Figure 4.14: The Barker Codes used by Data Lines 1 and 16

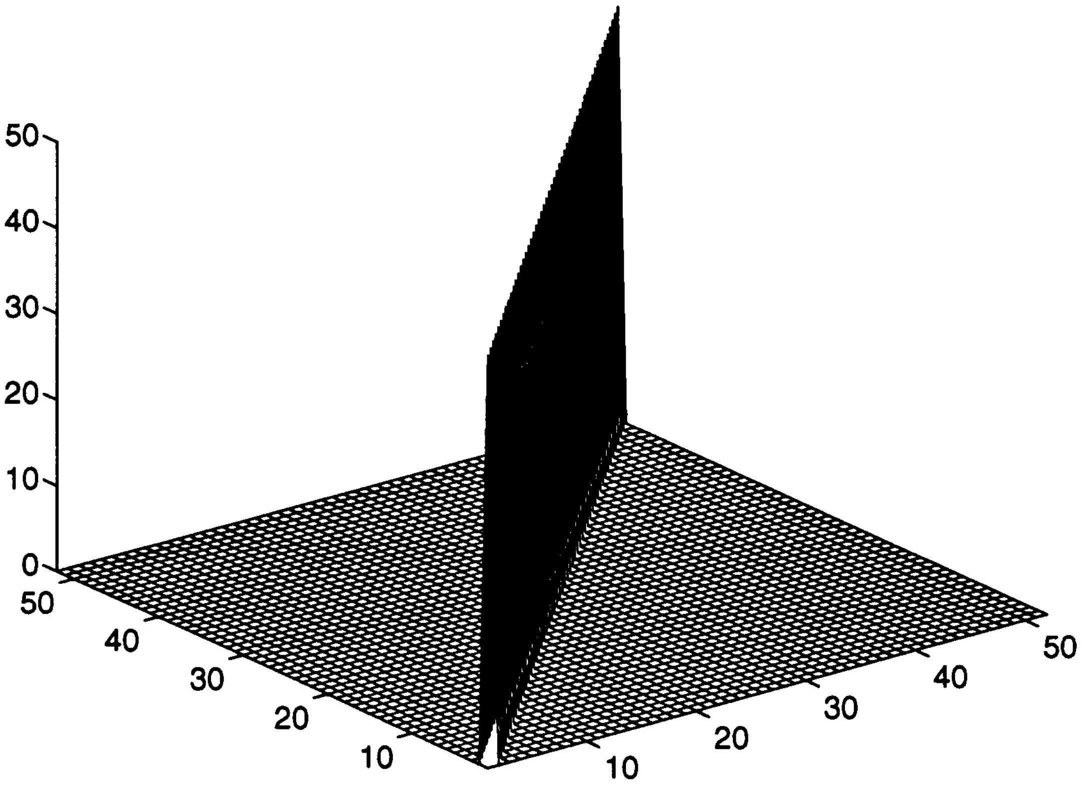


Figure 4.15: Barker-56 Code Correlation

4.6.4 GOLD CODES

Gold codes are binary CDMA spreading codes created from two maximal-length pseudonoise sequences. The cross-correlation of two Gold codes is constrained to take one of three values [7]

$$R_{ij} \in \begin{cases} -\frac{1}{N}t(n) \\ -\frac{1}{N} \\ \frac{1}{N}[t(n)-2] \end{cases} \tag{4.49}$$

where $N = 2^n - 1$ is the length of the code and

$$t(n) = \begin{cases} 1 + 2^{(n+1)/2} & \text{for } n \text{ odd} \\ 1 + 2^{(n+2)/2} & \text{for } n \text{ even} \end{cases}$$

Gold codes are generated from two maximal-length pseudonoise sequences ("pn-sequences") which must meet criteria described below. A length-N Gold code family contains N+2 codes. This allows a maximum of N+2 users to share the channel using CDMA. However, for a synchronous system, performance is improved if only N+1 users are allowed to share the system, since one of the codes has inferior cross-correlation properties as compared to the others. The method used here to generate Gold codes follows [7]. First, a table of irreducible polynomials, such as in Peterson and Weldon [12], is used to find a generating polynomial of degree n, where $n = \log_2(N+1)$ and N is the code length, as before. This generating polynomial represents the feedback connections of a shift-register circuit, which will generate a pn-sequence of length N. The circuit for generating the Gold-31 codes is shown in Figure 4.16 [7].

The second pn-sequence b_2 is a "decimation" of the first pn-sequence b_1; i.e., the second sequence is obtained by sampling every qth symbol of the first. Notationally, $b_2 = b_1[q]$. The decimation of a pn-sequence does not necessarily result in another pn-sequence. The table in Peterson and Weldon [12] indicates which decimations are "proper" and do yield another pn-sequence. Once a proper decimation is found, the following conditions sufficient for the two pn-sequences to be called a "preferred pair" of pn-sequences [7].

1. $n \neq 0 \bmod 4$; i.e., n is odd or $n = 2 \bmod 4$

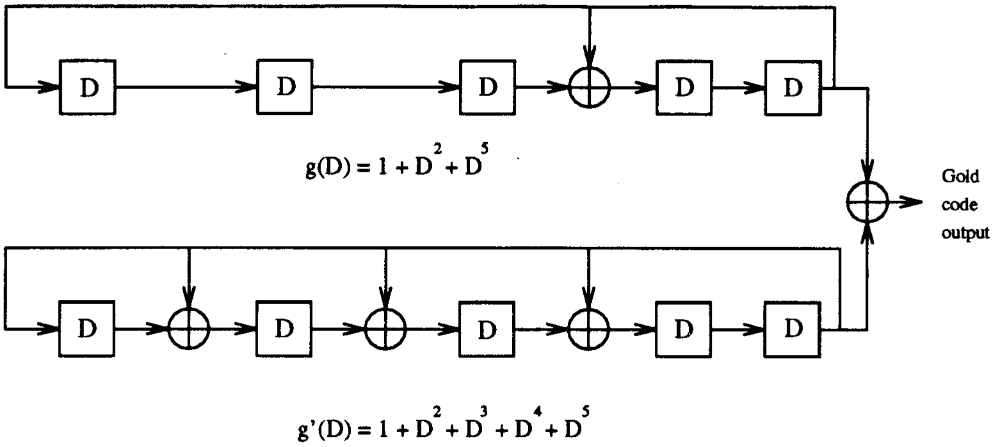


Figure 4.16: Shift-register Circuit for Generating Gold-31 Codes

2. $b_2 = b_1[q]$ where q is odd and either

$$q = 2^k + 1$$

$$\text{or} \quad q = 2^{2k} - 2^k + 1$$

for some k.

3. The greatest common divisor between n and k is

$$\gcd(n, k) = \begin{cases} 1 & \text{for } n \text{ odd} \\ 2 & \text{for } n = 2 \bmod 4 \end{cases}$$

If these criteria are met, the preferred pair can be used to generate the family of Gold codes. Each of the N-1 cyclic shifts of b_2 is added, modulo 2, to b_1 to form N-1 new codes. If the circuit in Figure 4.16 is used to generate the codes, the cyclic shifts can be generated by varying the initial conditions of the shift register, using a method described in [7]. These N-1 codes, along with the original pn-sequences b_1 and b_2, form the N+1 codes in the Gold code family.

Unlike Barker codes, Gold codes are not restricted to synchronous systems. However, for a synchronous system, the cross-correlation among N of the codes is uniformly -1/N. The cross-correlation of the second pn-sequence, b_2, is not uniformly -1/N but is restricted to the values in Equation 4.49.

ELIMINATION OF GOLD-CODE CROSSTALK

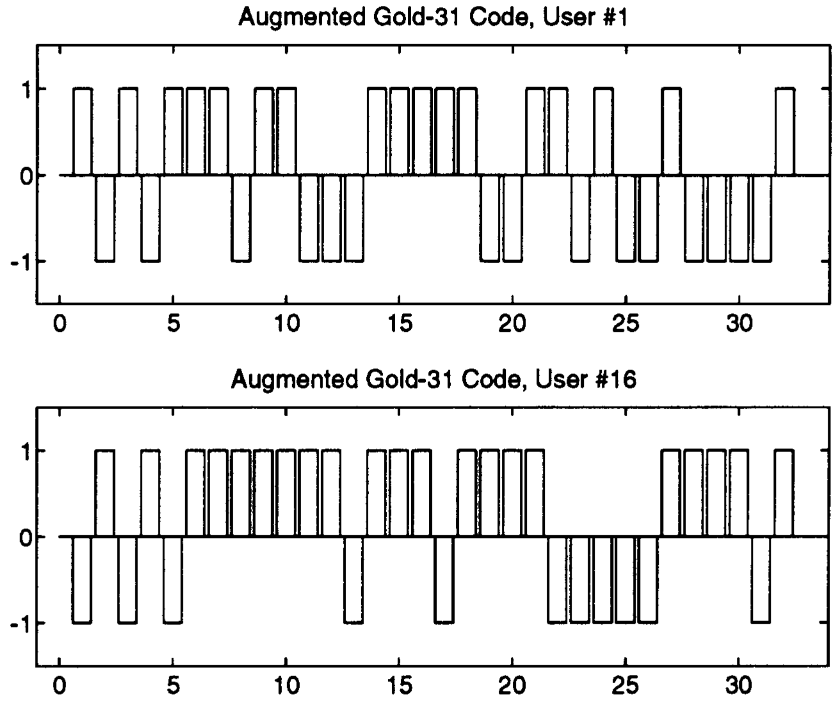
A Gold code correlation matrix has -1/N in all the off-diagonal positions. If +1/N could be added to every element of the correlation matrix, the matrix would become a multiple of the identity matrix, eliminating crosstalk. If the Gold code matrix, G, is augmented by a column vector to form a new matrix, G_{aug}, the following correlation matrix results:

$$R_{aug} = \frac{1}{N} G_{aug} G_{aug}^T \tag{4.50}$$

$$= \frac{1}{N} \begin{bmatrix} G & v \end{bmatrix} \begin{bmatrix} G^T \\ v^T \end{bmatrix} \tag{4.51}$$

$$R_{aug} = \frac{1}{N}(GG^T + vv^T) \tag{4.52}$$

Thus, finding a vector v such that vv^T is a matrix of all ones which is the same size as G would eliminate crosstalk. A column vector containing all 1's is one solution. Thus, appending a 1 to the end of every Gold code will eliminate crosstalk for synchronous users. Also note that the autocorrelation values along the main diagonal have also been increased by 1/N, increasing the signal-to-noise ratio. An example of two Gold codes is given in Figure 4.17, while the correlation matrix for the augmented Gold-31 code is normalized by 32 and graphed in Figure 4.18. Note that, for 32 of the 33 codes, the correlation is either 0 or 1. Only the last code has slightly worse correlation (indicated by the peaks in the last row and column of the matrix).


Augmented Gold-31 Code, User #1

Augmented Gold-31 Code, User #16

Figure 4.17: The Gold Codes used by Data Lines 1 and 16

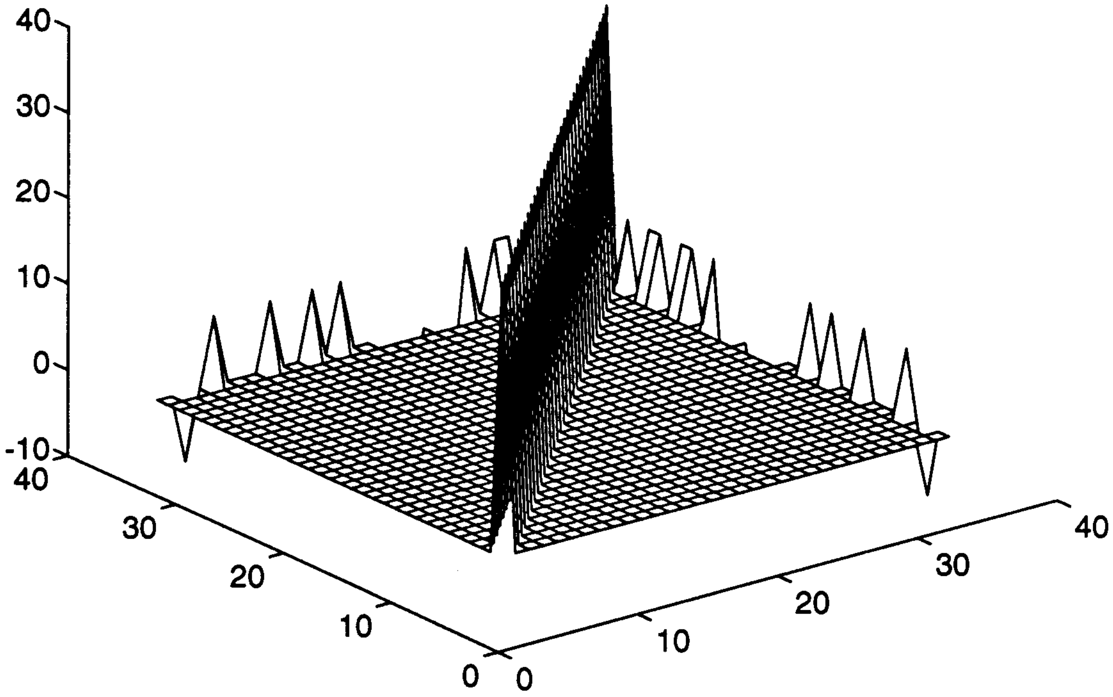


Figure 4.18: Augmented Gold-31 Code Correlation

4.7 MODULATION AND CODING ALTERNATIVES

The High-performance Parallel Interface (HIPPI) is a simplex point-to-point interface for transferring data at a rate of either 800 or 1600 Mb/s, depending on the specific implementation [1]. Duplex operation can be obtained by using two simplex HIPPI links, one for each direction of data travel. For this project, the 800 Mb/s data rate is assumed. The HIPPI interface is a parallel architecture with 32 data lines (for 800 Mb/s operation) and a 25 MHz clock.

Initially, three CDMA codes (Gold-31, Gold-63, and Barker-52) were considered for augmentation and use for spreading the HIPPI lines onto the channel. The bandwidth required for each code to support 800 Mbits/sec of data is shown in Table 4.5.

Spreading Code	Augmented Length	Required Bandwidth
Gold-31	32	600 MHz
Gold-63	64	1200 MHz
Barker-52	56	840 MHz

Table 4.5: Bandwidth Requirements for Spreading Codes

For the HIPPI control lines to be spread along with the 32 data lines, a code supporting at least 45 users is necessary. However, the Gold-63 code requires more bandwidth than the 900 MHz available on ACTS. Thus, only two choices for satellite transmission of both data and control lines remain.

1. Spread the control lines and the data lines using an augmented Barker-52 code. This will use almost all available bandwidth.

2. Spread only the data lines with an augmented Gold-31 code, which supports 32 users and requires only 600 MHz. The control lines would then be transmitted separately on the remaining 300 MHz of bandwidth.

Another option is to transmit the control lines over a terrestrial line. This offers the advantage of reduced propagation delays (which improves the performance of error recovery and control procedures) but requires the expense of buying or leasing the terrestrial lines.

Chapter 5

MODEL DESCRIPTION

The simulation model consists of a series of programs written in the language of the MathWorks' *MatLab* software and listed in the Appendix. The simulation of the overall system (shown in Figure 4.1) is broken down into the following smaller tasks:

1. Randomly generate data to be sent across the channel;
2. Spread the data on each data line with either a Gold or a Barker code;
3. Modulate the data with either MAMSK or CPFSK modulation and find the complex envelope of the signal transmitted via ACTS;
4. Simulate the effect of the uplink transmission (noise, rain attenuation);
5. Simulate the effects of ACTS nonlinearities;
6. Simulate the downlink transmission (noise, rain attenuation);
7. Simulate amplitude and phase equalization performed at the receiver;
8. Demodulate and despread the signal, to find the receiver's estimate of the transmitted data;
9. Compare the received and transmitted data and calculate the simulated bit-error-rate.

In beginning the simulation, the user specifies the number of bits to be simulated, the uplink noise variance, the uplink rain rate (in mm/hr), the downlink noise variance, the downlink rain rate (mm/hr), and the type of spreading code used (Gold or Barker).

5.1 GENERATION OF RANDOM INPUT DATA

Since there are 32 HIPPI lines each transmitting 2 bits at any given time, a total of 64 data bits are transmitted across the link during each bit time. Thus, the number of bit times required is 1/64th the number of bits to be transmitted. Since the data are assumed to be independent and equally likely values taken from the set {-3, -1, 1, 3}, MatLab's random number generator is used to generate a series of vectors with length 32 (one element for each data line) and values randomly taken from the set {-3, -1, 1, 3}. These vectors correspond to the x, the vector of input symbols in Section 4.6.2.

5.2 SPREADING CODE GENERATION

Following the matrix notation described in Section 4.6.2, the spreading codes are placed in the rows of a spreading matrix. The matrix is then augmented as described in Sections 4.6.4 and 4.6.3 to reduce the crosstalk between the data lines.

5.2.1 GOLD CODES

In order to generate the Gold-31 codes used in this investigation, suitable generating polynomials were found as described in Section 4.6.4. The octal numbers representing the shift-register connections in Figure 4.16 were converted to a binary array (1 means a connection exists at a node; 0 means no connection). As described in section 4.6.4, the output of each of the two shift-register was then computed, for the initial condition of 00001, giving two pn-sequences. Then, as described in 4.6.4, the second pn-sequence is repeatedly shifted and added (modulo 2) to the first, giving 31 additional Gold codes, which form the rows of a Gold-31 matrix. Since, in the synchronous case of interest here, the second pn-sequence (the sequence which was repeatedly shifted) has slightly inferior cross-correlation properties, it is placed at the end of the matrix, where it will not be used. It is only included in the matrix for completeness. Once the standard Gold codes are generated, the method of Section 4.6.4 is employed to reduce cross-talk.

5.2.2 BARKER CODES

Following the method outlined in Sections 4.6.3 and 4.6.3, the Barker-56 transmit and receive matrices were built from a 4×4 array of Barker-14 matrices. In order to construct the Barker-14 matrices, Table 4.4 is used to find a Barker sequence of length 13, and, as in Section 4.6.3, a Barker-13 matrix is formed from the cyclic shifts of the sequence. Following Section 4.6.3, the Barker-13 matrices are then augmented to form Barker-14 matrices, which is copied into a 4×4 array (with algebraic sign determined by the Barker-4 matrix) to form the Barker-56 matrix.

5.3 MODULATION

For computational convenience, signals are represented in the complex-envelope domain. In order to reduce the number of computations necessary, transitions between envelope states are not computed. Thus, only one envelope value is computed per chip.

5.3.1 MAMSK TRANSMITTER

Following the MAMSK modulator block diagram in Figure 4.9, the serial stream of coded input bit-pairs is converted into two parallel streams. One stream modulates the reference phase with angle 0, while the other is delayed and modulates the phase with angle 90 degrees. The complex envelope is formed by taking the 0-degree-envelope as the real part and the 90-degree-envelope as the imaginary part.

5.3.2 CPFSK TRANSMITTER

As discussed in Section 4.4.1, a variant of CPFSK is used here. Each data line can transmit one of four values at any given time (-3, -1, 1, or 3), corresponding to a vector with amplitude 1 and one of four angles (-45, -135, 45, or 135 degrees, respectively). The total signal transmitted across the channel is the vector sum of the individual channels' vectors.

5.4 UPLINK MODEL

The uplink is modeled as a linear channel corrupted by additive, white Gaussian noise and rain attenuation. The Gaussian noise is generated using MatLab's random-number generator to form random vectors whose values are zero-mean Gaussian values with variance specified by the user. The rain attenuation is modeled as described in Section 4.1. In order to simplify the model, the free-space loss is not modeled since it is a constant value, which can be easily compensated for by increasing the transmitted power.

5.5 ACTS SATELLITE MODEL

The ACTS channel is modeled as a noiseless frequency down-converter corrupted by both amplitude and phase nonlinearities. For computational convenience, third-order polynomial approximation (fitted to the curve of Figure 4.3 in a least-squares sense) of the saturation curve is used to model the amplitude nonlinearity. As shown in Figure 4.4, the phase nonlinearity is modeled as a linear function of input amplitude, expressed in decibels.

5.6 DOWNLINK MODEL

Like the uplink, the downlink is modeled as a linear channel corrupted by additive, white Gaussian noise and rain attenuation. The noise and rain are modeled in the same manner as for the uplink; only the carrier frequency is different from the uplink.

5.7 RECEIVER EQUALIZATION

Since both Gold and Barker codes require an approximately linear channel for best performance, equalization is necessary to compensate for the ACTS nonlinearities. Following the method discussed in Section 4.3, a third-order polynomial function is used for amplitude equalization, while a linear function is used for phase equalization. The amplitude-equalizing polynomial is evaluated at the value of the received envelope's amplitude. The result of the evaluation is the receiver's estimate of the transmitted envelope's amplitude before any AM-AM distortion. This estimated amplitude is then processed according to the phase-equalizing function. This is the receiver's estimate of the correction necessary to remove the phase nonlinearity. The estimated amplitude and corrected phase give the amplitude and phase of the receiver's estimate of the transmitted envelope.

5.8 DEMODULATION AND DETECTION

For both MAMSK and CPFSK modulation, the operations of demodulation and despreading are closely related. As described in Section 4.5, the signal is demodulated by correlating the received amplitude with appropriate reference phases. This correlation is then converted into an estimate of the transmitter's chips. The signal is then despread by correlation with the spreading codes.

5.8.1 MAMSK DEMODULATION

The MAMSK receiver simulated here operates on the complex envelope of the received signal. As in standard MSK, the envelope values are correlated with the cosine and sine carriers (i.e., phasors at angles 0 and 90 degrees, respectively), giving the receiver's estimates of the transmitted chips. The chips are then despread using either a Gold or a Barker matrix. The resulting values are then decoded according to the decision regions in Table 4.3, resulting in the receiver's estimates of the transmitted data. The only difference here between MAMSK and standard MSK is in the decision regions of the received, despread value. Standard MSK must only decide whether a given value is a +1 or a -1. In contrast, MAMSK must decide between the four possibilities of -3, -1, +3, and +1.

5.8.2 CPFSK DEMODULATION

The CPFSK receiver used here also performs correlation detection. Once amplitude and phase equalization have been performed on the received envelope, the resulting vector is correlated with two reference vectors with amplitude 1 and angles of 45 and 135 degrees. The result of the correlation with the 135-degree reference (which corresponds to input values of +3 and -3) is then tripled and added to the other correlation value (corresponding to inputs of +1 and -1). This sum is an estimate of the transmitted chips. As with MAMSK, the estimated chips are then despread by correlation with either the Barker or Gold codes and decoded with Table 4.3, resulting in an estimate of the input values.

5.9 SIMULATION OF SYSTEM PERFORMANCE

The system performance is simulated for all four possibilities of modulation technique (MAMSK or CPFSK) and CDMA code (Barker-56 code or augmented Gold-31 code). Since the control lines are spread with CDMA codes for the Barker-56 code but not for the augmented Gold-31 code, the control lines' bit-error-rate was not simulated. Instead, for both CDMA codes, 32 CDMA users (the data lines) were assumed to be active.

The symbol-error rate is found by counting the number of output symbols which differ from the corresponding input symbols. In order to convert this symbol-error rate to a BER, the number of symbols with two bit-errors must counted twice to include both errors in the BER. Note from Table 4.3 that a double bit-error occurs when a -3 is decoded as +1 or when -1 is decoded as +3. In both cases, the absolute value of the difference between the input symbol and the output symbol is 4. If all the input symbols are subtracted from the corresponding output symbols, the number of double-bit-errors is therefore the number of times that the absolute value of this difference is 4. The number of bit-errors is found by adding the number of double-bit-errors to the number of symbol-errors. The simulated bit-error-rate (BER) is then the number of bit-errors divided by the number of bits transmitted.

Chapter 6

RESULTS

In order to determine whether a HIPPI-ACTS-HIPPI connection is feasible, the simulated envelope distortion (due to channel nonlinearities) and bit-error-rate are examined. The simulated envelope distortion given of Section 6.1 shows that the channel equalization algorithm employed here is sufficient. Once a suitable equalization algorithm was found, the bit-error-rate was simulated for various levels of signal-to-noise ratio and described in Section 6.2

6.1 ENVELOPE DISTORTION

As described in Section 4.4.3, the amplitude of the transmitted envelope can take several values. A simulation of 6400 random input bits was run for each modulation technique, MAMSK and CPFSK, to generate a typical constellation of the complex envelope values for each modulation technique. For both modulation techniques, 32 active users (the HIPPI data lines) were assumed to employ augmented Gold-31 codes. In the MAMSK constellation, shown in Figure 6.1, no envelope values had amplitude of more than 40. Similarly, for the CPFSK constellation, shown in Figure 6.2, no amplitude values exceeded 15. For the CPFSK constellation, the transitions between envelope states are omitted here for clarity.

In order to examine the relative frequency of various envelope values, a histogram of 10,000 simulated transmitted envelope values was generated for each modulation technique, MAMSK and

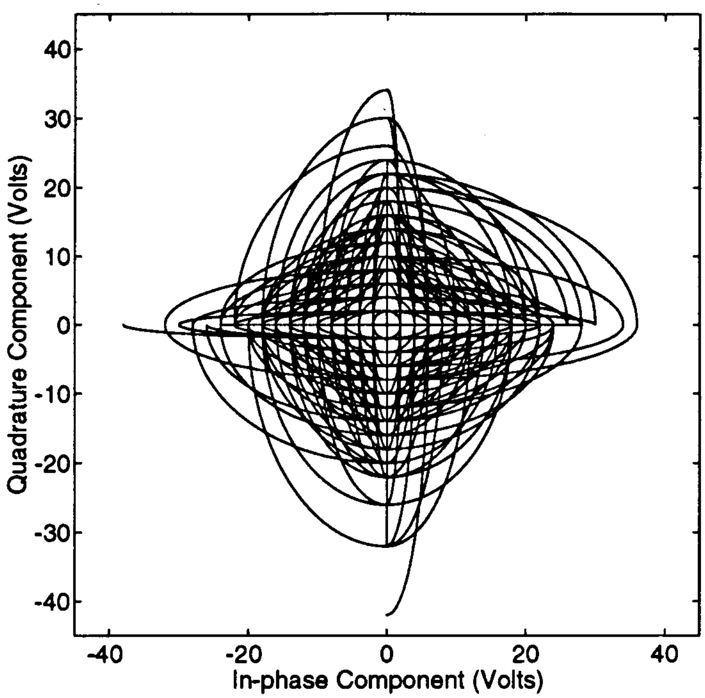


Figure 6.1: A Typical Constellation of Transmitted MAMSK Envelope Values

CPFSK. From the MAMSK histogram (shown in Figure 6.3), MAMSK amplitudes greater than 45 are highly unlikely. Thus, instead of requiring the channel to be linear for all possible MAMSK amplitudes between 0 and 96, the channel must only be approximately linear for amplitudes less than 45. Thus, the required dynamic range for linear amplification is cut approximately in half.

From the CPFSK histogram (shown in Figure 6.4), CPFSK amplitudes greater than 15 are highly unlikely. Since the maximum possible CPFSK amplitude is 32, the required linear dynamic range is again cut approximately in half.

Once the CPFSK envelope of Figure 6.2 is passed through the ACTS nonlinearities, both amplitude and phase are distorted, as shown in Figure 6.5. The figure also includes the effects of additive, white Gaussian noise (AWGN), which causes the smearing of points in the envelope.

Since both Gold and Barker codes require an approximately linear channel, equalization of the nonlinearities and rain attenuation is necessary. After equalization, the envelope becomes that shown in Figure 6.6.

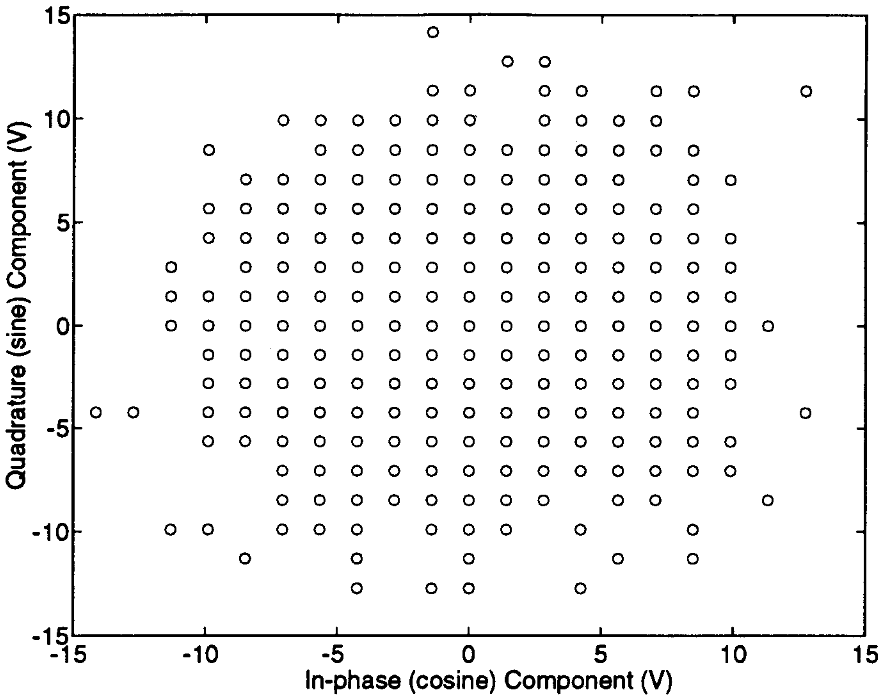


Figure 6.2: A Typical Constellation of Transmitted CPFSK Envelope Values

Because of the noise in the channel, the equalization performed here (which makes no attempt to compensate for noise) is not perfect. For comparison, the transmitted and equalized constellations are plotted on the same axis in Figure 6.7. Although the equalized constellation is not exactly identical to the transmitted constellation, these imperfections in equalization do not necessarily cause bit-errors. Since the spread-spectrum codes both perform correlation detection on many chips (32 for Gold, 56 for Barker), equalization imperfection in only a few chips will cause some decorrelation and cross-talk between the data lines, but not enough degradation to cause a bit-error. Bit-errors only result when a significant percentage of the chips are corrupted.

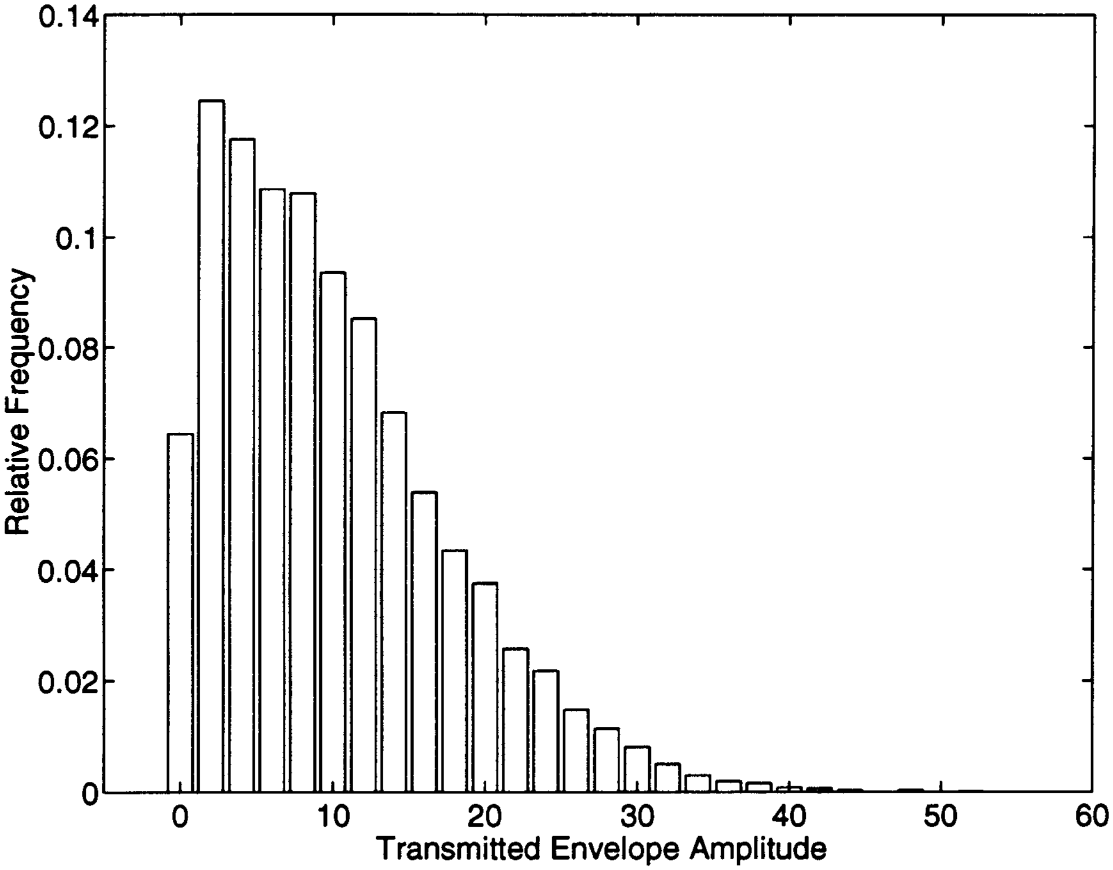


Figure 6.3: Histogram of Transmitted MAMSK Envelope Amplitudes

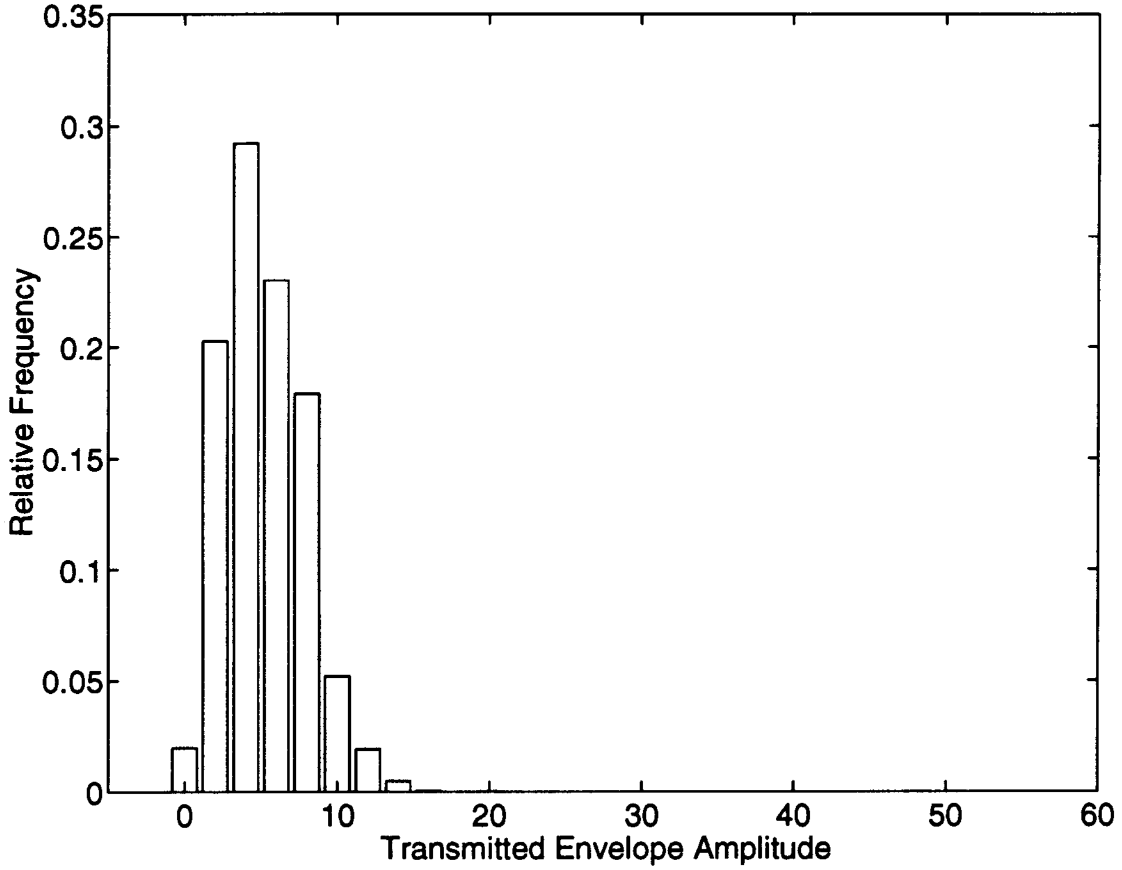


Figure 6.4: Histogram of Transmitted CPFSK Envelope Amplitudes

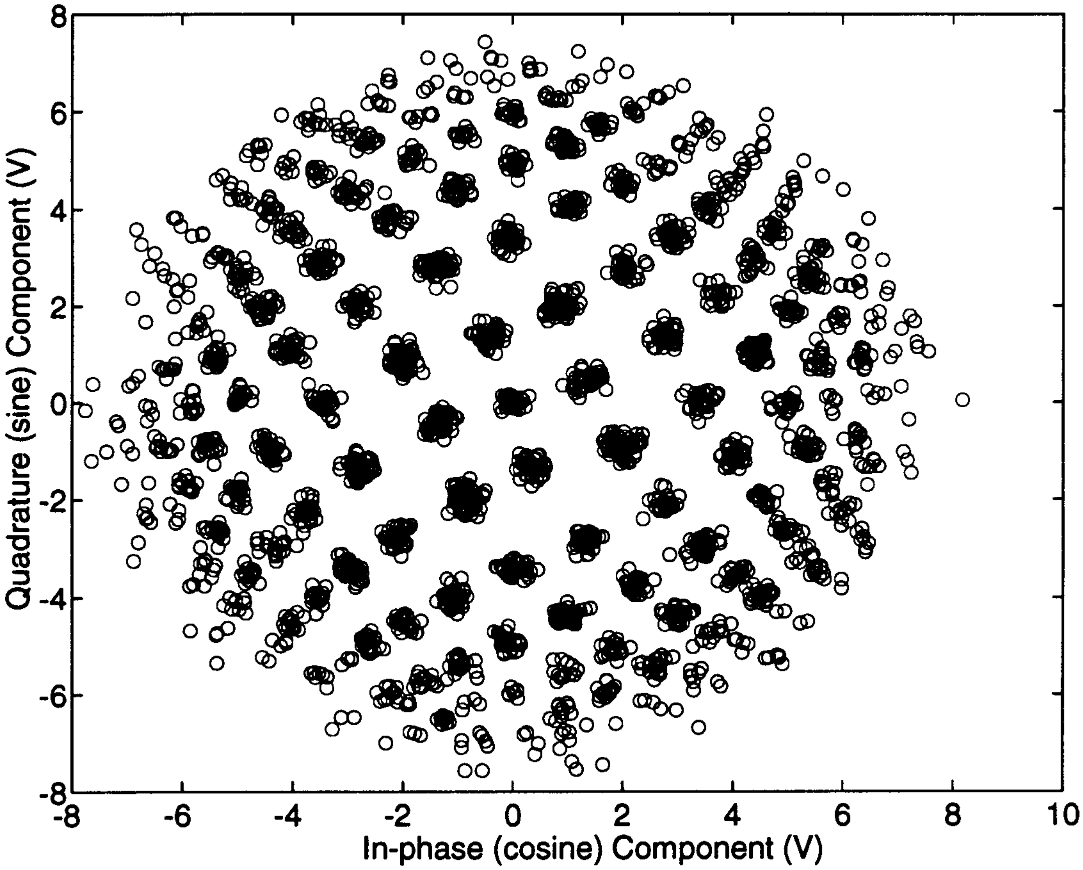


Figure 6.5: The Constellation of Figure 6.2, After Being Corrupted by the ACTS Nonlinearities and Gaussian Noise

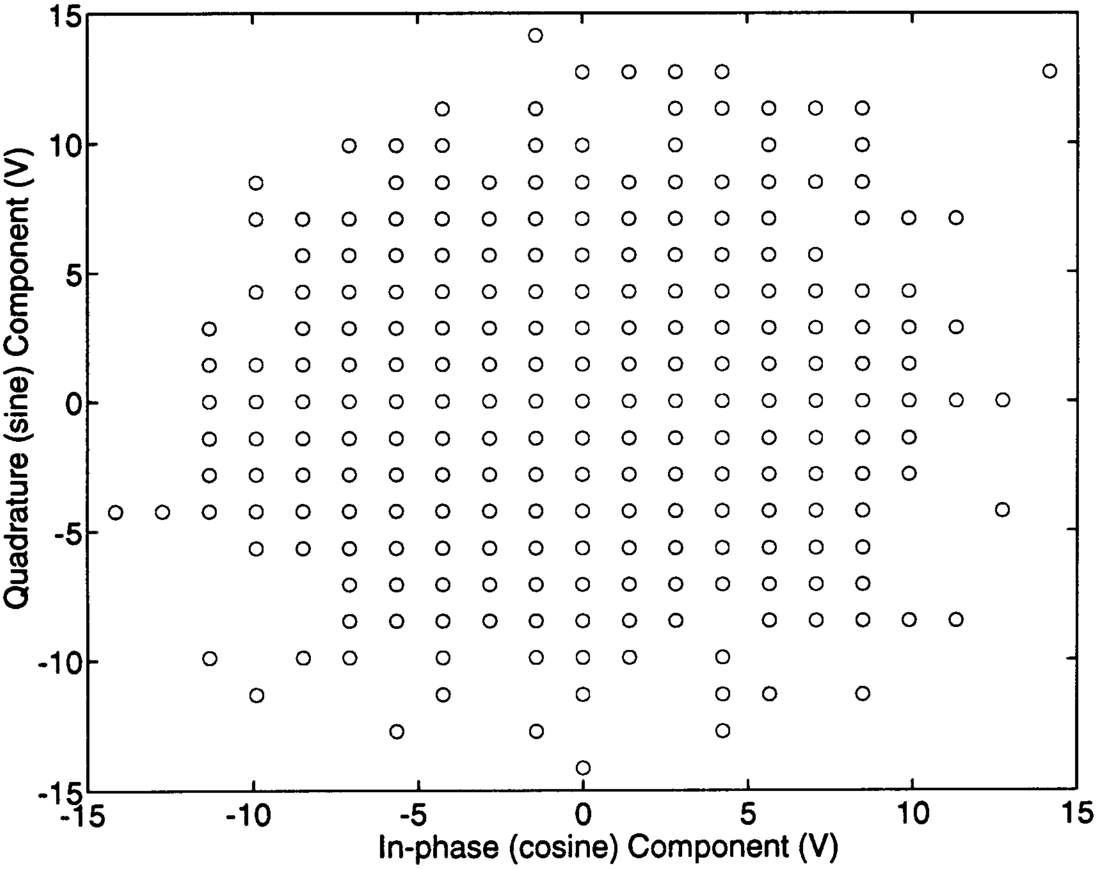


Figure 6.6: The Constellation of Figure 6.5, After Equalization

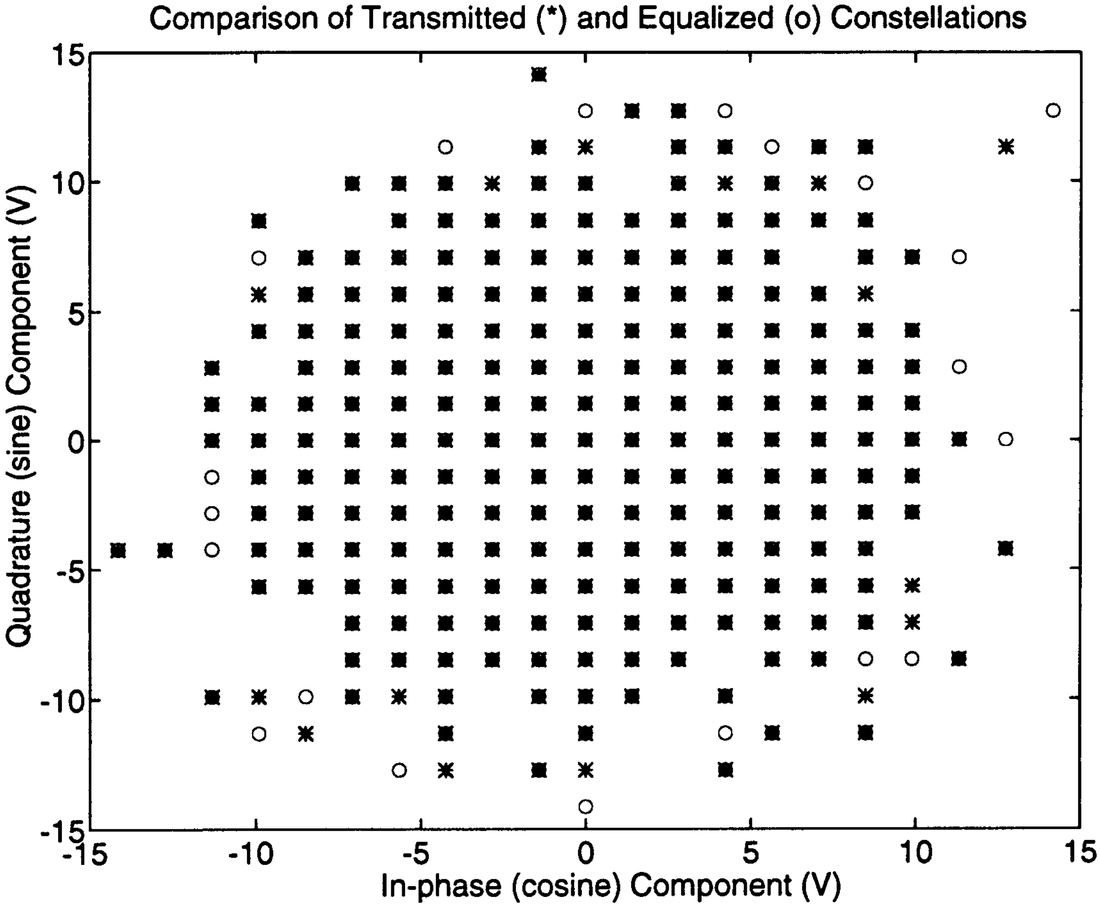


Figure 6.7: Comparison of Transmitted and Equalized CPFSK Constellations

6.2 BIT-ERROR-RATE PERFORMANCE

The bit-error-rate is simulated for all four combinations of modulation technique (MAMSK or CPFSK) and CDMA code (Augmented Gold-31 code or Barker-56 code). In all simulations, 32 users (the HIPPI data lines) are assumed to be active.

6.2.1 IN THE ABSENCE OF RAIN ATTENUATION

The channel was simulated for values of signal-to-noise ratio (SNR) ranging from 0 dB to 20 dB using all four combinations of modulation and spreading technique being investigated. For all cases, 10^5 bits were simulated. When the simulated BER was less than 10^4 (i.e., fewer than 10 errors were made in the 10^5 bits), another simulation of 10^6 bits was conducted to provide a more accurate BER. For the Barker codes, the simulated BER was observed to be on the order of 10^{-6} for 10^6 bits and SNR $>$ 15 dB. Therefore, 10^7 bits were simulated for the Barker codes with SNR $>$ 15 dB. The resulting plot, Figure 6.8, shows the intuitive result that, as the signal becomes larger with respect to the noise, the bit-error-rate decreases. When the SNR exceeded a threshold (approximately 17 dB for CPFSK and 19 dB for MAMSK), no bit errors were simulated for 10^7 bits. This suggests that the BER is less than 10^{-7} for these values of SNR.

6.2.2 IN THE PRESENCE OF RAIN ATTENUATION

Another simulation was conducted in which both amplitude and phase were equalized as before, but rain attenuation was added to the link. No attempt was made to equalize the effect of the rain, resulting in a poor BER performance, even in the complete absence of Gaussian noise. Even for the tiny rain rate of 1 mm/hour, the BER was 6.27×10^{-2} for a simulation of 10^5 bits. Thus, equalization in the form of adaptive gain control is needed to counteract the effects of the rain attenuation.

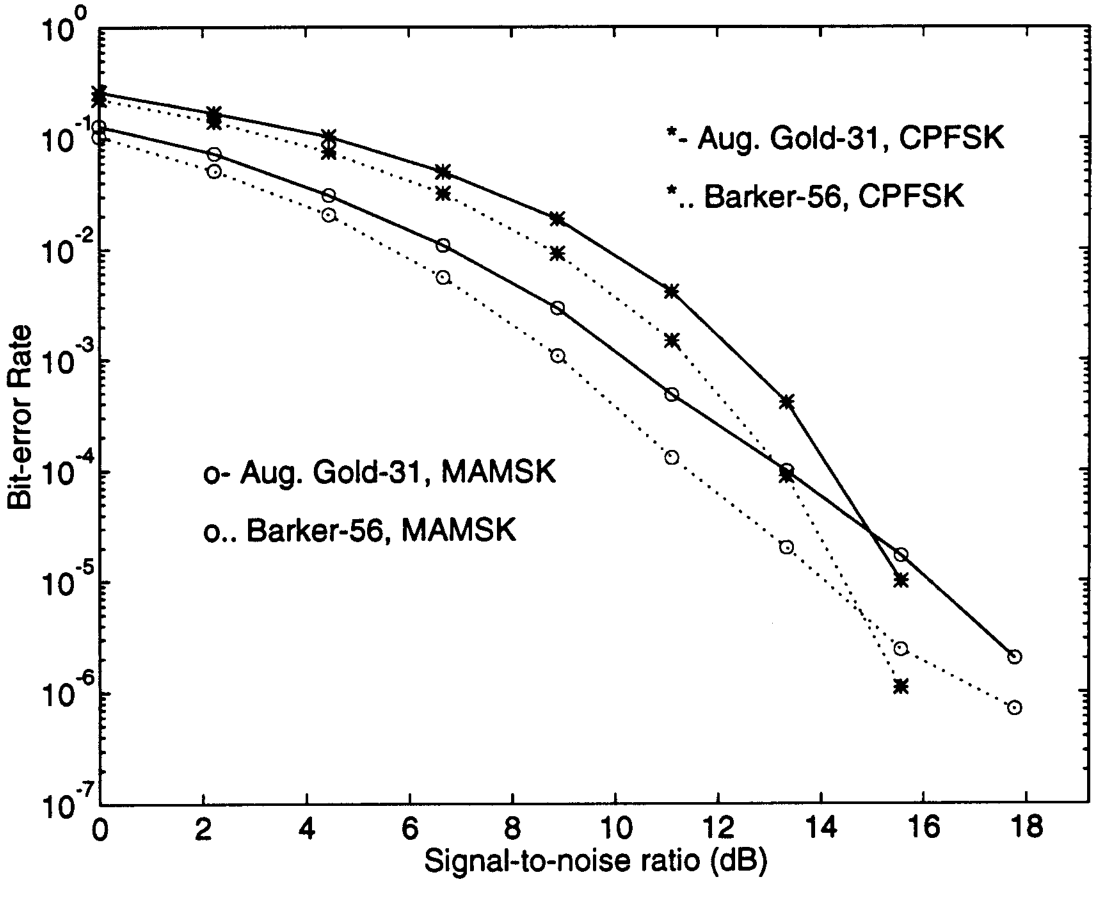


Figure 6.8: Simulated BER for All Four Combinations of Modulation and Spreading Code

Chapter 7

CONCLUSIONS

From the results of the simulation, several conclusions can be made concerning coding, modulation, and propagation as relevant the connection of two HIPPI LANs via the ACTS satellite.

Both CDMA codes, Gold and Barker, provide acceptable performance for the interconnection. For both Gold and Barker codes, it is possible to reduce the amount of mutual interference in a synchronous system (such as the HIPPI interconnection). In transmitting 32 data lines (i.e., 32 CDMA users), the Barker-56 code has a lower bit-error-rate but requires more bandwidth than the augmented Gold-31 code.

Both modulation schemes, MAMSK and CPFSK, provide acceptable performance in the interconnection for reasonable values of signal-to-noise ratio (i.e., SNR values of 17 dB or higher). It is observed that MAMSK outperforms CPFSK for low SNR, but CPFSK outperforms MAMSK for high SNR.

In the ACTS channel, the propagation issues of nonlinear distortion, noise, and rain attenuation are all important. When using CDMA for multiplexing the data lines, the transmitted envelope will not have constant amplitude. Therefore, equalization of the ACTS nonlinearities is essential. Fortunately, very large envelope amplitudes tend to be much less likely than small amplitudes, and there exists a threshold above which the probability of obtaining any amplitude is negligible for both MAMSK and CPFSK. MAMSK and CPFSK require linear amplification over only half of their total dynamic ranges. Since MAMSK's total possible dynamic range is much larger than that of CPFSK (96 vs. 32), MAMSK requires linear amplification over a dynamic range approximately 2.6 times as wide as CPFSK (range of 40 for MAMSK vs. 15 for CPFSK).

Finally, it is possible to obtain an acceptable bit-error-rate using either MAMSK or CPFSK with either Gold or Barker codes for a reasonable signal-to-noise ratio. For all four combinations of modulation and CDMA coding, the simulated BER was less than 10^{-7} for an SNR of 20 dB. Therefore, it is feasible to use the ACTS satellite to connect two HIPPI LANs.

Bibliography

[1] D. Tolmie and J. Renwick, "HIPPI: Simplicity Yields Success," *IEEE Network*, Vol. 7, No. 1, January 1993, pp. 28-32

[2] F. M. Naderi and S. J. Campanella, "NASA's Advanced Communications Technology Satellite (ACTS): An Overview of the Satellite, the Network, and the Underlying Technologies," NASA *ACTS Publication 100*

[3] J. C. Freeman, NASA-Lewis Research Laboratory, personal communication, August 18, 1993.

[4] L. J. Ippolito, "Radio Propagation for Space Communication Systems," *Proceedings of the IEEE*, Vol. 69, June 1981, pp. 697-727

[5] L. J. Ippolito, *Propagation Effects Handbook for Satellite Systems Design– A Summary of Propagation Impairments on 10 to 100 GHz Satellite Links With Techniques for System Design* NASA Reference Publication 1082, 4th edition, 1989

[6] S. Benedetto, E. Biglieri, and V. Castellani, *Digital Transmission Theory* (Englewood Cliffs, N. J.:Prentice-Hall, 1987)

[7] R. E. Ziemer and R. L. Peterson, *Digital Communications and Spread Spectrum Systems* (New York: Macmillian Publishing Company, 1985)

[8] K. E. Toerper, "Biphase Barker-coded Data Transmission," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-4, March 1968, pp. 278-82

[9] W. J. Weber III, P. H. Stanton, and J. T. Sumida, "M-ary MSK Modulation (MAMSK)," *IEEE Transactions on Communications*, Vol. COM-26, May 1978, pp. 543-51

[10] National Aeronautics and Space Administration, *Advanced Communications Technology Experiment Opportunity Announcement*

[11] W. K. Stelzenmuller and E. L. Walker, "Multiplex Data Communications Using Acoustical Surface Wave Filters," United States Patent 3,985,968

[12] W. W. Peterson and E. J. Weldon, *Error Correction Codes* (Cambridge, Mass.: MIT Press, 1972)

APPENDIX

This appendix contains the listings of all MatLab programs used in the simulation of the system.

FSKBER.M

```
function actsber=fskber(TotBits,noise1,rain1,noise2,rain2,spf)
% FSKBER(TotBits,noise1,rain1,noise2,rain2,sp-flag)
%   Simulates sending TotBits bits over the ACTS channel, using CPFSK,
%   measuring the bit-error-rate.
% The flag sp-flag is 0 for Gold codes, 1 for Barker codes.
%   If the flag is omitted, Gold codes are assumed.

BPIS=2;                 % Bits per input symbol.
InAmp=2^BPIS-1;         % Max. PAM amp. of input words
lines=32;               % Number of CDMA lines.  This must be EVEN.
cycles=ceil(TotBits/lines/BPIS);
Root2=sqrt(2);
bo=27;                  % Decibels to back off the ACTS TWTA.
if nargin==5            % If sp-flag was omitted,
   spf=0;               %   assume Gold codes.
end
if spf==0               % If want Gold codes...
   mt=[gold31 ones(33,1)];      % Augmented Gold code transmit-matrix
   mr=mt;                       % Augmented Gold code receive-matrix.
   AutoCorr=size(mt,2);         % Aug. Gold code's auto-correlation
  else                  % If want Barker codes...
   [mt,mr]=bark52aug;           % Aug. Barker-52 trans. and rec. matrices.
   AutoCorr=48;                 % Aug. Barker-52 has auto-corr. of 48.
end
ActsP=actstwt;                  % Polynomial approximation of ACTS Amplitude
ActsIP=actstinv;                % Polynomial to equalize ACTS Amplitude
SpLen=size(mt,2);               % Length of the spreading codes.
RefPh=exp(j*pi*[0.25; 0.75]);   % The reference phases, 45 and 135 degrees,
RefPhC=conj(RefPh);             %   and their conjugates.
MaxBlkLen=100;                  % Number of CYCLES to simulate at once.
                                %   A longer block-len requires more memory.

NoBlks=ceil(cycles/MaxBlkLen);  % No. of blocks needed.
berrs=0;                        % No. of measured bit-errors
for blk=1:NoBlks
   if blk==NoBlks
      blklen=cycles-(NoBlks-1)*MaxBlkLen;
     else
      blklen=MaxBlkLen;
   end

   inb=2*floor(rand(lines,blklen)*(InAmp+1))-InAmp;    % Random input words

   % Instead of generating the whole envelope, only generate its
   %   values at the sample times...
   in1=inb.*(abs(inb)==1);              % Separate the 1's and 3's
   in3=inb.*(abs(inb)==3);
   gin1=goldin(mt,in1);
   gin3=goldin(mt,in3);         % gin = gin1 + gin3;
   et=RefPh.'*[gin1(:) gin3(:)/3]';
   er=acts(et,noise1,rain1,noise2,rain2,ActsP,bo);

   % Now correlate the envelope-samples with the reference phases.
   erenv=10*log10(abs(er)+eps);
   erenv=polyval(ActsIP, erenv-bo)+bo;  % Amplitude equalization
   er2=10.^(erenv/10).*exp(j*angle(er));
   er2=er2.*exp(-j*1/20*(erenv-bo));    % Phase equalization

   er2=er2-(abs(er2)>9).*exp(j*angle(er2));     %Use for BO=27 only.

   er2=Root2*round(er2/Root2);
   rcorr=real(RefPhC*er2);      % Correlation with both reference phases
   chips=[1 3]*rcorr;           % Convert to chip-values
   chips=reshape(chips,SpLen,size(chips,2)/SpLen)';

   r=mr*chips'/AutoCorr;
   r=r(1:lines,:);

   outb=2*floor(r/2)+1;
   bitdiff=inb-outb;
   berrs=berrs+sum(sum(bitdiff~=0) +sum(abs(bitdiff)==4));
end
actsber=berrs/lines/cycles/BPIS;
```

MSKBER.M

```
function mskber=mskber(TotBits,noise1,rain1,noise2,rain2,spf)
% MSKBER(TotBits,noise1,rain1,noise2,rain2,sp-flag)
%   Simulates sending TotBits bits over the ACTS channel, measuring
%   the bit-error-rate, using MAMSK modulation.
% The flag sp-flag is 0 for Gold codes, 1 for Barker codes.
%   If the flag is omitted, Gold codes are assumed.

BPIS=2;                 % Bits per input symbol.
InAmp=2^BPIS-1;         % Max. PAM amp. of input words
lines=32;               % Number of CDMA lines.  This must be EVEN.
cycles=ceil(TotBits/lines/BPIS);
bo=30;                  % Decibels to back off the ACTS TWTA.
noise1=2*noise1;        % Since MSK is backed off 3 dB more than FSK,
noise2=2*noise2;        %  double the noise variance for a fair comparison.
if nargin==5            % If sp-flag was omitted,
   spf=0;               %   assume Gold codes.
end
if spf==0               % If want Gold codes...
   mt=[gold31 ones(33,1)];      % Augmented Gold code transmit-matrix
   mr=mt;                       % Augmented Gold code receive-matrix.
   AutoCorr=size(mt,2);         % Aug. Gold code's auto-correlation
  else                  % If want Barker codes...
   [mt,mr]=bark52aug;           % Aug. Barker-52 trans. and rec. matrices.
   AutoCorr=48;                 % Aug. Barker-52 has auto-corr. of 48.
end
ActsP=actstwt;                  % Polynomial approximation of ACTS Amplitude
ActsIP=actstinv;                % Polynomial to equalize ACTS Amplitude
SpLen=size(mt,2);               % Length of the spreading codes.
RefPh=exp(j*pi*[0.25; 0.75]);   % The reference phases, 45 and 135 degrees,
RefPhC=conj(RefPh);             %   and their conjugates.
MaxBlkLen=100;                  % Number of CYCLES to simulate at once.
                                %   A longer block-len requires more memory.

NoBlks=ceil(cycles/MaxBlkLen);  % No. of blocks needed.
berrs=0;                        % No. of measured bit-errors
for blk=1:NoBlks
   if blk==NoBlks
      blklen=cycles-(NoBlks-1)*MaxBlkLen;
     else
      blklen=MaxBlkLen;
   end

   inb=2*floor(rand(lines,blklen)*(InAmp+1))-InAmp;     % Random input words

   % Instead of generating the whole envelope, only generate its
   %   values at the sample times...
   gin=goldin(mt,inb);

   % The values of gin correspond to the amplitudes on the trellis.
   % Consecutive bits of "in" run down the matrix's columns.
   % However, since this is Type I MSK, alternate values of in for each channel
   %   are multiplied by (-1) before they appear on the trellis.
   in2 = reshape(gin,2,SpLen*blklen/2);
   flip=sign(rem([1:SpLen*blklen/2],2)-.5);
   ch1 = in2(1,:) .* flip;         % Mult. alternate values by -1.
   ch2 = in2(2,:) .* flip;
   et=[ch1; j*ch2];
   et=et(:).';
   er=acts(et,noise1,rain1,noise2,rain2,ActsP,bo);

   % Now correlate the envelope-samples with the reference phases.
   erenv=10*log10(abs(er)+eps);
   erenv=polyval(ActsIP, erenv-bo)+bo;  % Amplitude equalization
   er2=10.^(erenv/10).*exp(j*angle(er));
   er2=er2.*exp(-j*1/20*(erenv-bo));    % Phase equalization
   er2=round(er2);
   in2=reshape(er2,2,SpLen*blklen/2);
   ch1 = real(in2(1,:)) .* flip;         % Mult. alternate values by -1.
   ch2 = imag(in2(2,:)) .* flip;         % and correlate with ref. phases
   chips=[ch1; ch2];
   chips=reshape(chips,SpLen,blklen)';

   r=mr*chips'/AutoCorr;
   r=r(1:lines,:);

   outb=2*floor(r/2)+1;
   bitdiff=inb-outb;
   berrs=berrs+sum(sum(bitdiff~=0) +sum(abs(bitdiff)==4));
end
mskber=berrs/lines/cycles/BPIS;
```

BARK52AUG.M

```
function [b52t, b52r]=bark52aug();
% BARK52AUG: [trans, rcvr]=bark52aug;
% Returns the augmented Barker-52 matrix for both the transmitter and
% the receiver (they are not identical), built using a
% 4x4 supermatrix of the Barker-13 matrix, which has been augmented
% to eliminate crosstalk.
% The signs of the Barker-13 matrices are given by the
% Barker-4 matrix.

b13v=[1 1 1 1 1 -1 -1 1 1 -1 1 -1 1];
b13=b13v;
for row=2:13
   b13=[b13; b13v(row:13) b13v(1:row-1)];
end
b13=[b13 ones(13,1)];
b52t=[-b13 b13 b13 b13; b13 -b13 b13 b13; ...
      b13 b13 -b13 b13; b13 b13 b13 -b13];
b13=[b13(:,1:13) -ones(13,1)];
b52r=[-b13 b13 b13 b13; b13 -b13 b13 b13; ...
      b13 b13 -b13 b13; b13 b13 b13 -b13];
```

GOLD31.M

```
function b=gold31
% This function generates the lengh-31 gold codes as the rows of the matrix b,
%    whose elements are either +1 or -1.
% Since the second generating sequence (b2)'s cross-correlation is slightly
% worse than the rest of the sequences, b2 is put in the last row of the matrix.

len=31;
p1=octbin(45);          % The generating polynomials (coefficients in octal)
p2=octbin(75);
b1=regout(p1);          % The output sequences corresponding to p1 and p2.
b2=regout(p2);
b1=sign(b1-.5);         % Make sure b1 & b2 contain +1 and -1 instead of
b2=sign(b2-.5);         %   +1 and 0.
b=[b1;zeros(len+1,len)];
for row=2:len+1;
   b(row,:)=-b1 .* [b2(row:len) b2(1:row-1)];   % This line performs XOR.
end
b(len+2,:)=b2;
```

OCTBIN.M

```
function bin=octbin(Goct)
% This function converts a nonnegative octal integer to binary, with one bit
% per element of a vector.  The octal input number must contain 16 or fewer
% digits, because of the accuracy of the arithmetic.

G=[];
if Goct == 0    % Zero is a special case.
   G=0;
 else
   while Goct>0
      r=rem(Goct,10);
      g2=floor(r/4);
      r=r-4*g2;
      g1=floor(r/2);
      g0=r-2*g1;
      G = [g2 g1 g0 G];
      Goct=floor(Goct/10);
   end
   if g2==0
      G = G(2:size(G,2));
      if g1==0
         G = G(2:size(G,2));
      end
   end
end
bin=G;
```

REGOUT.M

```
function regout=regout(G)
% This function calculates the output of a shift-register, given the generator
% polynomial, G.

r=size(G,2)-1;
A=zeros(1,r);
A(r+1)=1;               % Initial condition of the register is A=[0 0 ... 0 1]
mem=2^r-1;
out=zeros(1,mem);
B=0;
for t=1:mem
  B=A(r+1);
  if B==1
     BG=G;
    else
     BG=zeros(1,r+1);
  end
  An=xor(BG,A);
  out(t)=B;
  A=[0 An(1:r)];
 end
 regout=out;
```

GOLDIN.M

```
function goldout=goldin(B,inputs)
% GOLDIN(B, INPUTS)
% This function generates the output of several Gold sequences talking at once.
% It returns a column vector containing the superposition of the sequences.
% B is the matrix whose rows contain the Gold sequences.
% Inputs is a column-vector (with the same number of rows as B).
% Inputs will contain a combination of +1, -1, and 0, which corresponds to the
% inputs of each channel (i.e., the rows of B).
% If inputs has fewer rows than required, it will be padded with zeroes.

inputs = [inputs; zeros((size(B,1)-size(inputs,1)), size(inputs,2))];
goldout= sign(B-0.5)'*inputs;
```

LINK.M

```
function out=link(in,fc,noisev,rain)
% Link(input-bit-envelope-vector, carrier-freq, noise-variance, rain-rate)
% corrupts the bits by rain & AWGN.
% Carrier freq. is in GHz, rain-rate in mm/hr.

n = sqrt(noisev)*(randn(size(in))+j*randn(size(in)));
ratt = rainatt(fc,rain);
latt = 1;               % Link attenuation (since it's constant, ignore it).
out = in*ratt*latt+n;
```

RAINATT.M

```
function att=rainatt(f,rain)
% Rainatt(freq,rain-rate) calculates the rain attenuation (in absolute units,
% not dB) given the carrier frequency (GHz) and the rain rate (mm/hr).
% Carrier frequency must be between 8.5 and 54 GHz.

ElevAngle=40*pi/180;            % Satellite elevation angle (rad).
L=4;                            % 4 km is approx. path length for rain.
a = 4.21e-5 .* f.^2.42;         %Valid for 2.9 < f < 54 GHz
if f < 25
   b = 1.41 .* f.^(-0.0779);    % 8.5 < f < 25 GHz
  else
   b = 2.63 .* f.^(-0.272);     % 25 < f < 164 GHz
end
attdb = L .*a .*rain.^b /sin(ElevAngle);
att= 10^(-attdb/10);
```

ACTS.M

```
function out=acts(in,noisev1,rain1,noisev2,rain2,AmpPoly,bo)
% Acts(input-bit-envelope-vector, uplink-noise-variance, uplink-rain-rate,
%    downlink-noise-var., downlink-rain-rate,
%    AM-AM distortion's generating polynomial
%    Back-off [in dB])
% simulates the whole ACTS channel, returning the received bits.
% Carrier freq. is in GHz, rain-rate in mm/hr.

% Uplink...
out1=link(in,30,noisev1,rain1);         % Uplink

% ACTS...
cenv=10*log10(abs(out1)+eps)-bo;        % Envelope, in dB.
csmall=(cenv<-10);                      % Subtr. amp. back-off from cenv. &
cenvdb=cenv.*csmall -10*(~csmall);      %  limit the maximum, because the
                                        %  polynomial is invalid  > -10 dB.
crenv=polyval(AmpPoly,cenvdb);
crenv=crenv+bo;         % Amplify at receiver.
ph=1/20*cenv; %+1.18;   % Phase disto., radians, from p. 540 in Benedetto
                        %  (normalized so that ph=0 at in=0 dB).
in2=10.^(crenv/10).*exp(j*angle(out1)).*exp(j*ph);

% Downlink...
out=link(in2,20,noisev2,rain2);         % Downlink
```

ACTSTWT.M

```
function p=actstwt
% ACTSTWT: Fit a polynomial to the ACTS TWT transfer curve
% IMPORTANT: All values here are in dBm.

in=-28:3:-4;    % Data points from the curve (in dBm)
out=[-29.3 -25 -21.6 -19.4 -18.3 -17.7 -17.6 -17.5 -17.5];

p=polyfit(in, out, 3);
```

ACTSTINV.M

```
function p=actstinv
% ACTSTINV: Fit a polynomial to the INVERSE of the ACTS TWT transfer curve
% IMPORTANT: All values here are in dBm.

in=-28:3:-4;    % Data points from the curve (in dBm)
out=[-29.3 -25 -21.6 -19.4 -18.3 -17.7 -17.6 -17.5 -17.5];

p=polyfit(out(1:6), in(1:6), 3);
```