LEWIS GRANT
IN-32-CR
333456
P79

Final Report
for
Grant NAG 3-982
covering the period
1/1/90 to 9/30/90

Dr. Robert E. Peile
Prof. Loyd Welch
Principal Investigators
Department of Electrical Engineering
University of Southern California, Los Angeles, CA 90089-0272

(NASA-CR-187961) STUDY OF A CO-DESIGNED DECISION FEEDBACK EQUALIZER, DEINTERLEAVER, AND DECODER Final Report, 1 Jan. - 30 Sep. 1990 (University of Southern California) 79 p
N91-18332
CSCL 17B H1/32
Unclas
0333456

Study of a Co-designed Decision Feedback Equalizer, Deinterleaver and Decoder[1].

Dr. Robert E. Peile and Prof. Loyd Welch
Principal Investigators

Department of Electrical Engineering
University of Southern California, Los Angeles, CA 90089-0272

[1]This work was supported in part by NASA Lewis Research Center under Grant No. NAG 3-982.

Abstract

Here we present a technique that promises better quality data from Band limited channels at lower received power in digital transmission systems. Data transmission, in such systems often suffers from *Intersymbol Interference* (ISI) and noise. Two separate techniques: channel coding and equalization have caused considerable advances in the state of communication systems and both concern themselves with removing the undesired effects of a communication channel. Equalizers mitigate the ISI whereas coding schemes are used to incorporate error-correction. In the past, most of the research in these two areas has been carried out separately. However, the individual techniques have strengths and weaknesses that are complementary in many applications; an integrated approach realizes gains in excess to that of a simple juxtaposition. Coding schemes have been successfully used in cascade with Linear Equalizers which in the absence of ISI provide excellent performance. However, when both ISI and the noise level are relatively high, non-linear receivers like the Decision Feedback Equalizer (DFE) perform better. The DFE has its drawbacks; it suffers from *error propagation.* The technique presented here takes advantage of interleaving to integrate the two approaches so that the error propagation in DFE can be reduced with the help of error correction provided by the Decoder. The results of simulations carried out for both, binary and non-binary, channels confirm that significant gain can be obtained by *Codesigning* Equalizer and Decoder. Although, we have looked into systems with time-invariant channels and with simple DFE having linear filters, the technique is fairly general and can be easily be modified for more sophisticated equalizers to obtain even larger gains.

Contents

Chapter 1

Introduction

Here we are interested in synchronous, linearly modulated data transmission systems with band limited Additive White Gaussian Noise (AWGN) channels. Bandlimited channels often have non-flat frequency characteristics that result into Inter-Symbol Interference (ISI). In systems with ISI, the transmitted pulse interferes with a number of past and future pulses. For systems with negligible ISI, coded modulation schemes [1] [2] can reduce the effect of WGN, effectively. On the other hand, Equalization is extremely useful to mitigate the ISI. On channels with significant amount of noise and ISI a simple cascade of coding and Equalization does not produce encouraging results. Recently schemes [3] [4] [5] [6] [7] have been investigated in which coding and Equalization is combined in an effective manner to improve the overall performance. The codesigned coding/equalization system can be described within the general structure of Figure 1.1. In this paper we present yet another technique which is closer in nature to the one presented by Eyuboglu [3] [4] (which we will refer to as Eyuboglu's Codesigned Receiver), but is quite novel in structure. Its implementation (at least in its current form) is more suitable for block codes, has a simpler structure and is expected to out perform the Eyuboglu's Codesigned Receiver.

Nyquist [8] presented the first important result for uncoded systems. He determined sufficient conditions for eliminating linear ISI at sampling instants (the Zero Forcing (ZF) condition), for Bandlimited channels. The Nyquist criterion gives the designer considerable freedom in designing Transmitter and Receiving Filters. Although he derived the condition disregarding the noise, the condition gives best possible scheme (in the sense that it minimizes probability of Error) even in the presence of AWGN under the constraint that all processing related to compensation of the channel, is done at the transmitter. This is precisely the solution of the problem of designing the transmitting and receiving filters by minimizing the noise variance at the input of the decision device while keeping the Nyquist criterion. The main idea is to pre-distort the signal at the transmitter by an *inverse* channel filter so that the signal at the output of the channel appears undistorted. Tomlison precoder [9] does precisely the same in a more power-effective manner. This approach has been combined with coding in [5] [7] and most recently by Eyuboglu and Forney [6]. It inherently assumes the complete knowledge of the channel at the transmitter, which is only partly true if at all, for many application. Another innovative approach which also falls in the class of coded systems where channel is known to the transmitter, was introduced by Sanjay et al [10].

We are primarily interested in the systems where transmitter does not have the knowledge of the channel so that preprocessing can not be performed to satisfy the ZF condition. In this direction a long time after Nyquist's contribution, research was mainly carried-out on assuming a receiver structure which consisted of a linear signal processor followed by a threshold device, referred to as a Linear Equalizer (LE). If this LE is designed so that the system satisfies the Nyquist (ZF) criterion, noise is enhanced at frequencies where channel characteristics have nulls. This degrades the performance considerably. Furthermore, a finite-length ZF Equalizer is guaranteed to minimize the worst case ISI (or peak distortion) only if the peak distortion is less than 100 percent prior to equalization (open eye condition). A better LE is the one that minimizes the Mean Square Error (MSE) [11] between the equalizer output and the corresponding input data symbols. The MSE LE is a compromise; it allows some ISI to pass through so as to reduce the noise enhancement. In most cases one is interested in reducing the Probability of Symbol Error (PSE) rather than MSE. Aaron and Tufts [12] and Yao [13] obtained the LE that minimizes PSE. Finding the system parameters requires considerable numerical effort, however.

Better receivers for systems with severe ISI can be obtained by removing the linearity constraint upon the receiver. Abend and Frichman [14] (also see [15] section 6.6) presented the receiver that is optimum in the sense that it minimizes PSE. The receiver obtained is highly non-linear, is parametric in the sense that the probability density of the noise must be known and is quite complex. The next best (known) receiver in the class of non-linear receivers, is the Maximum Likelihood Sequence Estimator (MLSE); its recursive version is known as Viterbi Detector/estimator (VD) [16] [17]. The VD is simpler than the minimum PSE receiver, but is still quite complex when the size of the signal alphabet M and the memory L of the channel are large (complexity $\propto M^L$). Several techniques for reducing the complexity of VD have been looked into [18]-[24]. Out of these the techniques, the one presented in [24] is most flexible and general.

Another very simple non-linear Equalizer is the Decision Feedback Equalizer (DFE). It has two linear transversal filters. A Feed-Forward Filter (FFF) and a Feed-Back Filter (FBF). The received sequence is first passed through the FFF. The FBF uses the previously detected symbols and forms a replica of the ISI which is then subtracted from the output of the FFF. The difference is then presented to the decision device. For many applications the DFE attains almost the same performance as the VD. The DFE just like its Linear cousin, the LE, can be optimized using either the ZF or the MSE criteria. A ZF DFE which operates with original transmitted data fed to the FBF, is called *ideal* DFE. The usual coded modulation schemes [1] [2] can be used in a simple cascade with the ideal DFE to produce excellent performance. In the practical situations, the exact transmitted sequence is not known to the receiver so ideal DFE is not feasible. A real DFE puts its own decisions in the FBF. Thus, presence of an incorrect decision in the FBF increases the probability of subsequent errors. This phenomenon is known as error propagation.

The error propagation could be reduced in coded systems if it were possible to place a decoder in the feedback path. However, in coded systems decoded decisions are available only after a finite delay which is undesirable because often the decoded decisions that are needed most are the ones immediately preceding the currently processed symbol. Eyuboglu's Codesigned receiver uses interleaving to generate the necessary delay between all but one adjacent symbols in a block. In his scheme the number of symbols *available* to the FBF (these are the decoded symbols) varies between 0 and $P-1$ with time, where P is the size of a interleaved block. This is precisely the reason that forced him to use only the decoded data in the FBF, and therefore justified the use of *noise-predictive* DFE [25] which has the advantage that its FFF is independent of the number of Feed Back coefficients. Noise-predictive DFE is an alternative to conventional DFE. The two are equivalent as long as both have infinitely long FFFs and have FBFs of equal length [26]. However, when FFFs are short the conventional DFE outperforms its noise-predictive counterpart.

Our codesigned receiver uses a particular type of interleaving called helical interleaving. The structure of the receiver is such that undecoded past decisions can also be used in the FBF along with the decoded decisions so that we can provide the FBF with as many decisions as desired. Therefore, we can use, the superior, conventional DFE without hesitation. The number of the decoded data in the FBF is still the same as that in Eyuboglu's approach, however, the availability of the undecoded data in the FBF improves the equalization process.

Chapter 2

System Model

In this report we will consider two types of data transmission systems: one with channel encoding and the other without. The former will be referred to as the *coded system* and the letter as *uncoded system*. For both systems, we assume that the source produces an independent identically distributed random binary sequence $\{a_l\}$ with $a_l \in \{-1, 1\}$. We assume that the input sequence is semi-infinite i.e. $a_l \neq 0$ only for $l \geq 0$. We will first discuss the transmission end of the coded system model with reference to the figure 1.1.

2.1 Coded System

The encoder takes K consecutive symbols of $\{a_k\}$ and produces N output symbols. We are primarily interested in block codes. We denote the output symbol of the encoder by c_{ij} which is the jth symbol of the ith code word. The symbols c_{ij} are not necessarily binary, and the Space they belong to depends upon the particular encoding scheme used. For our encoding scheme c_{ij} are integers such that $c_{ij} \in \{0, 1, \ldots, 7\}$. The encoder also introduces synchronization (sync) symbols which are in fact c_{i0}. The actual ith codeword is $[c_{i1}, c_{i2}, \ldots, c_{i(N-1)}]$. Note that c_{ij} occurs at time $Ni + j$ and that $0 \leq j \leq (N-1)$. The condition on the input time index $l \Rightarrow c_{ij} \neq 0$ only for $i \geq 0$. The details of the particular encoder used are presented in Chapter 3.

2.1.1 Interleaver

The encoded symbols are interleaved by the helical interleaver that delivers these symbols at time k to the output (denoted by c_k) according to following rule:

$$k = Ni + (N-1)j \quad ; 0 \leq j \leq (N-1). \tag{2.1}$$

Note that for sync symbols $j = 0, \Rightarrow k = Nj$ which indicates that the sync symbols are not delayed relative to input. For c_{i1}, $k = Ni + N - 1$ which indicates that it is delayed by $(N-1)$ symbol intervals. Similarly, jth symbol in any codeword will be delayed by $(N-1)j$ symbol intervals. The operation of Interleaver for case $N = 4$ is shown in figure 2.1

We can think of the interleaver as a device that performs a mapping I from the input time indexes (i, j) to the output time index. It follows from above that $i \in Z^+ \triangleq \{$ non-negative integers $\}$ and $j \in S \triangleq \{0, 1, \ldots, (N-1)\}$ so that $(i, j) \in Z^+ \times S$. Obviously, $k \in Z^+$. Then the mapping I is

$$I : Z^+ \times S \rightarrow Z^+$$

explicitly, $I(i, j) = k = Ni + (N-1)j$. We now proceed to show that the mapping I is *injective* (for every allowed pair (i, j) there is a unique k).

Theorem 1 *The mapping I as defined above is injective.*
Proof
Assume it is not injective then there are $(i, j), (i', j') \in Z^+ \times S$ such that $k = k'$ where $k = Ni + (N-1)j$ and $k' = Ni' + (N-1)j'$. Without loss of generality we can assume $i \geq i'$.

$$\begin{aligned} Ni + (N-1)j &= Ni' + (N-1)j' \\ N(i - i') + (N-1)(j - j') &= 0 \end{aligned}$$

If $i > i'$ we have to have $j < j'$ in order for the above to hold. The only possibility for the above to be zero is $i - i' = (N-1) \times n$, where n is an integer and $j - j' = -N \times n$. The later is impossible due to our constraint on js. Therefore, $i = i'$ and $j = j'$. ♣

We are also interested in *surjection (onto)* (i.e. every time index $k = 0, 1, 2, \ldots$ is a result of some input time index pair (i, j)) of the mapping. For semi-infinite sequences (sequences that start at time, $k = 0$) this is not strictly true e.g. for $k = 1$ there is no such pair (i,j) such that $i \geq 0$, $0 \leq j, j' \leq N-1$ and $k = Ni + (N-1)j$. We will now show that the mapping is surjective for $k \geq (N-1)(N-2)$.

Theorem 2 *The mapping I is surjective for* $k \geq (N-1)(N-2)$.

Proof

The proof follows from induction. For $k = (N-1)(N-2)$, $i = 0$ and $j = (N-2)$ work. Let the statement be true for a $k > (N-1)(N-2)$. Then there exist (i,j) such that

$$\begin{aligned} k &= Ni + (N-1)j \\ k+1 &= Ni + (N-1)j + 1 \\ \text{if} \quad 1 \leq j \leq (N-1); \quad k+1 &= Ni + (N-1)j + N - (N-1) \\ &= N(i+1) + (N-1)(j-1) \end{aligned}$$

i.e. a pair$(i+1, j-1)$exists.

$$\begin{aligned} \text{if} \quad j = 0; \quad k &= Ni \\ \text{since} \quad k &> (N-1)(N-2) \\ \Rightarrow k &\geq N(N-2) \\ \Rightarrow i &\geq (N-2) \end{aligned}$$

then k can rewritten as

$$\begin{aligned} k &= N(i-(N-2)) + N(N-2) \\ k+1 &= N(i-(N-2)) + N(N-2) + 1 \\ &= N(i-(N-2)) + (N-1)(N-1) \end{aligned}$$

i.e a pair $((i-(N-2)), (N-1))$ exists. ♣

The facts that the index mapping performed by the interleaver is injective and surjective ($\Leftrightarrow$ bijective: for $k \geq (N-1)(N-2)$) implies that the output is a continuous stream of symbols without any gaps (for $k \geq (N-1)(N-2)$) or repetitions.

2.1.2 Modulator

The Modulator maps every symbol in the interleaved sequence $\{c_k\}$ to a point in the complex plane. Its output is a sequence of complex symbols.

It can in fact correspond to assigning parameters to any two mutually orthogonal signals. In our case the real axis of the complex plane corresponds to the amplitude of cosine wave and the imaginary axis to that of the sine wave. This kind of mapping of signal to points in the complex plane (or the *Constellation*) is used e.g. in Phase Modulation (PM), the Amplitude Modulation (AM) or the combination of the two called Quadrature Amplitude Modulation (QAM). The AM corresponds to the situations where information is transmitted over one carrier such as a cosine wave alone. In this case we can map the interleaved symbols to the real line only. The kind of Modulation scheme chosen effects the choice of coding scheme employed and vice versa. In the current system that we discuss in detail, we are concerned with the PM systems. We will represent the output of the modulator by d_k. Details of the encoder and specific modulation scheme used will be presented in Chapter 3.

2.2 Uncoded System

We now briefly describe the transmission end of our model for the uncoded system shown in Figure 1.2. The source is identical to that in the coded system. The modulator now takes p consecutive binary source bits and maps them to a point in the signal constellation. The output to the modulation is represented by d_k.

2.3 Channel

Although, in many physical systems, the actual transmission takes place over *Band Pass* Channels, they can conveniently be modeled by Equivalent Low Pass Channels [15]. Furthermore, prior to sampling most channels are continuous. The continuous channels also require transmission and receiving filters to restrict the signal to required Bandwidth and its efficient reception. We simplify our model by including these filters and the sampler into the channel model. By doing this, we also assume that the channel is synchronous. Thus our channel is the Equivalent Discrete low pass Channel (EDC). The EDC is assumed to be stationary (which can be relaxed for later discussion), it introduces InterSymbol Interference (ISI) which for most of our discussion will be assumed to be *linear*. The channel also introduces Additive White Gaussian Noise (AWGN). The output to the channel is represented by r_k.

If g is the function performed by the channel upon the vector $d(k) = [d_{k+L_1}, \ldots, d_{k+1}, d_k d_{k-1} \ldots d_{k-L_2}]$ then

$$r_k = g(d(k)) + n_k \tag{2.2}$$

is the output of the channel, where n_k is complex AWGN. If g is linear

$$r_k = \sum_{i=-L_1}^{-1} g_i d_{k-i} + g_0 d_k + \sum_{i=1}^{L_2} g_i d_{k-i} + n_k, \tag{2.3}$$

where $\{g_k\}$ is called channel coefficient sequence. The first term in (2.3) will be referred to as pre-cursor ISI, traditionally $g_0 d_k$ is called the signal component, the third term is post-cursor ISI while the last term is the noise component. Like-wise L_1 and L_2 are called pre-cursor and the post-cursor memories, respectively. Note that we have assumed non-causal channel impulse response which is equivalent to disregarding any channel delay in the EDC model. The total memory of the channel is $L = L_1 + L_2$. Both systems: coded and uncoded, have the same type of channel i.e. EDC. The receivers in both cases operates upon the received sequence $\{r_k\}$ and produces decision sequence $\{\hat{a}_l\}$ which is hopefully close to $\{a_l\}$ except for some delay.

Chapter 3

Coding and Modulation Scheme

Error Control Coding is a technique for improving system performance. Most of the research effort in this area has been spent for Systems with AWGN channels. The main objective is to introduce redundancy to accentuate the uniqueness of each message. The redundancy is introduced in a manner so that it is very unlikely that the channel disturbances will corrupt the received message to an extent that it loses its uniqueness. Another important objective in designing a code is its ease of decoding. Thus a coding engineer tries to optimize the trade-off between accuracy of transmission and complexity of decoding. In general Error Control Encoders take a sequence in the information sequence space and map it to a point in a larger sequence space in such a manner that the minimum mutual distances between all possible pairs of information sequences is increased. For a good code, the increase in the distance improves the performance to an extent that it offsets the loss in performance due to introduction of redundancy.

3.1 Classes of Error Correcting codes

Two major classes of error control codes exist, namely *Block codes* and *Trellis Codes.* Trellis codes deal with continuous stream stream of symbols, whereas Block Codes encode a finite number of symbols at a time. The scheme to be presented can probably be adopted for trellis codes, however, it can be explained best with the help of Block Codes. Block codes can be subdivided into a number of classes including binary codes, RS-codes [27] and so-called Ungerboek codes[1] [1] [2].

For Ungerboek Codes a dense signal constellation is selected. Using a systematic encoding process the distance between the points is effectively increased. The construction of these codes usually involves one or more simpler codes. The choice of signal constellation effects the choice of code and vice versa. However, the technique for combining the coding with equalization is quite general and would work with any binary or non-binary block codes in its current from. We use Half Leech Lattice (HLL) code for the purpose of illustration. In the remaining sections of this chapter we explain the particular type of modulation-coding scheme we used.

3.2 Signal Constellation

Due to nature of the physical channel of our interest, we were forced to use PSK. Half Leech Lattice Code, suits well to 8-PSK signal constellation. 8-PSK contains eight signal points in its constellation (see figure 3.1), and therefore it can carry three bits of information per symbol. The 8-PSK signal constellation points can be considered as eight roots of unity i.e. $\{e^{j\pi\frac{n}{8}}\}_{n=0}^{7} \triangleq \Lambda$. These roots form an *abelian group* under the usual multiplication.

The subset $S_0 \triangleq \{e^{j\pi\frac{n}{4}}\}_{n=0}^{3} \subset \Lambda$ actually forms a subgroup with respect to the usual multiplication. We use the symbol "$\sqsubset$" to show that a subset is in fact a subgroup. In this notation we can write $S_0 \sqsubset \Lambda$. We define relation $\sim$ on Λ by: $x \sim y \Leftrightarrow x \div y \in S_0$. It can readily be verified that the relation defined above is a legitimate equivalence relation. An equivalence class is the set of elements that satisfy the equivalence relation. Set S_0 itself is the trivial equivalence class. Let S_1 be the class equivalent to $e^{j\pi\frac{1}{8}}$. It can be verified that $S_1 = \{e^{j\pi\frac{2n+1}{8}}\}_{n=0}^{3}$. All the equivalence classes always partition the group. Equivalence classes w.r.t. the subgroup are called its *Cosets.* In the present case, S_1 is the coset of S_0. The set of all equivalence classes of Λ w.r.t. S_0 can be denoted by $\frac{\Lambda}{S_0}$ and it contains two elements namely S_0 and S_1. The set of equivalence classes is also a group. The identity element of this group is the subgroup w.r.t. which the Equivalence relation is defined.

[1] Ungerboek's technique works for both, Trellis and block codes.

{ needs explanation }

Now consider the subgroup $S_{00} = \{e^{j\pi \frac{n}{2}}\}_{n=0}^{1}$ of Λ. It is also a subgroup of S_0 i.e. $S_{00} \sqsubset S_0 \sqsubset \Lambda$. The equivalence relation $x \sim y \Leftrightarrow x \div y \in S_{00}$ partitions Λ into four equivalence classes each containing two points. The set of Equivalence Classes $\frac{\Lambda}{S_{00}}$ contains four elements which are, S_{00}, $S_{01} = \{e^{j\pi \frac{2n+2}{8}}\}_{n=0}^{1}$, $S_{10} = \{e^{j\pi \frac{4n+1}{8}}\}_{n=0}^{1}$ and $S_{11} = \{e^{j\pi \frac{4n+3}{8}}\}_{n=0}^{1}$. S_{00} and S_{01} partition S_0, whereas, S_{10} and S_{11} partition S_1.

Similarly, each point in Λ can be thought of as equivalent class of the singleton set $S_{000} \triangleq \{e^{j\pi \frac{0}{8}} = 1\}$, which is the trivial subgroup of all previous subgroups i.e. $S_{000} \sqsubset S_{00} \sqsubset S_0 \sqsubset \Lambda$. Thus, $\frac{\Lambda}{S_{000}}$ is a the set of eight equivalence classes. The last partitioning, partitions S_{00}, S_{01}, S_{10} and S_{11} into two equivalence classes each. The series of partitioning described above is shown in figure 3.2. If every set in a class of sets A is partitioned by sets in a class of sets B then we write $A \prec B$ and we say that B is a *finer* partition of A. In this notation, we can write $\frac{\Lambda}{S_0} \prec \frac{\Lambda}{S_{00}} \prec \frac{\Lambda}{S_{000}}$. Note e.g. that $\frac{\Lambda}{S_{000}}$ is a finer partition of $\frac{\Lambda}{S_{00}}$.

3.3 Half Leech Lattice Code

Each HLL code word consists of 24 symbols, each capable of carrying 3 bits of information. Therefore, each symbol can be mapped to a point in Λ. The 24 symbols altogether can be mapped to a point in a 24 Dimensional Cartesian product complex space. The total capacity of uncoded 24-symbols is 24×3 bits which can carry $2^{24\times 3}$ different messages. Each message can be mapped to $2^{24\times 3}$ distinct signal points in the 24-D space that satisfy the 8-PSK constraint in every dimension. These points actually form a Cartesian product set $\Pi_{i=1}^{24}\Lambda = \Lambda^{24}$. The projection of each point in Λ^{24} over any (complex) dimension is one of the 8 points in Λ.

3.3.1 Algebraic Structure over Λ^{24}

Every point x in Λ^{24} is a 24-Dimensional point of the form $(e^{j\pi\frac{n_1}{8}}, e^{j\pi\frac{n_2}{8}}, \ldots, e^{j\frac{\pi n_{24}}{8}})$. We define the *product multiplication* over Λ^{24} as follows:

$$\begin{array}{lrcl} \text{let} & x & = & (e^{j\pi\frac{n_1}{8}}, e^{j\pi\frac{n_2}{8}}, \ldots, e^{j\frac{\pi n_{24}}{8}}) \\ \text{and} & y & = & (e^{j\pi\frac{k_1}{8}}, e^{j\pi\frac{k_2}{8}}, \ldots, e^{j\frac{\pi k_{24}}{8}}) \\ \text{then} & xy & \triangleq & (e^{j\pi\frac{n_1+k_1}{8}}, e^{j\pi\frac{n_2+k_2}{8}}, \ldots, e^{j\pi\frac{n_{24}+k_{24}}{8}}). \end{array} \tag{3.1}$$

If the multiplication identity is the point $(1, 1, \ldots, 1)$ and the multiplication inverses are defined in the obvious sense, the set Λ^{24} becomes an abelian group.

With reference to the notation introduced in section 3.2, consider the product set $P_0 \triangleq \Pi_{i=1}^{24} S_0 \subset \Lambda^{24}$. It is easy to show that $P_0 \sqsubset \Lambda^{24}$ w.r.t. the product multiplication defined above. We define relation $\sim$ on Λ^{24} w.r.t. P_0 as $x \sim y \Leftrightarrow x \div y \in P_0$ where $x, y \in \Lambda^{24}$. It is easy to show that the above relation is a legitimate equivalence relation and that it partitions the group Λ^{24} into 2^{24} equivalence classes. The set of these equivalence classes is denoted by $\frac{\Lambda^{24}}{P_0}$. Each element in $\frac{\Lambda^{24}}{P_0}$ is of the form $\Pi_{l=1}^{24} S_{i_l} = [S_{i_1} \times S_{i_2} \times \ldots \times S_{i_{24}}]$, where $i_l \in \{0, 1\}$; $1 \leq l \leq 24$.

The sets $P_{00} \triangleq \Pi_{i=1}^{24} S_{00} \subset \Lambda^{24}$ and $P_{000} \triangleq \Pi_{i=1}^{24} S_{000} \subset \Lambda^{24}$ are also subgroups, and in fact $P_{000} \sqsubset P_{00} \sqsubset P_0 \sqsubset \Lambda^{24}$. These subgroups partition Λ^{24} into equivalence classes in a manner similar to that described in the previous paragraph. Sets of equivalence classes thus obtained are $\frac{\Lambda^{24}}{P_{00}}$ and $\frac{\Lambda^{24}}{P_{000}}$, containing $2^{24\times 2}$ and $2^{24\times 3}$ equivalent classes, respectively. The elements of $\frac{\Lambda^{24}}{P_{00}}$ are of the form $\Pi_{l=1}^{24} S_{i_l j_l} \triangleq [S_{i_1 j_1} \times S_{i_2 j_2} \times \ldots \times S_{i_{24} j_{24}}]$, and those of $\frac{\Lambda^{24}}{P_{000}}$ are of the form $\Pi_{l=1}^{24} S_{i_l j_l k_l} \triangleq [S_{i_1 j_1 k_1} \times S_{i_2 j_2 k_2} \times \ldots \times S_{i_{24} j_{24} k_{24}}]$, where $i_l, j_l, k_l \in \{0, 1\}$; $1 \leq l \leq 24$. We now have a series of partitions of Λ^{24} viz. $\Lambda^{24} \prec \frac{\Lambda^{24}}{P_0} \prec \frac{\Lambda^{24}}{P_{00}} \prec \frac{\Lambda^{24}}{P_{000}}$. Note that, equivalence classes of $\frac{\Lambda^{24}}{P_{000}}$ partition each class of $\frac{\Lambda^{24}}{P_{00}}$ into 2^{24} parts. Similarly, each element of $\frac{\Lambda^{24}}{P_0}$ is partitioned into 2^{24} equivalence classes by $\frac{\Lambda^{24}}{P_{00}}$.

3.3.2 The Code and Encoding Process

We will denote the subset of Λ^{24} that consists of HLL the codeword points by C. HLL uses Extended Golay and Even Parity codes (both are binary codes) in the construction. We define *set distance* between two subsets A and B of N dimensional complex Space as

$$d(A,B) = \min\{d(a,b) : \forall a \in A, \forall b \in B\}, \tag{3.2}$$

where $d(a,b)$ is the usual **squared** Euclidean distance between a and b. By definition $d(A,A) = 0$. We also define the *Minimum distance* of a set A by

$$d_{\min}(A) \quad = \quad \min\{d(a,b) : \forall a,b \in A, a \neq b\} \tag{3.3}$$

e.g. $d_{\min}(\Lambda) = [2 \times \sin(22.5°)]^2 \simeq 0.586$. Note that $d(S_0, S_1) = d_{\min}(\Lambda) \simeq 0.586$. Thus points closest together in Λ belong to different equivalence classes. Also note that $d(S_{i0}, S_{i1}) = 2$ and $d(S_{ij0}, S_{ij1}) = 4$ for $i,j \in \{0,1\}$.

Now consider $Q_1 = \Pi_{l=1}^{24} S_{i_l}$ and $Q_2 = \Pi_{l=1}^{24} S_{p_l}$. Then $Q_1, Q_2 \in \frac{\Lambda^{24}}{P_0}$ and $Q_1 \neq Q_2$ if $i_l \neq p_l$ for at least one l. Then $d(Q_1, Q_2) = \sum_{l=1}^{24} d(S_{i_l}, S_{p_l})$. By taking all $i_l = p_l$ except one we see that

$$\begin{aligned} d_{\min}\left(\frac{\Lambda^{24}}{P_0}\right) &\doteq \min\{d(Q_1,Q_2) : \forall Q_1, Q_2 \in \frac{\Lambda^{24}}{P_0}, Q_1 \neq Q_2\} \\ &= d(S_0, S_1). \end{aligned}$$

The encoding process is carried out in three steps. The first step uses the Extended binary Golay Code $(24,12,8)$, the second step the Even Parity code $(24,23,2)$ and the third step uses the trivial code $(24,24,1)$.

Step 1 Twelve information bits are taken and encoded to obtain an extended Golay codeword, say $i = (i_1, i_2, \ldots, i_{24})$ where $i_l \in \{0,1\}$; $1 \leq l \leq 24$. Let the set of all Golay codewords be G. We define a map $f_1 : G \rightarrow \frac{\Lambda^{24}}{P_0}$, as follows

$$f_1(i_1, i_2, \ldots, i_{24}) \quad = \quad \Pi_{l=1}^{24} S_{i_l} \tag{3.4}$$

i.e. the lth bit of the codeword selects the equivalence class S_{i_l} in the lth dimension of Λ^{24}. Thus the Golay code word selects an equivalence class in $\frac{\Lambda^{24}}{P_0}$. Note that not all equivalent classes in Λ^{24} can be selected by Golay codewords. We denote the image of set A over any function f by $f(A)$. In this notation

$$f_1(G) = \{\Pi_{i=1}^{24} S_{i_l} : \forall (i_1, i_2, \ldots, i_{24}) \in G\}.$$

It can be shown that $f_1(G) \sqsubset \frac{\Lambda^{24}}{P_0}$. { needs explanation } Note that the function f_1 preserves the group structure of G. We now proceed to find the minimum distance of $f_1(G)$.

Lemma 1 *The minimum distance of $f_1(G)$ is $8 \times d(S_0, S_1)$.*

Proof: Let $Q_1, Q_2 \in f_1(G)$ then $Q_1 = \Pi_{i=1}^{24} S_{i_l}$ and $Q_2 = \Pi_{i=1}^{24} S_{p_l}$. Then $i = (i_1, i_2, \ldots, i_{24})$ and $p = (p_1, p_2, \ldots, p_{24})$ are Golay codewords ($i, p \in G$), and they differ at least in eight locations. This means that $S_{i_l} \neq S_{p_l}$ for at least eight values of l. Thus,

$$d(Q_1, Q_2) = \sum_{i_l \neq p_l} d(S_{i_l}, S_{p_l}) \geq \sum_{i=1}^{8} d(S_0, S_1) = 8 \times d(S_0, S_1) \simeq 4.686$$

Since, there is a codeword in G with exactly 8 non-zero bits we have

$$\begin{aligned} d_{\min}(f_1(G)) &= \min\{d(Q_1, Q_2) : Q_1, Q_2 \in \frac{\Lambda^{24}}{P_0}, Q_1 \neq Q_2\} \\ &= 8 \times d(S_0, S_1) \simeq 4.686. \clubsuit \end{aligned}$$

Hence, the selected sets of points are at least at a distance of 4.686 from each other.

Step2 Twenty three additional information bits are encoded to obtain a twenty four bit Even Parity code say $j = (j_1, j_2, \ldots, j_{24})$ where $j_l \in \{0,1\}; 1 \leq l \leq 24$. Let the set of all even parity codewords be E. It follows from our discussion in section 3.2 that S_{i_l} is partitioned by $S_{i_l 0}$ and $S_{i_l 1}$. We define a map $f_2 : E \rightarrow \frac{\Lambda^{24}}{P_{00}}$ by

$$f_2(j_1, j_2, \ldots, j_{24}) = \Pi_{l=1}^{24} S_{i_l j_l};\ \ i_l\text{'s as selected in} f_1$$

i.e. the lth bit of the codeword selects the j_lth equivalence class in S_{i_l}. The range of f_2 is $f_2(E)$. We observe that $f_2(E)$ depends upon the choice of the equivalence class made in step 1. We define $F_1 = f_1$ and $F_2 = f_2 \circ F_1 = f_2 \circ f_1$. Then $F_2 : G \times E \Rightarrow \frac{\Lambda^{24}}{P_{00}}$ and

$$F_2(i, j) = \Pi_{l=1}^{24} S_{i_l j_l};$$

The range of F_2 i.e. $F_2(G \times E)$, is fixed. { $F_2(G \times E)$ is a subgroup ? } We Now find the minimum distance of $f_2(E)$.

Lemma 2 *The minimum distance of all possible sets $f_2(E)$ is* $2 \times d(S_{00}, S_{01})$.

Proof: Let $Q_1, Q_2 \in f_2(E)$, then $Q_1 = \Pi_{l=1}^{24} S_{i_l j_l}$, $Q_2 = \Pi_{l=1}^{24} S_{i_l p_l}$; $(i_1, i_2, \ldots, i_{24}) \in G$ and $(j_1, j_2, \ldots, j_{24}), (p_1, p_2, \ldots, p_{24}) \in E$. Then j and i differ at least in two locations when $j \neq p$. This means that $S_{i_l j_l} \neq S_{i_l p_l}$ for at least two values of l. Therefore,

$$\begin{aligned} d(Q_1, Q_2) &= \sum_{j_l \neq p_l} d(S_{i_l j_l}, S_{i_l p_l}) \\ &\geq 2 \times d(S_{i_l 0}, S_{i_l 1}) \\ &= 2 \times d(S_{00}, S_{01}) = 2 \times 2 = 4. \end{aligned}$$

Since, there is a codeword in E with exactly two non-zero bits

$$d_{\min}(f_2(G)) = 2 \times d(S_{00}, S_{01}) \clubsuit$$

Step 3 Twenty four additional information bits are taken, say $k = (k_1, k_2, \ldots, k_{24})$. Let T be the set of all possible 24−tuples. We define $f_3; T \rightarrow \frac{\Lambda^{24}}{P_{000}}$ as

$$f_3(k_1, k_2, \ldots, k_{24}) = \Pi_{l=1}^{24} S_{i_l j_l k_l},$$

where i_l's and j_l's are as selected in steps 1 and 2, respectively. $f_3(T)$ depends upon the choice of equivalence class made in steps 1 and 2. Define $F_3 = f_3 \circ F_2 = f_3 \circ f_2 \circ f_1$, then $F_3 : T \times G \times E \rightarrow \frac{\Lambda^{24}}{P_{000}}$ and

$$F_3(i, j, k) = \Pi_{l=1}^{24} S_{i_l j_l k_l}.$$

The range of F_3 i.e. $F_3(T \times G \times E)$ is fixed. The points in $\frac{\Lambda^{24}}{P_{000}}$ are singleton sets, so that map F_3 can also be considered as $F_3 : T \times G \times E \rightarrow \Lambda^{24}$ and $F_3(T \times G \times E) = C$. Note that $d(S_{i_l j_l 0}, S_{i_l j_l 1}) = 4$, therefore, $d(Q_1, Q_2) = 4; \forall Q_1, Q_2 \in f_3(T); \forall$ possible $f_3(T)$.

3.3.3 Minimum Distance of the Code

We now show that the minimum distance of the HLL code is 4.

Theorem 3 *The minimum distance of C is 4.*

Proof: Let $c_1, c_2 \in C$. Then there are three cases:

Case 1: c_1, c_2 belong to different equivalence class of $\frac{\Lambda^{24}}{P_0}$. Then

by Lemma 1 $d(c_1, c_2) \geq 8 \times d(S_0, S_1) = 4.686$.
Case 2: c_1, c_2 belong to different equivalence class of $\frac{\Lambda^{24}}{P_{00}}$ but same equivalence class of $\frac{\Lambda^{24}}{P_0}$. Then from Lemma 2 $d(c_1, c_2) \geq 2 \times d(S_{00}, S_{01}) = 4$.
Case 3: c_1, c_2 belong to same equivalence class of $\frac{\Lambda^{24}}{P_{00}}$. then from our discussion in step 3 it follows $d(c_1, c_2) = 4$. Hence, the minimum distance of the Code is 4. ♣

3.3.4 Ecoding in practice

We now discuss the encoding process for the HLL codeword. The manner in which encoding is actually performed is depicted in figure 3.3. We consider the ith HLL codeword. First, 12 bits of information bits are encoded to form the ith 24 bit Golay codeword $g_i = [g_{ij}]_{j=1}^{24}$. g_{ij} determines the most significant bit in the binary representation of the jth symbol of the ith HLL codeword i.e. c_{ij}. Note that $c_{ij} \in \{0, 1, \ldots, 7\}$. Second, 23 additional information bits are encoded to form the ith even parity codeword $e_i = [e_{ij}]_{j=1}^{24}$. The second most significant bit of c_{ij} is determined by e_{ij}. Finally, another 24 information bits determine the least significant bit of c_{ij}. Hence,

$$c_{ij} = 2^2 g_{ij} + 2e_{ij} + u_{ij}$$

where u_{ij} denotes the uncoded information bits used in the final step.

Chapter 4

Decision Feedback Equalization

Our Codesigned receiver is primarily based upon a number of Decision Feedback Equalizers. In this chapter we explain how a DFE works. In the Decision Feedback equalizer, a channel symbol is demodulated via a two fold process;

1. by applying filtering to a register containing channel samples taken at times earlier and including the time of the channel symbol being demodulated.

2. by applying filtering to a register containing the decision which are the previously demodulated symbols.

The first filter is known as the Feed Forward Filter (FFF) and the latter as the FeedBack Filter (FBF). Figure 4.1 shows a schematic for such an equalizer. The function applied to the registers are intended to remove the effect of channel distortion and to produce an accurate prediction of the transmitted modulation symbol.

4.1 Classification of Equalizers

There are two major types of equalizers fitting this description.

1. A Linear Equalizer; consists of FFF and demodulator – FBF is not present.

2. A Decision Feedback Equalizer; both FFF and FBF are present. Presence of a FBF constitutes a non-linear process. The act of replacing a prediction point (which can lie anywhere in the complex plane) by a decision which is a signal constellation point, is a non-linear process.

Each of the above can be further classified into two subclasses.

1. Linear Function: the functions applied to the registers are both linear. If h is the function applied by a filter and the inputs to the filter are $\underline{x} = [x_1, x_2, \ldots, x_r]^T$, the output of the filter can be expressed as $\sum_{i=1}^{r} x_i h_i$, where h_i are the filter coefficients.

2. Non-Linear Functions: at least one of the functions applied to one of the registers contains non-linear term, so $h(\underline{x})$ cannot be expressed as a linear combinations of elements of $\underline{x}$.

Our emphasis will mainly be on Decision Linear Feedback Equalizer. However, the technique to be described is equally applicable to Decision Non-linear FeedBack Equalizers.

4.2 Operation of the DFE

For convenience we repeat equation (2.2).

$$r_k = g(d(k)) + n_k.$$

Let the function applied by FFF be h^f. Define

$$r'_k \triangleq h^f(r_k) = (h^f \circ g)(d(k)) + h^f(n_k),$$

and define

$$f \triangleq h^f \circ g \quad \text{and} \quad v_k \triangleq h^f(n_k).$$

Then,

$$r'_k = f(d(k)) + v_k.$$

Let $\{\hat{d}_k\}$ be the sequence of decisions made by the DFE which is hopefully close to $\{d_k\}$, the transmitted sequence. Let $\hat{\underline{d}}_k \triangleq [\hat{d}_{k-1}, \hat{d}_{k-2}, \ldots, \hat{d}_{k-L_2}]^T$. Let the function applied by the FBF be h^b. Then the output of the FBF is $h^b(\hat{d}(k))$.

The *prediction* to the transmitted symbol, denoted by $\tilde{d}_k$, is defined as

$$\tilde{d}_k \triangleq r'_k - h^b(\hat{d}(k)) = h^f(r_k) - h^b(\hat{d}(k)).$$

The prediction is demodulated into the nearest (in the squared Euclidean distance sense) signal constellation point. The demodulated symbol is actually the decision of DFE $\hat{d}_k$. Thus,

$$\hat{d}_k = \min_n \{d(\tilde{d}_k, p_n)\},$$

where p_n is the nth constellation point. For 8-PSK, $p_n = e^{j\pi\frac{n}{8}}$. For Decision linear FeedBack Equalizer, we have

$$\tilde{d}_k = \sum_{i=-K_1}^{0} h_i r_{k-i} - \sum_{j=1}^{K_2} h_j \hat{d}_{k-j},$$

where $\{h_i\}_{i=-K_1}^{0}$ and $\{h_j\}_{j=1}^{K_2}$ are the feed forward and feedback coefficients, respectively. From here onwards by DFE we will mean Decision Linear Feedback Equalizer unless mentioned otherwise.

4.3 Optimizing the Coefficients of DFE

One can optimize the coefficients of the DFE using Mean Square Error (MSE) criterion [15] or the Zero Forcing criterion [29]. Because of its superiority, we discuss the optimization using MSE criterion. We introduce some notation first,

$$\underline{h}^f \triangleq [h_{-K_1}, \ldots, h_{-1}, h_0]^T \tag{4.1}$$

$$\underline{h}^b \triangleq [h_1, h_2, \ldots, h_{K_2}]^T \tag{4.2}$$

$$r(k) \triangleq [r_{k+K_1}, \ldots, r_{k+1}, r_k]^T \tag{4.3}$$

For convenience, we repeat,

$$\hat{d}(k) \triangleq [\hat{d}_{k-1}, \hat{d}_{k-2}, \ldots, \hat{d}_{k-K_2}]^T \tag{4.4}$$

We also define

$$d(k) = [d_{k-1}, d_{k-2}, \ldots, d_{k-K_2}]^T. \tag{4.5}$$

Then,

$$\tilde{d}_k = (\underline{h}^f)^T r(k) - (\underline{h}^b)^T \hat{d}(k).$$

Define

$$\underline{h} \triangleq \begin{bmatrix} \underline{h}^f \\ -\underline{h}^b \end{bmatrix} \text{ and } \underline{r} \triangleq \begin{bmatrix} r(k) \\ \hat{d}(k) \end{bmatrix}. \tag{4.6}$$

Then,

$$\tilde{d}_k = \underline{h}^T \underline{r}.$$

4.4 Minimizing the MSE

For the analysis we assume that $d(k) = \hat{d}(k)$ which implies that the previous decisions of the DFE are correct. The mean square error is

$$\varepsilon(\underline{h}) = E\{|\tilde{d}_k - d_k|^2\}, \tag{4.7}$$

which is minimized when the error $e_k \triangleq \tilde{d}_k - d_k$ is orthogonal to $\tilde{d}_k$ (Orthogonality Principle), i.e.

$$\begin{aligned} E\{\tilde{d}_k^*(\tilde{d}_k - d_k)^T\} &= 0 \\ E\{\tilde{d}_k^*\tilde{d}_k\} &= E\{\tilde{d}^*(k)d_k\}. \end{aligned}$$

By substitution, value of $\tilde{d}_k$ we get

$$E\{\underline{h}_{opt}^H \underline{r}^* \underline{r}^T \underline{h}_{opt}\} = E\{\underline{h}_{opt}^H \underline{r}^* d_k\},$$

where superscript "^{H}" indicates Hermitian transpose and the subscript "$_{\text{opt}}$" indicates the particular vector $\underline{h}$ that gives minimum MSE. Thus,

$$E\{\underline{r}^* \underline{r}^T\}\underline{h}_{opt} = E\{\underline{r}^* d_k\}.$$

Let $X \triangleq E\{\underline{r}^*\underline{r}^T\}$ and $W \triangleq E\{\underline{r}^* d_k\}$. Then the previous equation can be rewritten as

$$X\underline{h}_{opt} = W. \tag{4.8}$$

4.4.1 Finding the Coefficients

Let us see what it means in terms of the individual coefficients. Consider X, substituting values from equation (4.6) (with $\hat{d}(k) = d(k)$),

$$\begin{aligned} X &= E\{\underline{r}^*\underline{r}^T\} \\ &= E\left\{\begin{bmatrix} r^*(k) \\ d^*(k) \end{bmatrix} [r^T(k) \;\; d^T(k)]\right\} \\ &= \begin{bmatrix} E\{r^*(k)r^T(k)\} & E\{r^*(k)d^T(k)\} \\ E\{d^*(k)r^T(k)\} & E\{d^*(k)d^T(k)\} \end{bmatrix} \end{aligned}$$

Let

$$R \triangleq E\{r^*(k)r^T(k)\} \text{ and } G \triangleq E\{r^*(k)d^T(k)\}.$$

Note that $E\{d^*(k)d^T(k)\} = I$, where we assume that signal zero mean and its variance is 1. Then,

$$X = \begin{bmatrix} R & G \\ G^H & I \end{bmatrix}.$$

Similarly,

$$W = \begin{bmatrix} E\{r^*(k)d_k\} \\ E\{d^*(k)d_k\} \end{bmatrix} = \begin{bmatrix} W_1 \\ 0 \end{bmatrix},$$

where $W_1 = E\{r^*(k)d_k\}$. Equation (4.8) can now be written as

$$\begin{bmatrix} R & G \\ G^H & I \end{bmatrix} \begin{bmatrix} \underline{h}^f_{opt} \\ -\underline{h}^b_{opt} \end{bmatrix} = \begin{bmatrix} W_1 \\ 0 \end{bmatrix} \tag{4.9}$$

$$R\underline{h}^f_{opt} - G\underline{h}^b_{opt} = W_1 \tag{4.10}$$

$$G^H\underline{h}^f_{opt} - \underline{h}^b_{opt} = 0 \tag{4.11}$$

From equation (4.11) we have

$$\underline{h}^b_{opt} = G^H\underline{h}^f_{opt}. \tag{4.12}$$

Substituting this value in equation (4.10),

$$\begin{aligned} R\underline{h}^f_{opt} - GG^H\underline{h}^f_{opt} &= W_1 \\ (R - GG^H)\underline{h}^f_{opt} &= W_1 \end{aligned} \tag{4.13}$$

Feedforward Coefficients

Now we find the elements of the vector W_1 and matrix $(R - GG^H)$ in order to find elements of $\underline{h}^f_{opt}$.

$$W_1 = E\{r^*(k)d_k\} = \begin{bmatrix} E\{r^*_{k+K_1}d_k\} \\ \vdots \\ E\{r^*_{k+1}d_k\} \\ E\{r^*_k d_k\} \end{bmatrix} = \begin{bmatrix} g^*_{K_1} \\ \vdots \\ g^*_1 \\ g^*_0 \end{bmatrix} \tag{4.14}$$

$$\begin{aligned} G &= E\left\{ \begin{bmatrix} r^*_{k+K_1} \\ \vdots \\ r^*_{k+1} \\ r^*_k \end{bmatrix} [d_{k-1}\ d_{k-2} \ldots d_{k-K2}] \right\} \\ &= \begin{bmatrix} g^*_{K_1+1} & g^*_{K_1+2} & \cdots & g^*_{K_1+K_2} \\ \vdots & \vdots & & \vdots \\ g^*_2 & g^*_3 & \cdots & g^*_{1+K_2} \\ g^*_1 & g^*_2 & \cdots & g^*_{+K_2} \end{bmatrix} \end{aligned} \tag{4.15}$$

$$GG^H = \begin{bmatrix} \sum_{i=1}^{K_2} g^*_{K_1+i}g_{K_1+i} & \cdots & \sum_{i=1}^{K_2} g^*_{K_1+i}g_{1+i} & \sum_{i=1}^{K_2} g^*_{K_1+i}g_i \\ \vdots & \vdots & & \vdots \\ \sum_{i=1}^{K_2} g^*_{1+i}g_{K_1+i} & \cdots & \sum_{i=1}^{K_2} g^*_{1+i}g_{1+i} & \sum_{i=1}^{K_2} g^*_{1+i}g_i \\ \sum_{i=1}^{K_2} g^*_i g_{K_1+i} & \cdots & \sum_{i=1}^{K_2} g^*_i g_{1+i} & \sum_{i=1}^{K_2} g^*_i g_i \end{bmatrix} \tag{4.16}$$

$$R = \begin{bmatrix} E\{r^*(k+K_1)r^T(k+K_1)\} & \ldots & E\{r^*(k+K_1)r^T(k)\} \\ \vdots & & \vdots \\ E\{r^*(k+1)r^T(k+K_1)\} & \ldots & E\{r^*(k+1)r^T(k)\} \\ E\{r^*(k)r^T(k+K_1)\} & \ldots & E\{r^*(k)r^T(k)\} \end{bmatrix} \tag{4.17}$$

$$= [s_{lm}]^{0}_{l=-K_1}{}^{0}_{m=-K_1}, \tag{4.18}$$

where

$$\begin{aligned} s_{lm} &= E\{r^*(k-l)r^T(k-m)\} \\ &= \sum_{i=-K_1}^{K_2} \sum_{j=-K_1}^{K_2} g^*_i g_j E\{d^*_{k-l-i}d_{k-m-j}\} + E^2\{n_{k-l}n_{k-m}\} \\ &= \sum_{i=-K_1}^{K_2} g^*_i g_{i-m+l} + N_0\delta_{lm} \end{aligned}$$

Using equations (4.17) and (4.16) we get $(R - GG^H)$

$$(R-GG^H) = \begin{bmatrix} \sum_{i=-K_1}^{-K_1} g_i^* g_i + N_0 & \cdots & \sum_{i=-K_1}^{-K_1} g_i^* g_{i+1-K_1} & \sum_{i=-K_1}^{-K_1} g_i^* g_{i-K_1} \\ \vdots & & \vdots & \vdots \\ \sum_{i=-K_1}^{1} g_i^* g_{i+K_1-1} & \cdots & \sum_{i=-K_1}^{1} g_i^* g_i + N_0 & \sum_{i=-K_1}^{1} g_i^* g_{i-1} \\ \sum_{i=-K_1}^{0} g_i^* g_{i+K_1} & \cdots & \sum_{i=-K_1}^{0} g_i^* g_{i+1} & \sum_{i=-K_1}^{0} g_i^* g_i + N_0 \end{bmatrix}$$

Putting value of $(R - GG^H)$ and W_1 in equation(4.13), we get coefficients of the feed forward filter.

$$\begin{bmatrix} \sum_{i=-K_1}^{-K_1} g_i^* g_i + N_0 & \cdots & \sum_{i=-K_1}^{-K_1} g_i^* g_{i+1-K_1} & \sum_{i=-K_1}^{-K_1} g_i^* g_{i-K_1} \\ \vdots & & \vdots & \vdots \\ \sum_{i=-K_1}^{1} g_i^* g_{i+K_1-1} & \cdots & \sum_{i=-K_1}^{1} g_i^* g_i + N_0 & \sum_{i=-K_1}^{1} g_i^* g_{i-1} \\ \sum_{i=-K_1}^{0} g_i^* g_{i+K_1} & \cdots & \sum_{i=-K_1}^{0} g_i^* g_{i+1} & \sum_{i=-K_1}^{0} g_i^* g_i + N_0 \end{bmatrix} \begin{bmatrix} h_{-K_1} \\ \vdots \\ h_{-1} \\ h_0 \end{bmatrix} = \begin{bmatrix} g_{-K_1}^* \\ \vdots \\ g_{-1}^* \\ g_0^* \end{bmatrix}.$$

Let $[x_{ij}]_{i=0,j=0}^{K_1,K_1} = (R - GG^H)$, then

$$\sum_{j=0}^{K_1} x_{ij} h_{-j} = g_i^* \quad i = 0, 1, 2, \ldots, K_1,$$

where $x_{ij} = \sum_{l=-K_1}^{K_1-i} g_l^* g_{l+j-i} + N_0 \delta_{ij}$.

Feedback Coefficients

Putting value G in equation (4.12), we observe that the feedback coefficients can be expressed in terms of the coefficients of the feed forward section.

$$\begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{K_2} \end{bmatrix} = \begin{bmatrix} g_{K_1+1} & \cdots & g_2 & g_1 \\ g_{K_1+2} & \cdots & g_3 & g_2 \\ \vdots & & \vdots & \vdots \\ g_{K_1+K_2} & \cdots & g_{K_2+1} & g_{K_2} \end{bmatrix} \begin{bmatrix} h_{-K_1} \\ \vdots \\ h_{-1} \\ h_0 \end{bmatrix}$$

$$h_i = \sum_{j=-K_1}^{0} g_{i-j} h_{-j} \quad i = 1, 2, \ldots, K_2$$

4.5 Adaptive Decision Feedback Equalizer

When the channel is unknown, the matrix X and the vector $\underline{W}$ are unknown. Any of the well known adaptive techniques can be used to estimate the channel coefficients. The simplest adaptive algorithm is the stochastic gradient algorithm.

The gradient of $\varepsilon(\underline{h})$ w.r.t. $\underline{h}$ is given by,

$$\nabla_{\underline{h}}\varepsilon = 2E\{(\tilde{d}_k - d_k)\underline{x}_k^*\}.$$

Separating the feed forward and feedback parts and estimating the gradient, we get

$$\begin{aligned}\underline{h}^f(k+1) &= \underline{h}^f(k) - \alpha\underline{e}_k r^*(k)\\ \underline{h}^b(k+1) &= \underline{h}^b(k) - \beta\underline{e}_k\hat{\underline{d}}^*(k),\end{aligned}$$

where $e_k = \tilde{d}_k - d_k$, and α, β are step sizes which determine convergence rate.

4.6 A Lower Bound on Signal to Noise Ratio

A lower bound on the Signal to noise ratio at the input of the Demodulator achieved by the MSE-DFE is obtained by assuming absence of decision errors in the FBF. Under this assumption, by putting values of $\underline{h}_{opt}$ from equation (4.8) into equation (4.7), we obtain

$$\varepsilon_{min}(K_1) = 1 - \sum_{j=-K_1}^{0} g_{-j}h_{-j}.$$

By taking the limit $K_1 \longrightarrow \infty$ (infinitely many feed forward coefficients), we obtain the smallest possible MSE denoted as ε_{min}. The signal to noise ratio ($\frac{E_s}{N_0} = \frac{\text{energy per symbol}}{\text{noise variance}}$) is [15]

$$\frac{E_s}{N_0} \leq \frac{1-\varepsilon_{min}}{\varepsilon_{min}},$$

where $E_s = 1$. In absence of ISI, the $\varepsilon_{min} = \frac{N_0}{1+N_0}$ [15] and $\frac{E_s}{N_0} = \frac{1}{N_0}$.

4.7 Error Propagation

Consider equations (2.2) and (2.3). DFE attempts to cancel the ISI. The best any ISI canceller can do is to remove both precursor and post cursor ISI. Such a canceller can be called *ideal canceller.* The prediction from the ideal canceller would be

$$\tilde{d}_k = g_0 d_k + n_k,$$

i.e the overall impulse response (say $\{f_k\}$) of the system prior to demodulation is $f_k = g_0\delta(k)$. This also implies the overall frequency response is flat. The manner in which this hypothetical receiver makes errors is same as that by the receiver in a system which is designed according to the Nyquist criterion. Hence, the ideal canceller makes *random* errors; every symbol error is uncorrellated to other symbol errors. A sample of output symbol errors of the ideal canceller is shown in figure 4.3(a)

The FBF filter of the DFE in fact attempts to from a replica of the post cursor ISI. In doing this, it succeeds only if the previous decisions ($\hat{d}(k)$) are correct. Whenever there is an error in $\hat{d}(k)$, the FBF is not able to form a perfect replica of the post cursor ISI. Therefore, subsequent decision errors become more likely whenever $\hat{d}(k) \neq d(k)$. This results into *error propagation*; errors tend to occur in clusters. For DFE we have cluster of one or more errors for every error that could have occurred in the ideal canceller. Error propagation is depicted in figure 4.3(b).

Chapter 5

The Codesigned Receiver

Now we describe the Codesigned receiver. We consider the data transmission system shown in figure 1.1. The transmitter contains a block encoder, helical interleaver and the modulator as discussed in chapter 2. We want to use Decision Feedback Equalizer (DFE) in the receiver. However, we know from our discussion in chapter 4 that DFE suffers from *error propagation*. An attempt to use a simple juxtaposition of DFE, Deinterleaver and Decoder as shown in figure 5.1, gives poor performance (sometimes worse than that of the DFE alone) because the number of errors are far too many for the decoder to handle. One might consider solving this problem by adding error correction between the symbol detection and the feedback register as shown in figure 5.2. This transpires, at least in the form suggested, to be impractical and/or disappointing in practice. The reason is simple. The correction either involves excessive delay before decoded data is available to the feedback filter or the code selection is limited to codes of short length, consequently, limiting the performance. Eyuboglu [3] introduced the use of interleaving which scrambles the input data stream in such a manner that every received symbol is preceded by either a symbol of a codeword transmitted earlier or by a sync symbol. His technique, however, suffers from the disadvantage that only the decoded data symbols are available to the Feed Back Filter (FBF). This, not only restricts the size of the FBF, but also makes its length variable, periodically (it varies with the index j for $k = Ni + (N-1)j$). We present a technique that makes it possible to provide both the decoded and undecoded data to FBF. Thus for the symbol locations where decoded version of the symbol is not available, the undecoded symbols can be used.

Therefore, a full length Feed Back Filter can be used. This, in turn gives superior equalization and hence better performance. Before we discuss the structure of the receiver of the receiver, it is instructive to consider some properties possessed by the sequence $\{c_k\}$.

5.1 Properties of the Interleaved Sequence

Consider the channel symbol r_k; the corrupted version of d_k which in turn is the modulated version of c_k. We can say that r_k is the output of the channel corresponds to the encoded symbol c_k. We denote this correspondence by $r_k \simeq d_k \simeq c_k$; the symbol "$\simeq$" should be read as "corresponding to". Note that $k = Ni + (N-1)j;\ 0 \le j \le (N-1)$. The c_k is in fact the encoded symbol c_{ij}, i.e. it belongs to the i^{th} codeword. We consider the received symbol immediately preceding r_k , i.e. $r_{k-1} \simeq c_{k-1}$. We now show that c_{k-1} either belongs to $(i-1)^{th}$ codeword or is a sync symbol.

Theorem 4 *If* $c_k = c_{ij};\ k = Ni+(N-1)j;\ j = 0,1,\ldots,(N-1)$ *and* $k > (N-1)(N-2)+1$, *then* $c_{k-1} = c_{(i-1)j'};\ j' = 1,\ldots,(N-1)$ *or* $c_{k-1} = c_{i'o}$; *some* $i' \ge 0$.

Proof:

$$\begin{aligned} k-1 &= Ni + (N-1)j - 1 \\ &= Ni + (N-1)j - N(N-1) \end{aligned} \tag{5.1}$$

Case 1: If k is such that $j = 0,1,\ldots,(N-2)$, then

$$k-1 = N(i-1) + (N-1)(j+1).$$

Thus, in this case, c_{k-1} belongs to the $(i-1)^{th}$ codeword.
Case 2: If $j = N-1$, from equation (5.1), we have

$$\begin{aligned} k-1 &= Ni + (N-1)(N-1) - N + (N-1) \\ &= N(i-1) + (N-1)(N) = N(i+N-2). \end{aligned}$$

Then c_{k-1} is a sync symbol. ♣

From the above theorem, it follows that when the reception of the channel symbols corresponding to the symbols of the ith is completed, it is guaranteed that the symbols corresponding to the $(i-1)$th codeword would also be received earlier. Therefore, it is possible to process the channel symbols corresponding to the $(i-1)$th codeword before we process those corresponding to the ith codeword. Hence, the processed symbols of $(i-1)$th codeword can be used during processing of channel symbols corresponding to the ith codeword. Actually, we can do better than this, as the following theorem predicts.

Theorem 5 *If* $c_k = c_{ij}$; $k = Ni+(N-1)j$, $j = 0, 1, 2, \ldots, (N-1)$, *and* $k > (N-1)^2$, *then either* $c_{k-l} = c_{i'j'}$ *with* $i' < i$; *for some* $j' = 1, \ldots, (N-1)$ *(i.e.* c_{k-l} *belongs to a previously transmitted codeword) or* $c_{k-l} = c_{i'0}$ *(i.e.* c_{k-l} *is a sync symbol);* $\forall$ *l s.t.* $l = 1, 2, \ldots, (N-j)$.

Proof:

We prove it by induction using theorem 4. For $l = 1$, the given statement is true, by theorem 4. Let it be true for $l = n$, $n = 1, 2, \ldots, (N-j-1)$. Then, either $k - n = Ni'' + (N-1)j''$ for $i'' < i$ or $k-n = Ni''$, i.e. c_k is a sync. We now rule out the second possibility. Note however, $k - (N-j) = Ni + (N-1)j - (N-j)$ $= N(i-1+j)$. Thus, $c_{k-(N-j)} = c_{(i-1+j)0} \Rightarrow$, $c_{k-(N-j)}$ is a sync. Thus c_{k-n} for $n = 1, 2, \ldots, (N-j-1)$ cannot be a sync since two consecutive syncs should be exactly N symbols apart. Therefore, our assumption for $l = n$ definitely implies $k - n = Ni'' + (N-1)j''$ for $i'' < i$.

Now we investigate the case for $l = n+1$, $c_{k-n-1} = c_{Ni''+(N-1)j''-1}$ and from theorem 4, it follows that either $c_{k-n-1} = c_{Ni'+(N-1)j}$ for $i' < i'' < i$ or c_{k-n-1} is a sync. Thus the statement is true for $l = n+1$. Hence it must be true for all $l = 1, 2, \ldots, (N-j)$. ♣

What this theorem says can be summarized in the following table. Here we assume that r_k is the symbol being processed (demodulated).

Encoded Symbol $c_k \simeq r_k$	Symbols of earlier codewords/syncs
$c_{i(N-1)}$	$c_{k-1}=$ sync
$c_{i(N-1)}$	$c_{k-1} = c_{(i-1)(N-1)}$ $c_{k-2} =$ sync
$c_{i(N-2)}$	$c_{k-1} = c_{(i-1)(N-2)}$ $c_{k-2} = c_{(i-2)(N-1)}$ $c_{k-3} =$ sync
$\vdots$	$\vdots$
c_{i1}	$c_{k-1} = c_{(i-1)2}$ $c_{k-2} = c_{(i-2)3}$ $\vdots$ $c_{k-(N-2)} = c_{(i-(N-2))(N-1)}$ $c_{k-(N-1)} =$ sync

5.2 Processing Using Array of DFEs

Since $r_k \simeq c_k = c_{ij}$, we can also write $r_{ij} = r_k$ then $r_{ij} \simeq c_{ij}$. We have already defined $r(k)$ as $r(k) = [r_{k+K_1}, \ldots, r_{k+1}, r_k]^T$, where $(K_1 + 1)$ is the number of coefficients in FFF. In the same notation, we define,

$$r(k + j(N-1)) \triangleq \left[r_{k+K_1+j(N-1)}, \ldots, r_{k+1+j(N-1)}, r_{k+j(N-1)}\right]^T$$

for $j = 1, \ldots, N-1$. We also had $\hat{d}(k) = \left[\hat{d}_{k-1}, \hat{d}_{k-2}, \ldots, \hat{d}_{k-K_2}\right]^T$, where K_2 is the number of coefficients of FBF. As for $r(k + j(N-1))$, we define

$$\hat{d}(k + j(N-1)) \triangleq \left[\hat{d}_{k-1+j(N-1)}, \hat{d}_{k-2+j(N-1)}, \ldots, \hat{d}_{k-K_2+j(N-1)}\right]^T$$

for $j = 1, \ldots, N-1$. Note that each of the vector pairs

$$\left[r(k + j(N-1)), \hat{d}(k + j(N-1))\right]$$

can be considered as the contents of a DFE. We can form an array of DFEs in which the j^{th} DFE – denoted as DFE$_j$ – contains pair $\left[r(k + j(N-1)), \hat{d}(k + j(N-1))\right]$ in its registers. Let F_j be the feed forward register and B_j be the feedback register of DFEj. Then F_j contains $r(k + j(N-1))$ denoted by $F_j \hookleftarrow r(k + j(N-1))$, also $B_j \hookleftarrow \hat{d}(k + j(N-1))$.

It follows from our discussion in Chapter 4, that the symbol being demodulated by DFE$_j$ is $r_{k+j(N-1)} \simeq c_{k+j(N-1)}$. The corresponding prediction is $\tilde{d}_{k+j(N-1)}$ (thus $r_{k+j(N-1)} \simeq \tilde{d}_{k+j(N-1)} \simeq c_{k+j(N-1)}$). We can stack the predictions from all the DFEs of the array in a vector

$$\tilde{\underline{d}}_k \triangleq \left[\tilde{d}_{k+(N-1)}, \tilde{d}_{k+2(N-1)}, \ldots, \tilde{d}_{k+(N-1)(N-1)}\right].$$

If the channel is free of noise and ISI, then $\tilde{\underline{d}}_k$ should be same as the encoded symbols' vector

$$\underline{c}_k \triangleq \left[c_{k+(N-1)}, c_{k+2(N-1)}, \ldots, c_{k+(N-1)(N-1)}\right].$$

We say that $\tilde{\underline{d}}_k \simeq \underline{c}_k$. For $k = Ni$, $c_{k+j(N-1)} = c_{ij}$, therefore we have

$$\underline{c}_k = \underline{c}_i \triangleq \left[c_{i1}, c_{i2}, \ldots, c_{i(N-1)}\right],$$

which is the i^{th} codeword. Hence, at time intervals $k = Ni$, the DFE array defined above, processes symbols corresponding to the same codeword.

Suppose we are provided with decision vectors $\hat{d}(k + j(N-1))$ for $j = 1, \ldots, N-1$ and $k = Ni$. Then all the DFEs in the array collectively predict the $(N-1)$ dimensional prediction vectors $\tilde{\underline{d}}_{(Ni)}$. This vector ($\tilde{\underline{d}}_{(Ni)} \simeq \underline{c}_i$) is taken by the decoder which soft decodes $\tilde{\underline{d}}_{(Ni)}$ and provides the estimated codeword $\breve{\underline{c}}_i = \left[\breve{c}_{i1}, \breve{c}_{i2}, \ldots, \breve{c}_{i(N-1)}\right]$. When $\underline{c}_i = \breve{\underline{c}}_i \Rightarrow$ the decoder made error-free decision. These decoded symbols can be re-modulated to form $\check{\underline{d}}_i = \left[\check{d}_{i1}, \check{d}_{i2}, \ldots, \check{d}_{i(N-1)}\right]$, where $\check{d}_{ij}$ is the modulated version of $\breve{c}_{ij}$. Note that we have modified the decision mechanism of the DFEs in the array. Rather than replacing $\tilde{d}_{ij}$ by the closest modulation point $\hat{d}_{ij}$ in the single (possibly complex) dimensional constellation — local estimate, we replace the complete vector $\tilde{\underline{d}}_i$ by a closest codeword point $\check{\underline{d}}_i$ in the $(N-1)$ (complex) dimensional constellation — global estimate. The local estimate $\hat{d}_{ij}$ is not necessarily equal to $\check{d}_{ij}$. The decoding algorithm, working on a global basis, might reverse adverse local estimates. Note that the DFE array is capable of processing $(N-1)$ symbols at a time.

In order to process the next block of symbols [1] the DFE array is updated as follows,

[1] Here we are not interested in processing the sync symbol which may or may not contain information. If it contains information it has to be demodulated separately in the scheme discussed in this section.

1. registers of DFE_j are shifted as usual:
   - an appropriate fresh channel symbol is shifted into F_j.
   - $\check{d}_{ij}$ is shifted into B_j

2. the contents of DFE_j are transferred to DFE_{j-1}, $j = 2, 3, \ldots, (N-1)$. After this transfer has taken place, the contents of the registers of the updated DFE array are as follows,

$$\begin{aligned} F_{j-1} &\hookleftarrow r(k + j(N-1) + 1) \\ B_{j-1} &\hookleftarrow \left[\check{d}_{k+j(N-1)}, \hat{d}_{k-1+j(N-1)}, \ldots, \hat{d}_{k-K_2+1+j(N-1)}\right]^T ; \end{aligned}$$

   for $j = 2, \ldots, (N-1)$. Note that $k + j(N-1) + 1$, for $k = Ni$, is $Ni + j(N-1) + N - (N-1) = N(i+1) + (N-1)(j-1)$. Which shows that, $\forall j = 1, \ldots, (N-2)$, DFE_j now contains the received symbol $r_{(i+1)j} \simeq c_{(i+1)j}$, as the last entry in the register $F_{(j-1)}$. This implies that these DFEs are ready to process the symbols of the next codeword $\underline{c}_{(i+1)}$.

3. The coefficients of the filters are transferred along with the contents of the registers.

4. The contents of DFE_{N-1} are still unknown. The F_{N-1} is provided with $r(k+(N-1)(N-1)+N)$. For $k = Ni$, $k+(N-1)(N-1)+N = N(i+1)+(N-1)(N-1)$, so that the last entry of F_{N-1} is $r_{(i+1)(N-1)} \simeq \underline{c}_{i+1}$.

After the update, the last entry in F_j is $r_{(i+1)j}$; $j = 1, 2, \ldots, (N-1)$. If we *assume* [2] that the decision vector $\hat{d}(k + (N-1)(N-1) + N)$ is known, the DFE array can form the prediction vector $\bar{\underline{d}}_{(k+N)} \simeq \underline{c}_{(i+1)}$; $k = Ni$. The decoder can then estimate the closest codeword $\check{\underline{c}}_{i+1}$ to the prediction vector $\bar{\underline{d}}_{k+N}$. The whole process can then be repeated for future codewords. Note that when we initiate the process, we assume the availability of all contents of B_j for all js. It is clear from the update mechanism that after processing the very first block, contents of all the B_js are available except for $B_{(N-1)}$.

Note that the first element of the assumed vector — $\hat{d}(k + (N-1)(N-1) + N)$ — is $\hat{d}_{k-1+(N-1)(N-1)+N}$. For $k = Ni$, $k - 1 + (N-1)(N-1) +$

[2]This assumption will be relaxed latter in discussion

$N = N(i + (N-1))$. Thus $\hat{d}_{k-1+(N-1)(N-1)+N}$ is the sync symbol. For $j = 1, 2, \ldots, (N-2)$, the first element of the decision vector $\hat{d}(k + N + j(N-1))$; the first entry in B_j, is the modulated version of earlier decoded codeword symbols $\check{\underline{d}}_{ij}$. If we assume that the decoder corrects all errors, i.e. $\check{\underline{c}}_i = \underline{c}_i$, we can expect better feedback.

After processing $(N-1)$ consecutive codewords, the first $(N-j-1)$ symbols in B_j are decoded symbols of earlier codewords for $j = 1, 2, \ldots, (N-2)$, and, the $(N-j)$th symbol is the sync for $j = 1, 2, \ldots, (N-1)$. This is exactly what theorem 5 predicted. Both the sync and the decoded symbols can be considered relatively reliable, if the decoder is operating in a region of reliability relative to the uncoded channel. The contents of the registers for two consecutive operations for the DFE array are shown in figure 5.3 for $N = 4$. Only the indices of the symbols are shown. The register B_j is more reliable than the register $B_{j'}$, when $j < j'$, because it holds relatively unreliable decisions and they are further back in time relative to the symbol being predicted.

For the DFE array, please note that:

1. The DFE's have to operate only at a speed $\frac{1}{N}$ times the rate of transmission.

2. In terms of implementation, the register contents and coefficients do not have to be "transferred" from DFE$_j$ to DFE$_{(j-1)}$. The registers can be arranged in a circular area of memory with a pointer to each register. The transfer can then be replaced by "decrementing" each register's pointer modulo $(N-1)$.

3. The DFE array does not process the sync symbol. If the sync does contain information, alternative arrangements have to be made for its demodulation.

In the above discussion, we have made the assumption that the decision vector $\hat{d}(k + (N-1)(N-1))$, $k = Ni$ is available. Its availability is not essential (we will soon discuss a technique in which it would be available). If $\hat{d}(k + (N-1)(N-1))$ is not available, one can still process the received symbol, however, full length feedback filter is not possible with the DFE array. In this case the length of FBF for the j^{th} DFE would be restricted to $(N-j)$. In the conventional DFE, if the length of FBF is changed, the optimal coefficients of the FFF change as well. Therefore, when we want to use conventional DFE, for the case where the vector $\hat{d}(k+(N-1)(N-1))$ is not available, the optimal coefficients for the DFE_i will be different from those of DFE_j, for $i \neq j$.

In order to remedy a similar problem in his receiver, Eyubogolu [3] suggested use of an inferior version of DFE called noise-prediction DFE [15], in which the FFF coefficients are not effected by the size of FBF. We however, do not need that since we provide the decision vector $\hat{d}(k+(N-1)(N-1))$ from a preliminary DFE as discussed in next section.

5.3 Our Codesigned Receiver

Our codesigned receiver has four main components,

1. A conventional DFE operating at the transmission speed. This DFE will be referred to as *fast* DFE.
2. An array of $(N-1)$ DFEs as discussed in previous section. The DFEs in the array are referred to *Slow DFEs.*
3. Soft Decoder
4. Modulator: which re-modulates the decoded codeword.

These components are shown in figure 5.4 for $N = 4$. The operation of 2, 3, and 4 has already been discussed in previous section.

We now describe how the fast DFE fits into the picture. The fast DFE does not enjoy any advantage over that of a conventional DFE — it is not trusted to be as correct as the slow DFEs. It processes the channel symbols $\{r_k\}$ in the conventional process as discussed in chapter 4. It operates at transmission symbol rate, predicts the value of a transmission symbol and replaces the prediction by that of the closest modulation symbol (local estimate). This demodulated symbol is fed into the decision register. The DFE then takes a new sample into the feed forward register and repeats the process. Let the feed forward and the feedback registers be F and B, respectively. As soon as the fast DFE has processed the received symbol corresponding to a sync symbol (i.e. after processing the symbol r_k for $k = Ni$, and shifting the demodulated symbol in B), the contents of F and B are transferred to

$F_{(N-1)}$ and $B_{(N-1)}$, respectively, along with corresponding filter coefficients. This is how we provide the decision vector $\hat{d}(k + (N-1)(N-1))$, where $k = Ni$, to the DFE$_{(N-1)}$. The DFE array processes as discussed in previous section, whereas, the fast DFE continues its processing in parallel, dumping its contents to DFE$_{(N-1)}$ after processing every sync symbol. Hence, the fast DFE is just a preliminary receiver that works without reference to the code structure and makes crude decisions. However, the slow equalizers use the code structure and they improve the quality of decisions.

This complete the description of our codesigned receiver. We will make some final remarks,

1. We do not need special processing for the sync symbol when it contains information

2. The technique does not depend upon the type of coding or modulation.

3. The fast and slow DFEs can be made adaptive.

4. The functions applied by the FFFs and FBFs of the DFEs are linear here, but they can as well be non-linear.

Chapter 6

Simulation results and Analysis

In this chapter we present the simulation results and analysis. The most interesting simulations are for 8-PSK modulation with Half Leech Lattice (HLL) codes. However, in order to show the utility of the technique for binary systems, we have also included results for BPSK system with (8,4) Reed-Muller code. Reed Muller code was used primarily for its simplicity of encoding and decoding. One can improve the performance to greater extent by using other complicated but more efficient binary codes. For the binary system three channels selected from published literature are selected. For the 8-PSK system with HLL, extensive simulations are made for a class of channels with characteristics of the form $(1 + aD)(1 - bD)$.

For both the binary case, performance of three different systems are compared,

1. Uncoded System: As described in Chapter 1. We call this system *System 1.* This is the bench mark system with respect to which we compute all gains of other coded systems.

2. Coded System with Simple Cascade: This is the coded system described in Chapter 1. However, the receiver of this system constitutes a simple cascade of conventional DFE, de-interleaver and Decoder. The receiver of this system does not attempt to reduce error propogation of the DFE with the help of coding. Therefore, its performance serves as a lower bound on the performance of Our Codesigned receiver. Block diagram

for this system is shown in figure 5.1. This system will be referred to as *System 2*.

3. Coded System with codesigned receiver: This is the coded system described in Chapter 5. It will be referred to as *System 3*.

For the 8-PSK case in addition to the above systems, performance of Coded System with perfect Feedback (i.e. the receiver contains a DFE whose Feedback Filter is provided with the actual decisions.) is also provided. This is a hypothetical system which provides an upperbound on the performance. We will refer to this system as *System 4*.

Figure 6.1 lists all types of systems considered in this research. Simulations are made for a wide range of Signal to Noise Ratios ($\frac{E_b}{N_0}$) in each case and Bit Error Rate (BER) are estimated. In order to obtain a fair comparison the $\frac{E_b}{N_0}$ for coded systems is compensated for coding rate. The BER are plotted as the function of $\frac{E_b}{N_0}$ (in dBs). The difference between $\frac{E_b}{N_0}$ at a particular BER, for System 1 and 2, gives the gain due to coding and Interleaving. We call this gain *Coding gain*. The difference between $\frac{E_b}{N_0}$s of Systems 2 and 3 gives the gain due to *Codesigning* the receiver by our particular technique. We shall call this gain the *Technique gain*. Finally, the difference between $\frac{E_b}{N_0}$s of Systems 1 and 3 gives the over gain of the system. Note

$$\text{Technique gain} + \text{ Coding gain} = \text{Overall gain.}$$

6.1 Simulation Results for the Binary Systems

We will first discuss the characteristics of the channels selected for the Binary Systems. In section 6.1.2 simulations are presented.

6.1.1 Characteristics of channels

Channel A was obtained from a published paper [3]. The channel models a telephone leased line with some pre-equalization at the transmitter. For this channel the the performance of DFE is very close to that of the matched filter bound leaving little room for improvement. In this channel ISI is quite small, therefore , the error propagation in the DFE is very small.

The channels B and C model digital magnetic recording systems as discussed by Bergmans in [28]. Their channel transfer function $|F(w)|$ satisfies

$$|F(w)| = \sqrt{2\cosh D(\pi - |w|)};\ \text{for}|w| \leq \pi.$$

Here the parameter D characterizes the spatial information Density on the Magnetic Storage and ranges roughly between 0.2 and 3.0 in present systems. Apart from a zero at the lower band edge ($w = 0$) the transfer functions in this class have attenuation near the upper band edge ($w \simeq \pi$) which increases rapidly with the information density D. Channels B and C correspond to $D = 2.0$ and $D = 3.3$, respectively. Their impulse responses and frequency characteristics are shown in figure 6.2. Channel B represents a relatively low density system whereas channel C represents the one with high density. From previous research [23] we know that channel B produces less amounts of error propagation as compared to channel C. Channel C was chosen for its harshness observed in previous research. In fact, it failed to respond well to processing by much more complicated receivers [23][28]. Error propagation is a serious problem for this channel. The effect of error propagation for this channel can be seen from the DFE performance which shows a 5-6dB degradation versus the matched filter bound (Figure 6.5).

6.1.2 Analysis of Simulations

Channel A Figure 6.3 gives the performance curves for System 1, System 2 and System 3. The conventional DFE, for the uncoded system, (System 1) for Channel A almost achieves the matched filter bound indicating that their is very little energy present in the ISI part of the received signal (figure 6.4). Since, the error propagation is insignificant, very small improvement over System 2 is obtained by using codesigned receiver (System 3). The Coding gain in this case is 1.65dBs at BER= 10^{-3} whereas the Overall gain is about 1.8 dBs. Thus, the performance gain in this case is mostly due to coding.

Channel B Figure 6.4 shows that the Conventional DFE (System 1) does suffer a degradation of about 1.7dBs relative to the matched filter bound. A part of this degradation is due to error propagation. The other two curves (Systems 2 and 3) show that significant improvement (1.4dBs) over System 2 is obtained by using the codesigned receiver (Technique agin), whereas the coding gain is about (0.3 dBs).

Channel C This is the worst channel that we used. The three simulation curves are presented in figure 6.5. The performance of a simple combination of coding interleaving and DFE (System 3) is hopeless and its performance is about 2.0 dBs worse than that of the Conventional DFE. This means that coding gain is -2.0 dBs. The Co-designed receiver, however, performs about 1dB better than the DFE (Overall gain). It is obvious that error correction in the feedback path of the DFE, as provided by the codesigned receiver, is absolutely essential for this channel, when the coding schemes are to be used. The complexity of our receiver is roughly proportional to that of two conventional receivers and a decoder which is quit low as compared to receivers used earlier for this channel [23]. However, high $\frac{E_b}{N_0}$s performance approaches (and probably surpasses at higher $\frac{E_b}{N_0}$s due to typical coding characteristics) to that of much more complicated receivers used in [23].

The above results are summarized in the following table.

Channel	$\frac{E_b}{N_0}$ at BER= 10^3 in dBs			Gains in dBs		
	Sys. 1	Sys. 2	Sys. 3	Coding	Techn.	Overall
A	6.95	5.3	5.15	1.65	0.15	1.8
B	8.3	8.0	6.6	0.3	1.4	1.7
C	11.6	13.6	10.9	-2.0	2.9	0.9

6.2 Simulation Results for the 8-PSK Systems

In order to evaluate the performance we performed simulations for a family of channels. First, this family is described, and then, the simulation results are presented.

6.2.1 Family of Channels

A family of channels whose characteristics are of form $(1 + aD)(1 - bD)$ are selected. The parameters a and b range between 0 and 0.9 with increments of 0.1. This corresponds to a channel with a two zeros in the *D plane* as shown in figure 6.6. When $a = b = 0$ both zeors are at infinity, and the channel reduces to a simple Additive White Gaussian Noise channel with no ISI. When either a or b is zero, the channel channel has one zero which moves from infinity towards the unit circle as the non-zero parameter increases from zero to 0.9. Worst channels are obtained when eihter or both zeros are on the Unit circle.

6.2.2 Simulation results, Analysis and Bounds

In order to make reference to the figures convenient for the reader, we make a table for the figures. In this table, by *absolute performance* we mean the E_b/N_0 (in dBs) at which BER of 10^{-3} is achieved, where as, by *performance of System X relative to System Y* (Gain achieved by System X relative to System Y), we mean the difference between E_b/N_0 of System X and E_b/N_0 of System Y (in dBs) at BER of 10^{-3}.

Performance				
		relative to		
of	absolute	System 1	System 2	System 3
System 1	Fig. 6.7			
System 2	Fig. 6.8	Fig. 6.11		
System 3	Fig. 6.9	Fig. 6.12	Fig. 6.14	
System 4	Fig. 6.10	Fig. 6.13	Fig. 6.15	Fig. 6.16

All the simulation results are presented as 3-D plots, where the Vertical axis is for E_b/N_0 in dBs, and the other two axis are for parameters a and b respectively. The plane spanned by these two axis will be referred to as horizontal plane. The simulations were carried out using C language for a large range of E_b/N_0 for every channel and corresponding BERs were tabulated. The results were interpolated and values for E_b/N_0 corresponding to BER of 10^{-3} were obtained. Mat-lab software was used to obtain the 3-D plots. Each figure shows a 2-D surface in 3-D, the lowest point of which should be visualized to be towards the reader. Every point (α, β) in the horizontal plane corresponds to a channel parameters $a = \alpha, b = \beta$ i.e. the channel D-transform is $1 + (\alpha - \beta)D - \alpha\beta D^2$. The hieght of the 2-D surface at (α, β) is the absolute or relative performance of the corresponding channel. The figures should not be compared to each other interms of the Vertical axis: each has its own scale for the Vertical axis. The plots are symetrical (wihtin experimental error) which is due to the fact that changing the sign of any channel coefficient of a channel does not change the performance.

Figure 6.7, 6.8, 6.9 and 6.10 give performance curves for Systems 1, 2, 3 and 4, respectively. It is obvious from these curves that the performance degrades as the parameters a or/and b increase because the error propagation increases. The worst performances are obtained for $(a = 0.9, b = 0)$, $(a = 0, b = 0.9)$ and $(a = 0.9, b = 0.9)$. Note that the left peak corresponds to $(a = 0, b = 0.9)$, where as the right peak corresponds to $(a = 0, b = 0.9)$. The height of the two peaks are equal within the experimental error. The center peak (which should be visualized to be away from the reader) corresponds to $(a = 0.9, b = 0.9)$ and is slightly lower from the other two. Note that the horizontal plane is tilted at an angle of 20°. We summarize the performances and Gains of three points: $(a = 0, b = 0)$, $(a = 0, b = 0.9)$ and $(a = 0.9, b = 0.9)$ for all the 3-D plots.

	(a , b)			
Figure	$(0,0)$	$(0,0.9)$	$(0.9,0.9)$	**Brief Explaination**
Fig. 6.7	11.0	23.1	19.4	System 1: Uncoded system
Fig. 6.8	7.7	19.0	15.7	System 2: Upper bound
Fig. 6.9	7.7	14.9	14.1	System 3: Our Codesigned receiver
Fig. 6.10	7.7	10.3	9.9	Ststem 4: Lower bound
Fig. 6.11	3.3	4.1	3.7	Coding Gain
Fig. 6.12	3.3	8.7	5.3	Overall Gain
Fig. 6.13	3.3	12.8	9.5	Upper Bound on Overall Gain
Fig. 6.14	0.0	4.6	1.6	Technique Gain
Fig. 6.15	0.0	8.7	5.8	Upper Bound on Tech. Gain
Fig. 6.16	0.0	4.2	4.1	Error Propagation loss Unrecovered by the codesigned receiver.

Performance of System 1 (the uncoded System) serves as the bench with respect to which the Gains of variuos systems are calculated. Figure 6.11 shows the gain obtained w.r.t. System 1 by System 2 which has a simple cascade of DFE, Interleaver and Decoder. The receiver in this system makes no attempt to use the decisions made by the decoder in order to reduce the error propagation. This system does, however, uncorrelates the error bursts that occur due to error propagation. Thus, the gain obtained is solely due to Coding and De-interleaving. Notice that there is very little change in coding gain (within 0.8 dBs) with parameters a and b.

Figure 6.14 gives the gain of System 3 w.r.t. System 2. System 3 is our Co-designed receiver which utilizes the coding in order to reduce the error propagation. Gain obtained by this system over that of System 2 is solely due to the reduction in the error propagation. Thus It gives the *Technique Gain.* Several observations must be made from this figure. There is a large variation (about 4.6 dBs) in the Technique Gain. It is highest for the points $(a = 0.0, 0.9)$ and $(a = 0.9, b = 0.0)$ indicating that the technique works at its best for channels with short impulse responses. Note that the technique gain is about 1.6 dBs at point $(a = 0.9, 0.9)$ which looks low at the first sight. Improvement of the Technique gain for such channels (which require more than one feedback coefficient in the FBF of DFE) is the topic of our future research. We are quite confident that Significant improvements can be made for such channels. Figure 6.15 gives an upper bound on the Technique Gain i.e. ultimate gain possible by a receiver that reduces the error propagation, in fact, it is the gain provided by a receiver that completely eliminates the error propagation. Figure 6.16 gives the difference of figures 6.14 and figures 6.15. It indicates the weakeness of our receiver. It points out that the our receiver was unable to obtain high gains when both parameters a and b where high.

Figure 6.12 shows the Overall gain obtained by our Codesigned receiver. Figure 6.13 gives the upper bound for the same.

6.3 Discussion

The above results clearly show that the Codesigned receiver is quite successful in reducing error propagation in channels with but severe ISI that spans over a small number of neighboring channels. They also shed light over the weakenesses of the Codesigned receiver. The Gains abtained here are expected to increase even more dramaticaly when a and b move closer to one. The Complexity of the receiver is equal to that of a decoder plus two conventional DFEs.

Chapter 7

Future Work

Here we discuss the directions in which the future work will be carried out. We will primarily concentrate on the 8-PSK with HLL system. The suggested avenues include bounds on the performance, further simulations, and techniques for further improvement in the performance. Section 7 discusses a technique which can has the potential of improving the performance for channels that require DFEs with large number of feed back coefficients. Finally, section 7.1 discusses possible improvements n the adaptation algorithm. The number of corrected decisions in the Feed Back Filter of ith slow DFE is greater than that for jth slow DFE when $i > j$. Therefore, the ith DFE suffers less from error propagation as compared to the jth DFE. This in turn implies that the decisions made by the jth DFE are more reliable than those made by the ith DFE. Decisions from all slow DFEs are provided to the soft HLL decoder. Since, they have unequal reliability, we can weigh them with unequal weights prior to soft decoding so that more reliable decisions are weighted by higher weights. By doing this we can improve the performance of the decoding algorithm.

7.1 Improvement for the adaptation Algorithm

We have discussed that the fast and slow equalizers can be made adaptive. Since, the decisions of the last slow DFEs has the highest reliability. Thus, the coefficients obtained in this equalizer are also most reliable. However, in the current form of the scheme, these coefficients are thrown away, and do not contribute in the overall adaptation process. Therefore, the reliability of the coefficients is determined dominantly by the fast DFE. We want to investigate a scheme were the more reliable coefficients of the last slow equalizer are utilized to improve the quality of the coefficients of the over all system.

Bibliography

[1] G. Ungerboek, *Channel coding with multi-level/phase signals,* IEEE Trans. Inform. Theory, vol. IT-28, pp. 55-67, Jan. 1982.

[2] G.D.Forney Jr., *Coset Codes I: Introduction and geometrical classification,* IEEE Trans. Inform. Theory, vol. 34, no. 5 pp. 1123-1151, Set. 1988.

[3] M. Vedat Eyuboglu, *Detection of Coded Modulation Signals on Linear, Severly Distorted Channels Using Decision-Feedback Noise Prediction with Interleaving,* IEEE Trans. Comm., vol.36, no.1 pp.13-20, Jan. 1988.

[4] K. Zhou, J. G. Proakis, F. Ling, *Decision-Feedback Equalization of time-Dispersive Channels with Coded Modulation*, IEEE Trans. Comm. vol. 38 no.1 Jan. 1990.

[5] G.D.Forney, Jr., A.R.Calderbank, *Coset codes for partial response channels; or, Coset codes with Spectral nulls,* IEEE Trans. Inform. Th., vol.35 no.5, pp.925-943, Sept.1989.

[6] V.M.Eyuboglu, G.D.Forney, Jr. *Trellis precoding* in ARO Comm. Workshop, Ruidoso, NM, May 1989.

[7] A.R.Calderbank, J.E.Mazo, *Baseband line codes via spectral factorization* IEEE J. Selected Areas Comm. vol. SAC-7, 1989.

[8] H. Nyquist, *Certain topics in telegraph transmission theory*, AIEE (Comm. Electron.), vol. 47, pp. 617-644. Apr. 1928.

[9] M. Tomlison, *New automatic equalizers employing modulo arithmatic,* Electron. Lett., vol. 7 no.3, pp.138-139, Mar. 1971.

[10] S.Kasturia, J.T.Aslanis, J.M.Cioffi, *Vector Coding for Partial Response Channels,* IEEE Trans. Inform. Th., vol.36 no.4, pp.741-762, Jul.1990.

[11] E.Hansler, *Some properties of transmission systems with minimum mean-square-error,* IEEE Trans. Comm. Tech., vol. COM-19, pp.576-579, Aug. 1971.

[12] M.R.Aaron, D.W. Tufts *Intersymbol interference and error probability,* IEEE Trans. Inform. Theory, vol. IT-12, pp. 26-34, Jan.1966.

[13] K.Yao, *On the minimum average probability of error expression for binary pulse communication systems with intersymbol interference*, IEEE Trans. Inform. Th., vol.IT-17, pp.528-531, May 1971.

[14] K.Abend, B.D.Fritchman, *Statistical detection for communication channels with intersymbol interference,* Proc. IEEE, vol.58, pp.778-785, May 1970.

[15] J.G.Proakis, *Digital Communications*, McG raw-Hill, 2nd Edition, 1989.

[16] J.K. Omura, *Optimal receiver design for convolution codes and channels with memory via control theoretical concepts,* Inform. Sci., vol.3, pp.243-266, Jul.1971.

[17] G.D.Forney,Jr., *Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference,* IEEE Trans. Inf. Theory IT-18, 363 (1972).

[18] S.U.Qureshi and E.E.Newhall, *An adaptive receiver for data transmission over time-dispersive channels*, IEEE Trans. Inform. Th., vol. IT-19, pp.448-457, Jul.1973.

[19] D.D. Falconer and F.R.Magee, Jr., *Adaptive channel memory truncation for maximum-likelihood sequence estimation,* Bell Syst. Tech. J., vol.52, pp.1541-1562, Nov.1973.

[20] Vermuelen and M.E.Hellman, *Reduced-state Viterbi decoding for channels with intersymbol interference,* Proc. ICC 1974, Minneapolis, MN, Jun.1974.

[21] G.J.Foschini, *A reduced-state variant of Maximum-likelihood sequence detection attaining optimum performance for high signal-to-noise ratio performance,* IEEE Trans. Inform. Th., vol.IT-24, pp.505-509, Sept. 1977.

[22] A.E.Duel, C.Heegard, *Delayed Decision-Feedback Sequence Estimation*, 23rd Annual Allerton Conf. on Communication, Control and Computing, Monticello, IL. 1985.

[23] J.W.M.Bergmans, S.A.Rajput, F.A.M.Van de Laar, *On the Use of Decision Feedback for Simplifying the Viterbi Detector,* Philips J. Res. 42, pp. 399-428, 1987.

[24] M.V.Eyuboglu, S.U.Qureshi, *Reduced-State Sequence Estimation for Trellis-coded Modulation On Intersymbol Interference Channels,* Proc. of Globecom, 1986, Houston, Texas, Dec. 1986.

[25] C.A.Belfoire, J.H.Park, Jr., *Decision Feedback Equalization,* Proc. IEEE, vol.67 no.8, pp.1143-1156, Aug.1979.

[26] F. Ling and S.U.Qureshi, *Lattice-predictive decision-feedback equalization,* presented at Proc. GLOBECOM '86, Houston, Texas, Dec.1986.

[27] F.J.MacWilliams, N.J.Sloane, *The theory of Error Correcting Codes*, North-Holland, 1977.

[28] J.W.M.Bergmans, *Discrete-Time Models for Digital Magnetic Recording* Philips J. Res.41,531 (1986).

[29] R.Price, *Nonlinearly feedback-equalized PAM vs capacity for noisy filter channels* , Int. Conf. Communications, ICC-72 (Philadelphia,PA), 1972.

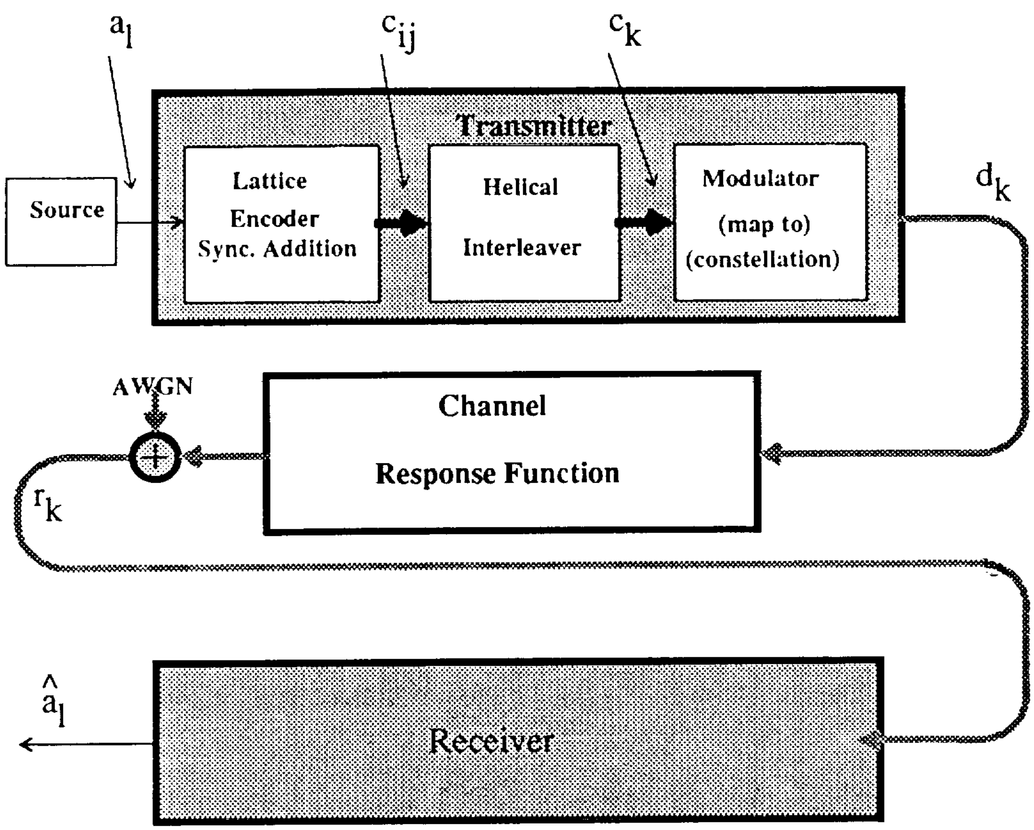


Figure 1.1 Block Diagram for the system with channel encoding scheme.

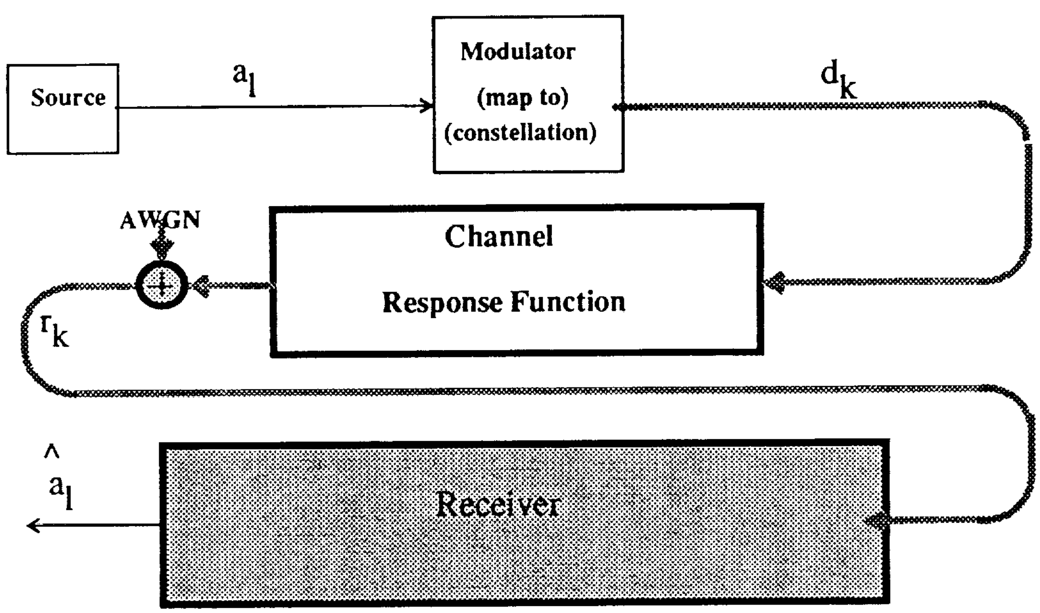


Figure 1.2 Block Diagram for the system without channel encoding scheme.

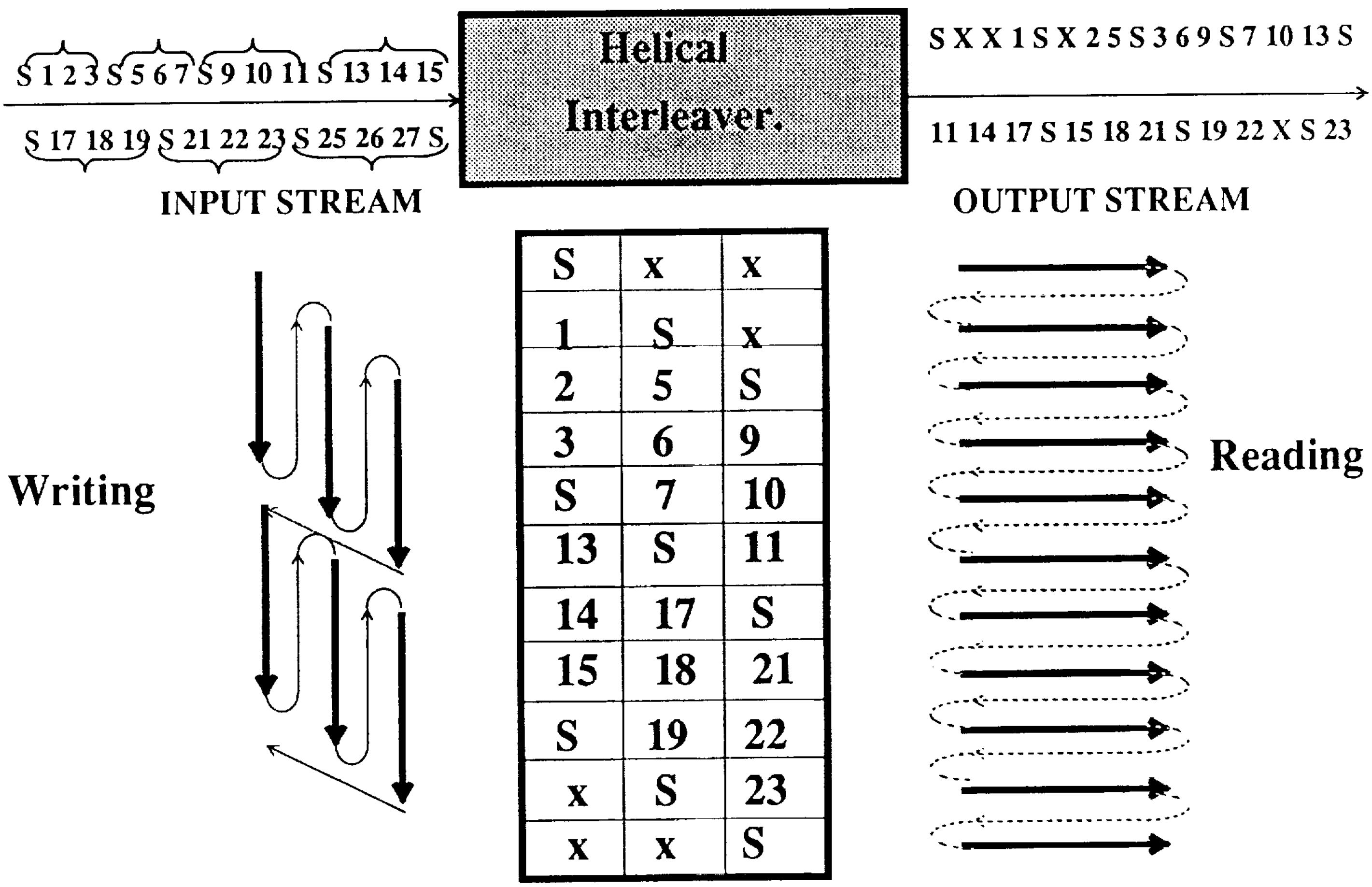


Figure 2.1 Example of helicl interleaving for a code with codewords of length 3.

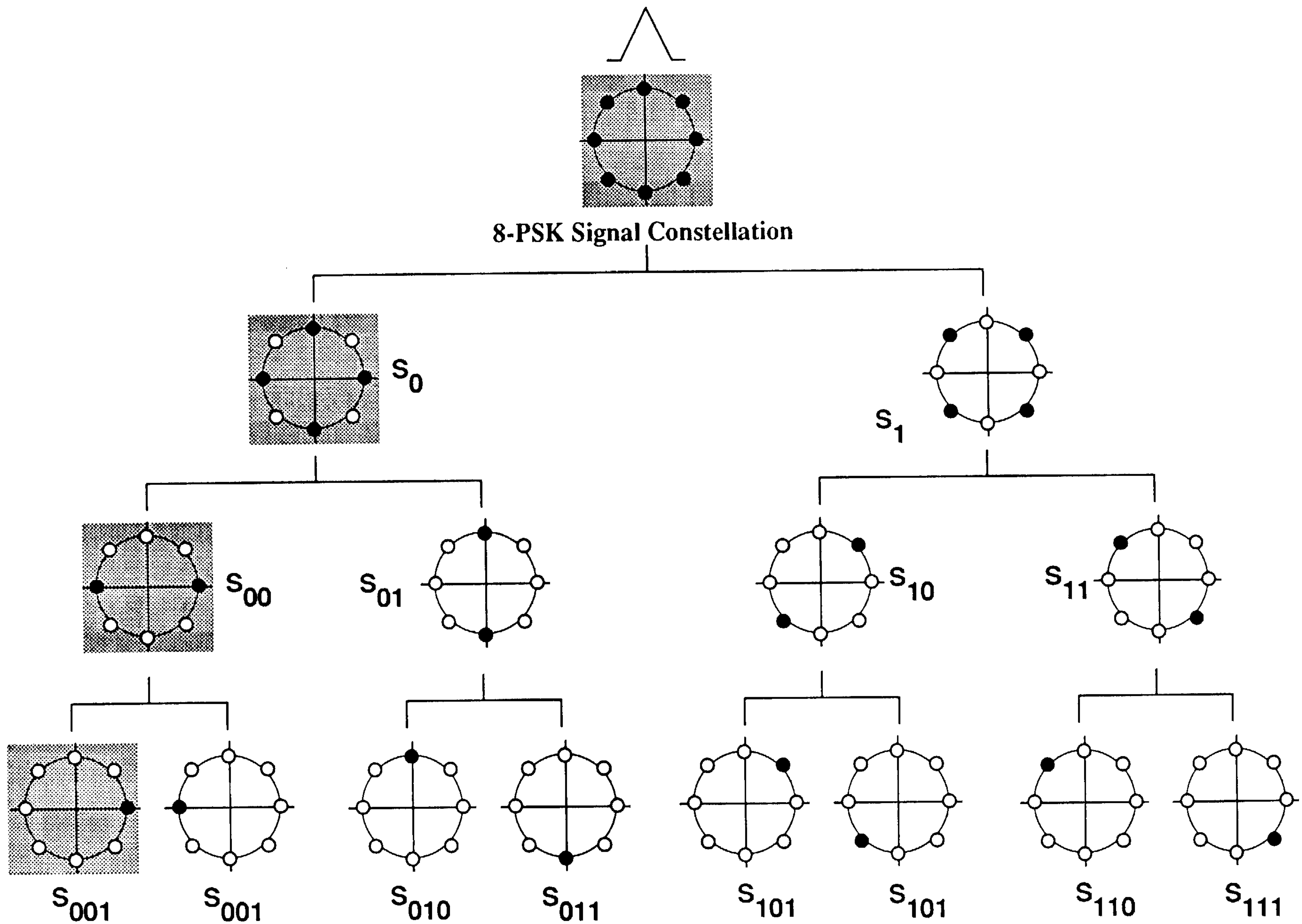


Figure 3.1 Signal Constellation for 8-PSK and Set partitioning.

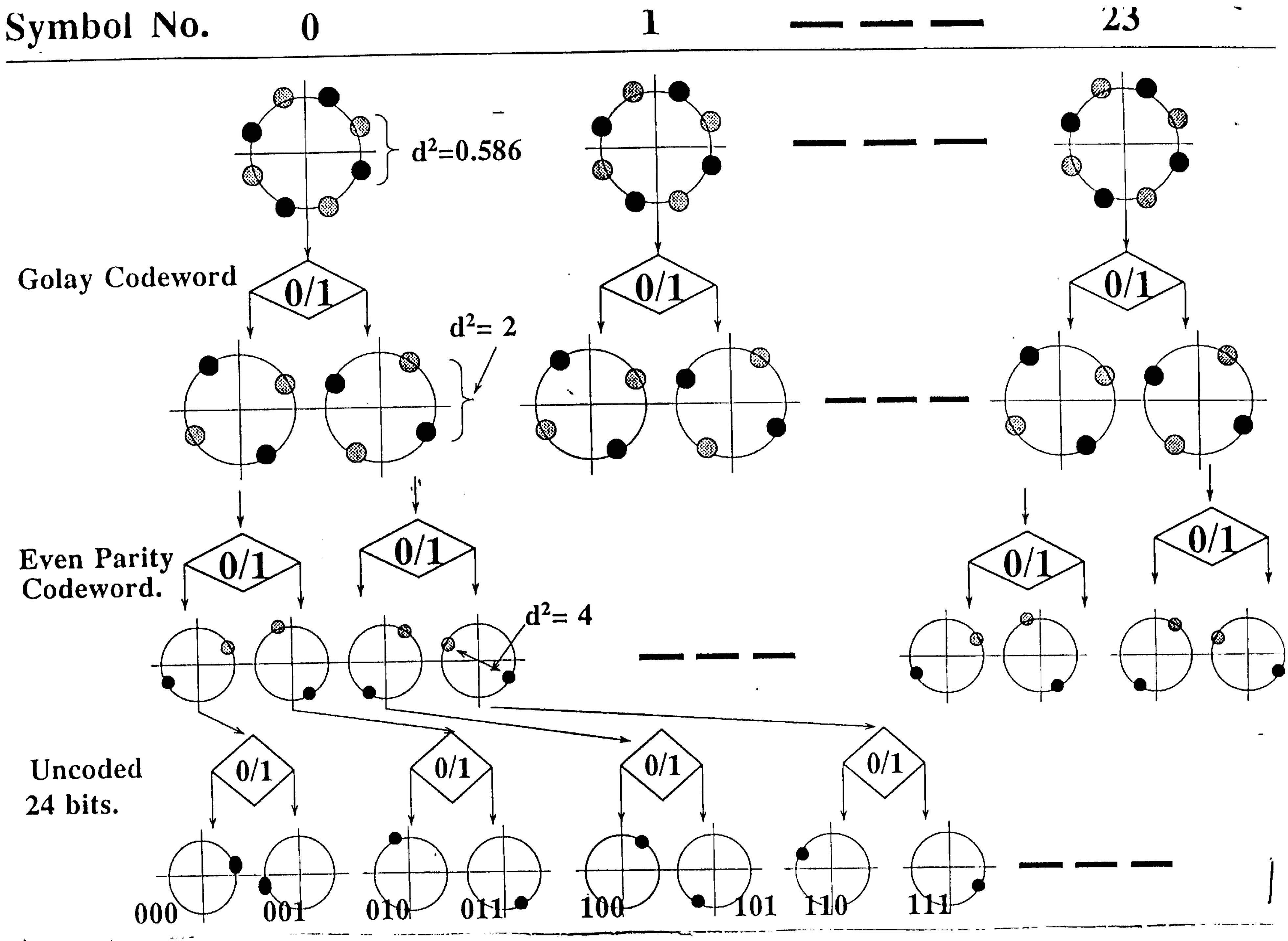


FIGURE 3.2 Encoding a Codeword for HLL.

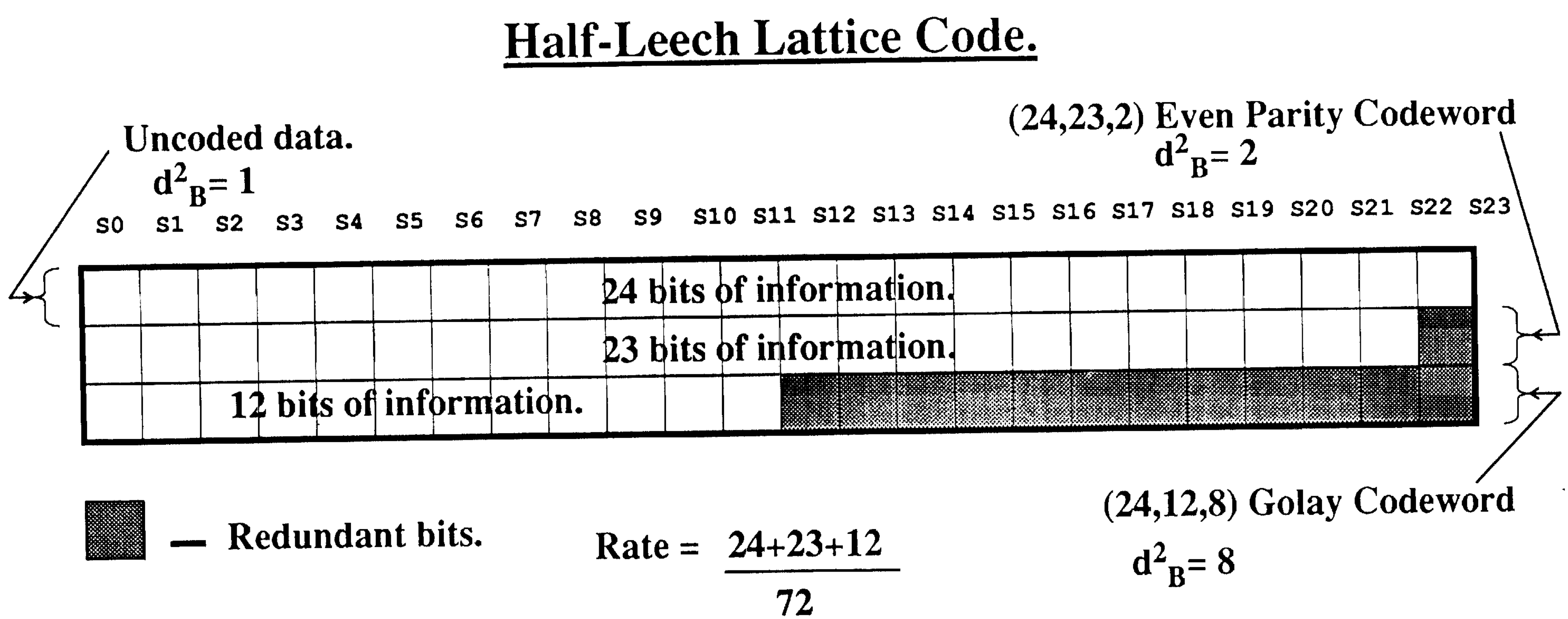


Figure 3.3 Encoding Process for Half Leech Lattice Code.

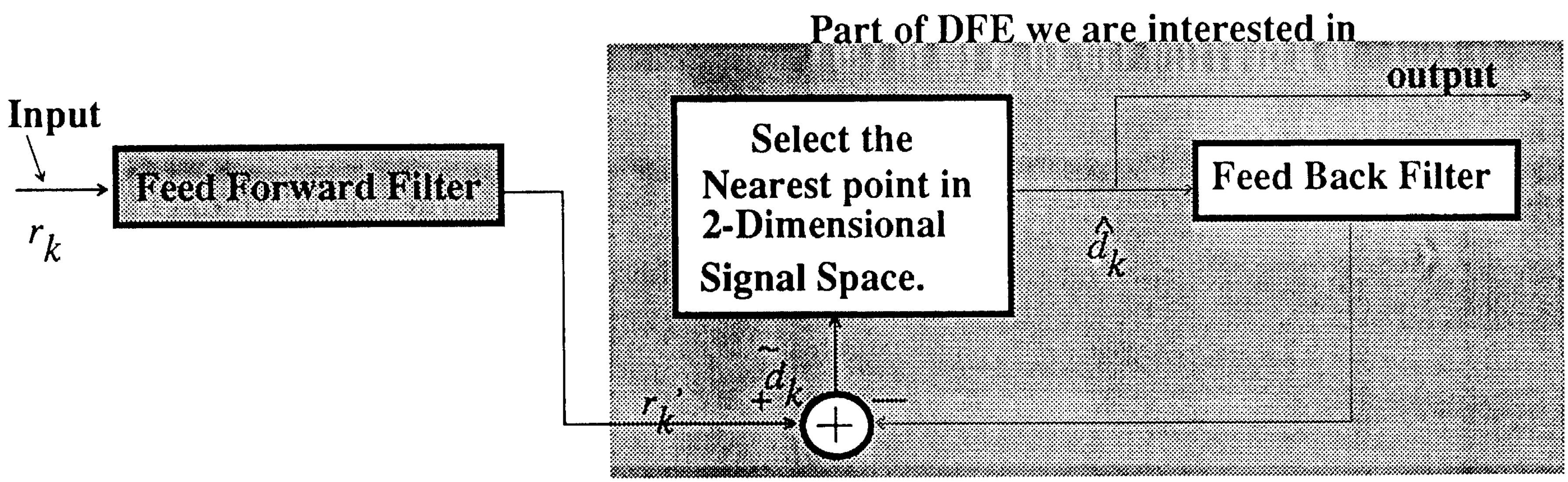


Figure 4.1 Schematic Diagram for Decision Feedback Equalizer

Error sequence at the output of the ideal canceller:
x -- Error.

0 0 0 x 0 0 0 0 0 0 0 0 0 0 x 0 0 0 0 0 x 0 0 0 0 0 0 0 0 0 0 0 x 0 0 0 0 0

Figure 4.3(a)

Error sequence at the output of the DFE:

0 0 0 x 0 x x 0 0 0 0 0 0 0 x x x 0 0 x 0 0 x 0 0 0 0 0 0 0 0 x x x x x 0

Error Bursts

Figure 4.3(b)

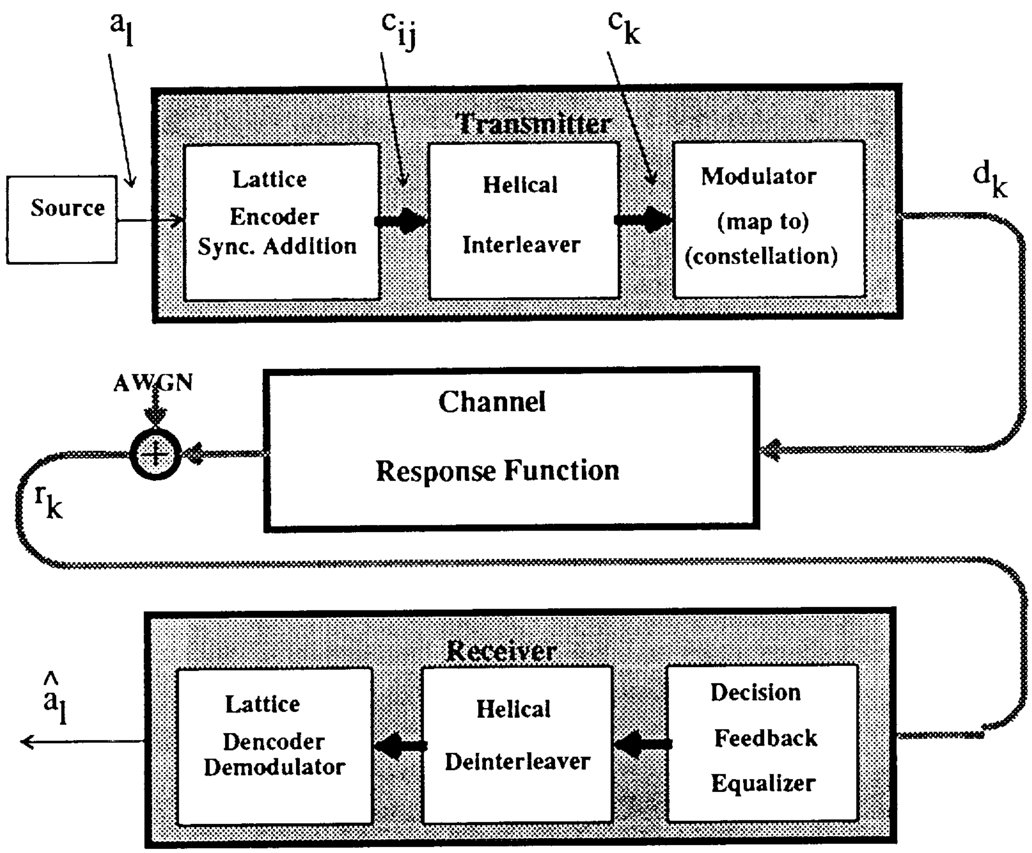


Figure 5.1 Block Diagram for the system with simple cascade of Equalizer Deinterleaver and Decoder.

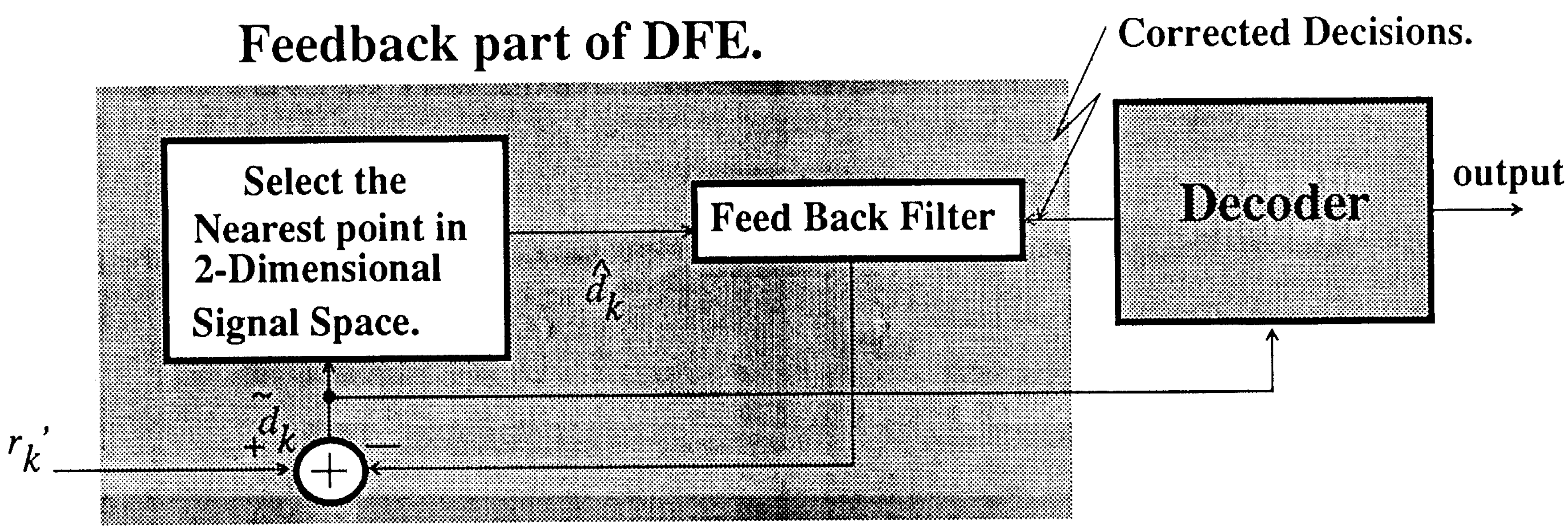


Figure 5.2 An attempt to provide corrected Decisions to the Feedback filter

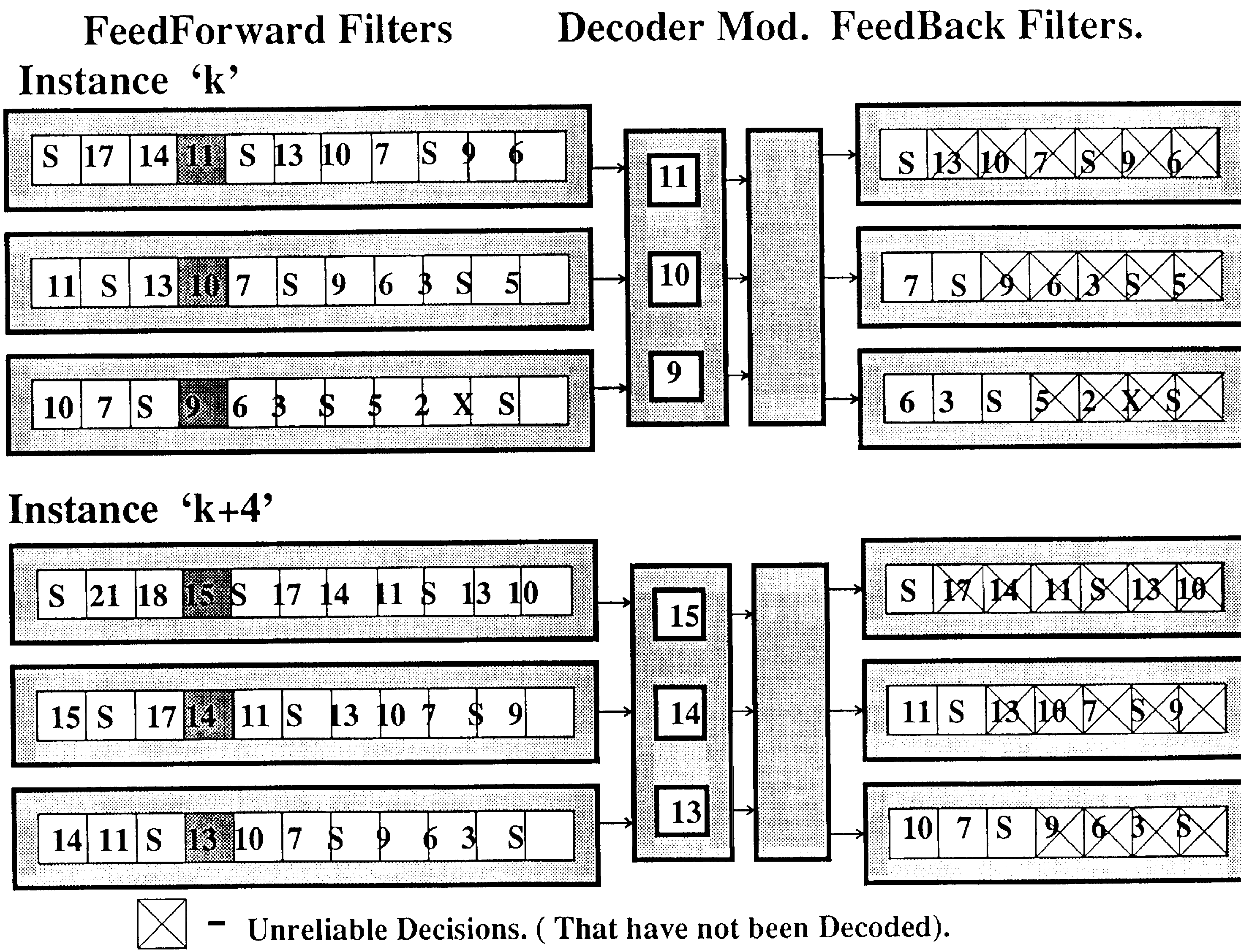


Figure 5.3Snap shot of the DFE Array at two consecutive time intervals

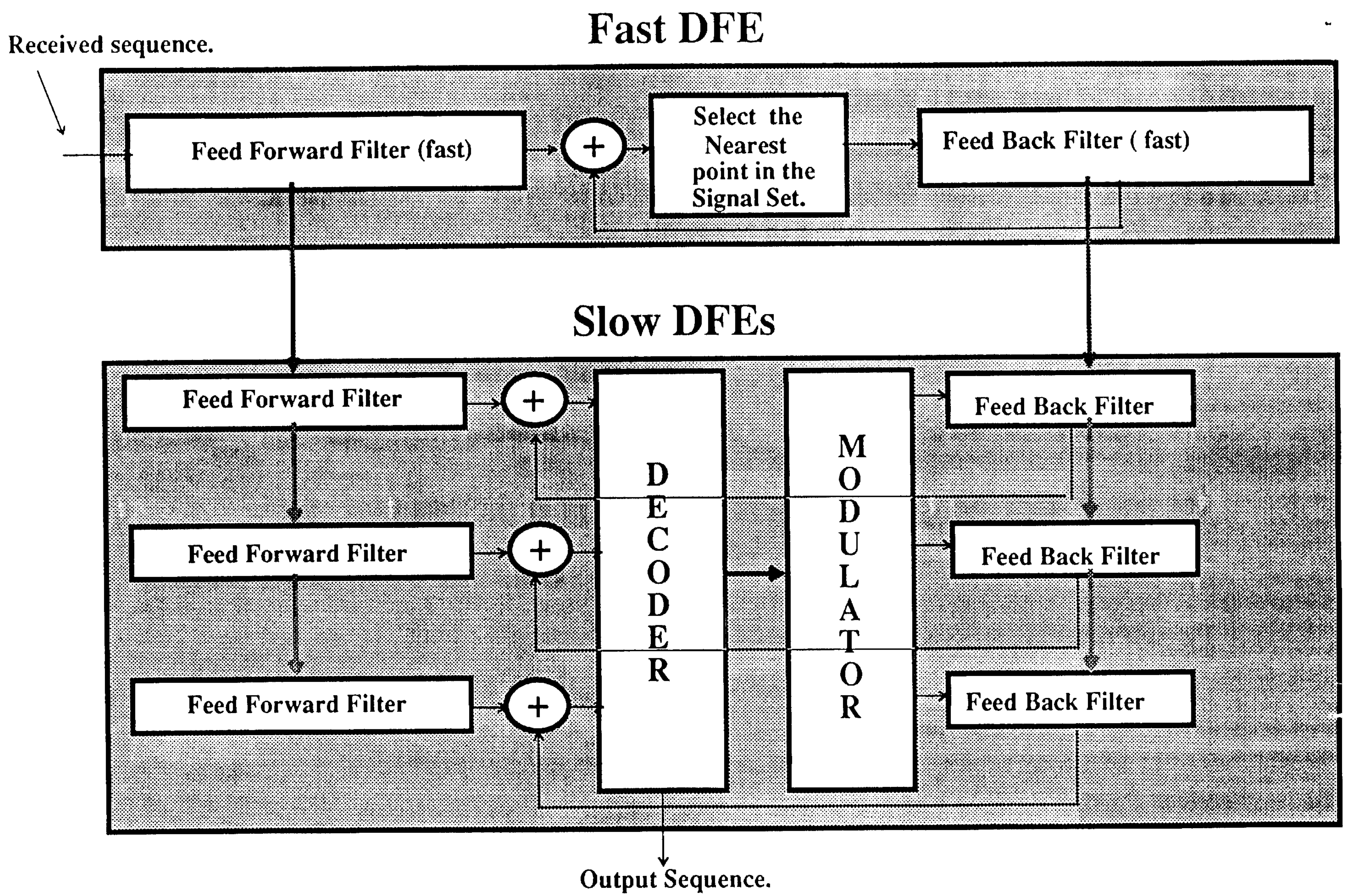


Figure 5.4 Schematic Diagram of the Codesigned Receiver

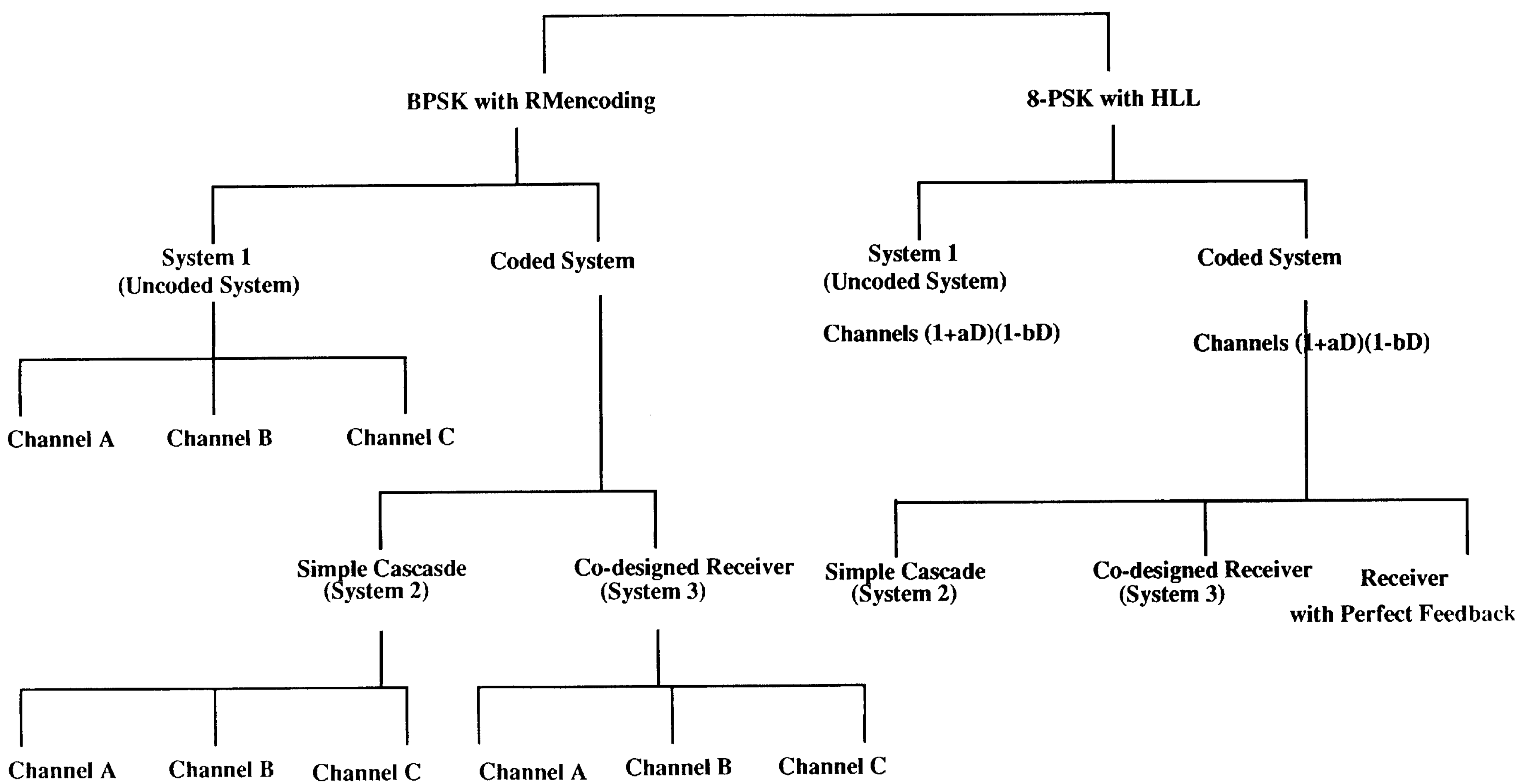
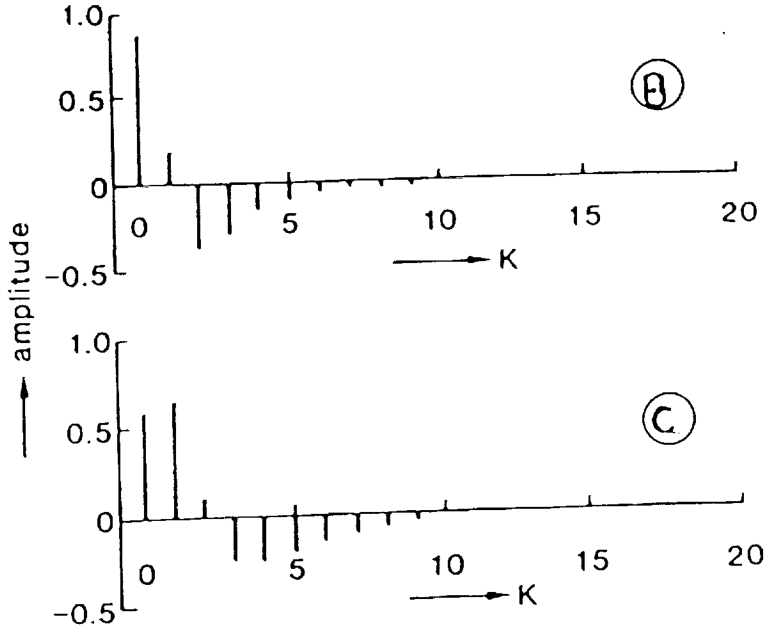
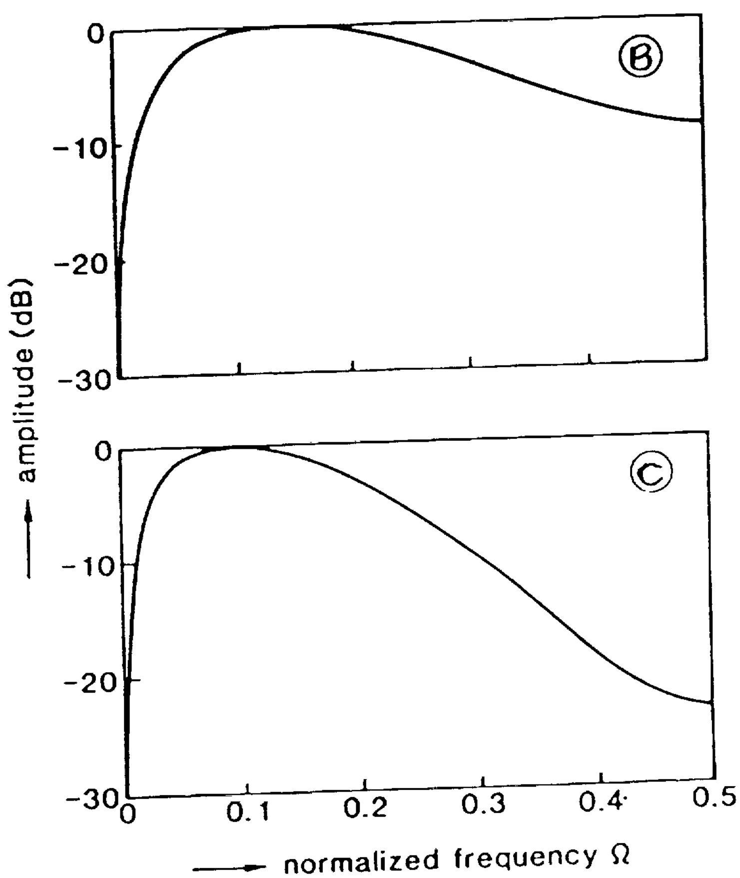


Figure 6.1 Classification of Systems used for Simulations.



(a)
Fig. 6.2 Impulse responses



(b)
Fig. 6.2 Amplitude-frequency characteristics

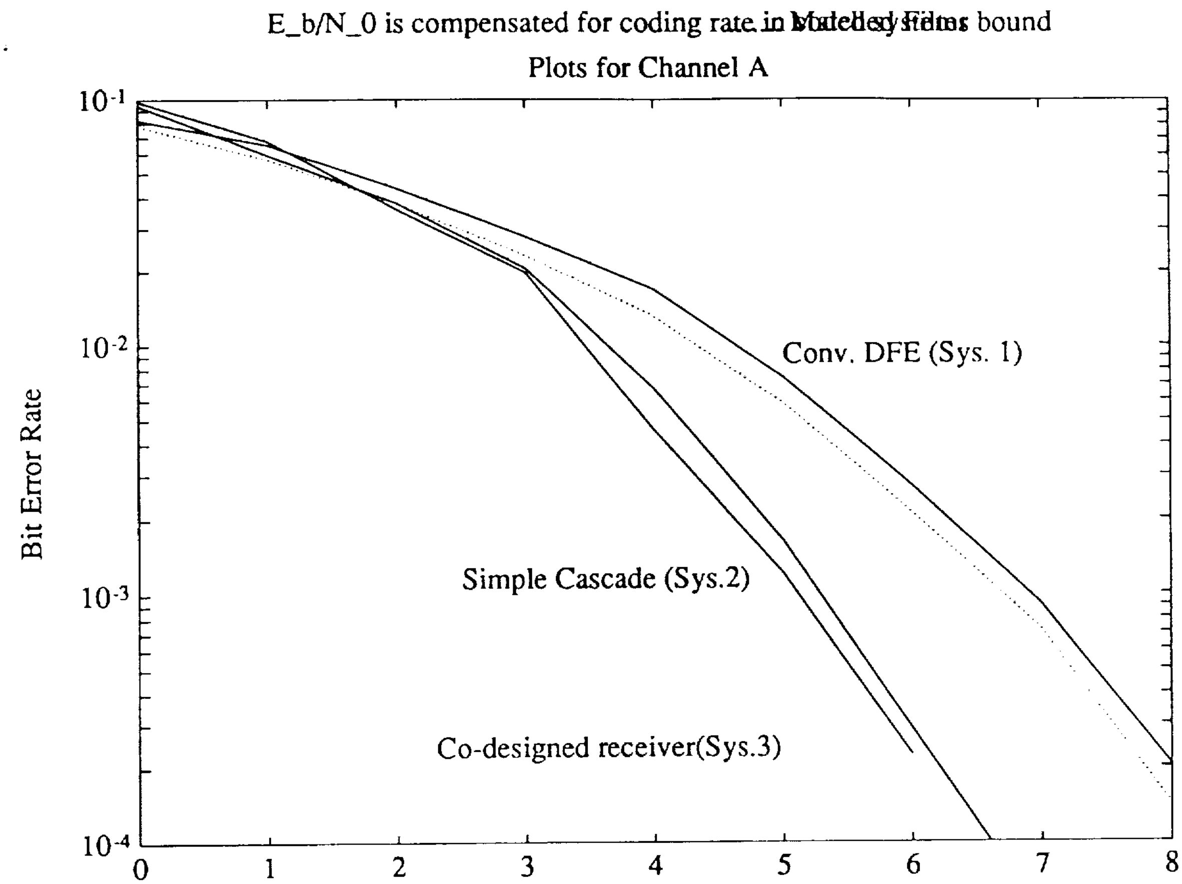


Figure 6.3: E_b/N_0 vs Pr(E)

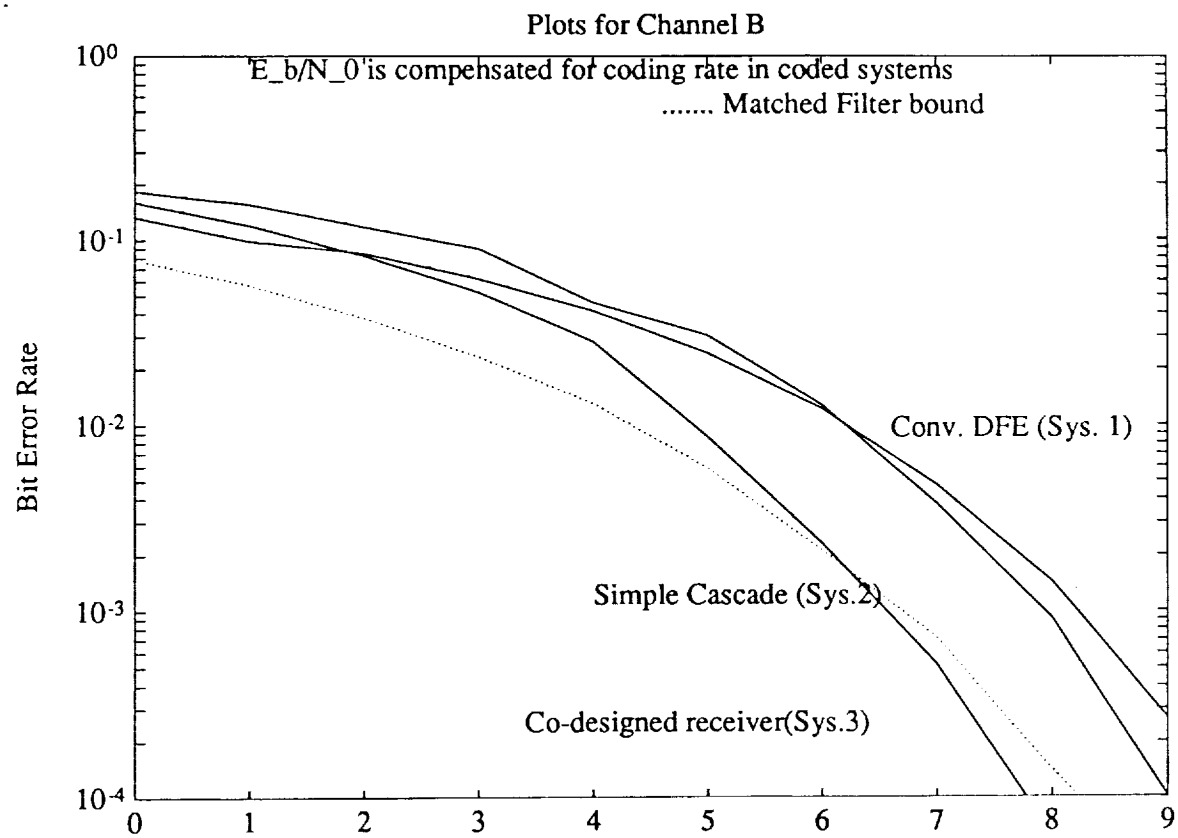


Figure 6.4: E_b/N_0 vs Pr(E)

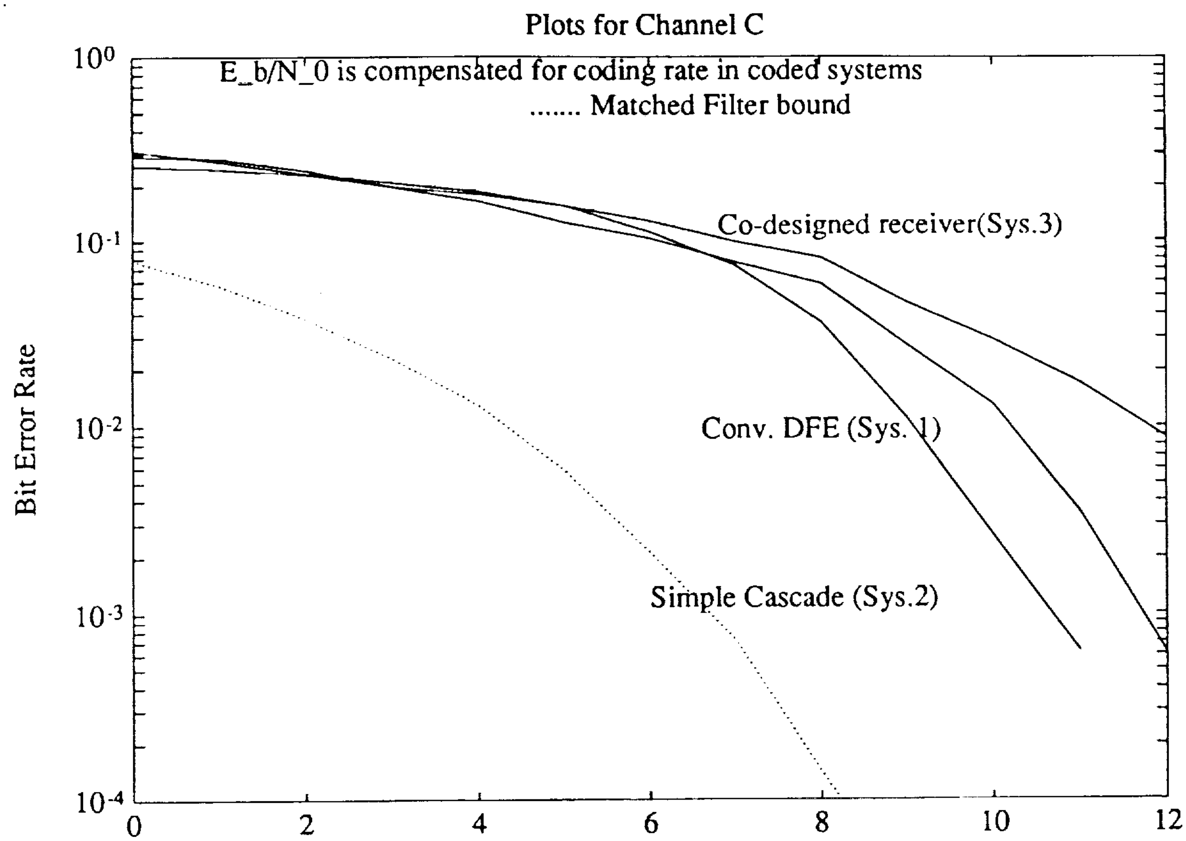


Figure 6.5: E_b/N_0 vs Pr(E)

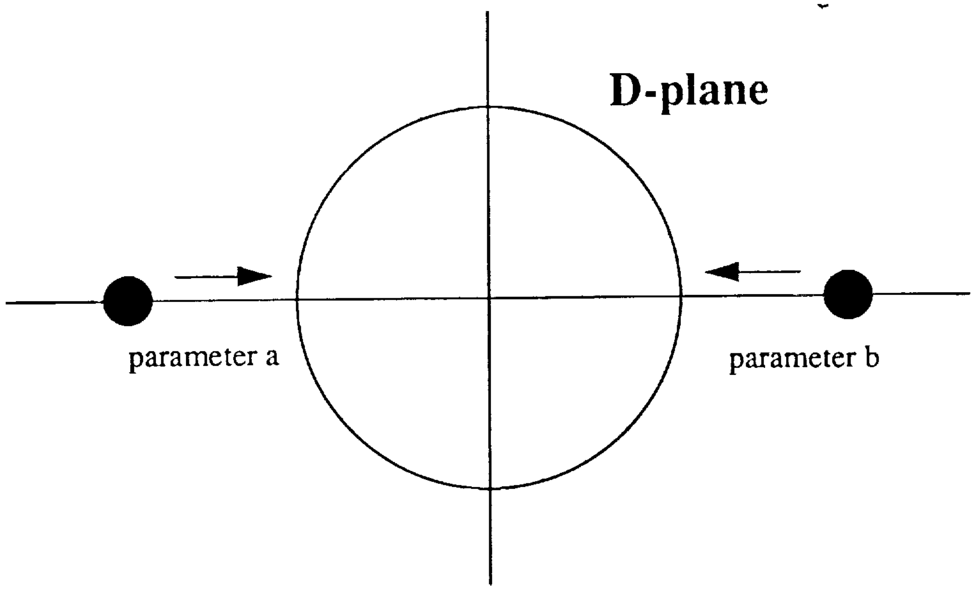
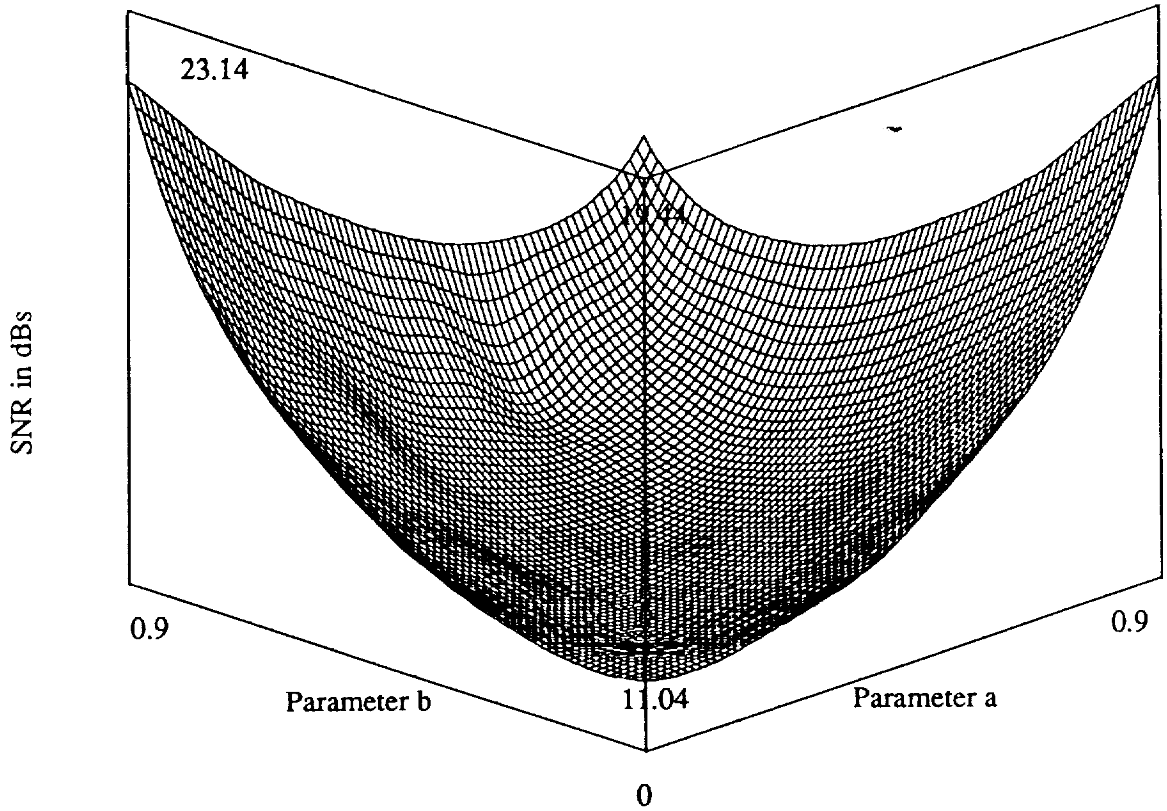


Figure 6.6: The D-plot for the channel (1+aD)(1-bD).

Performance of Uncoded System 1



Probability of Error = .001

Corresponding Contour plot

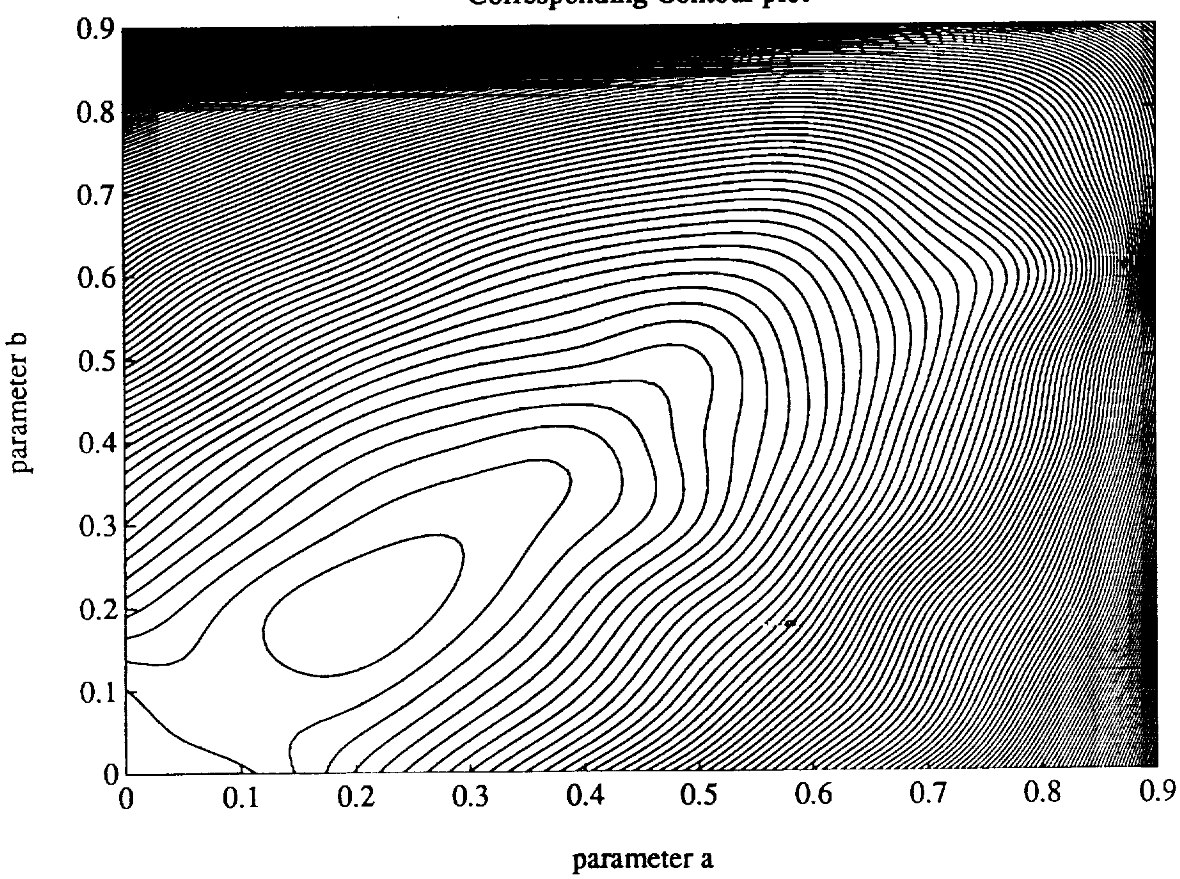
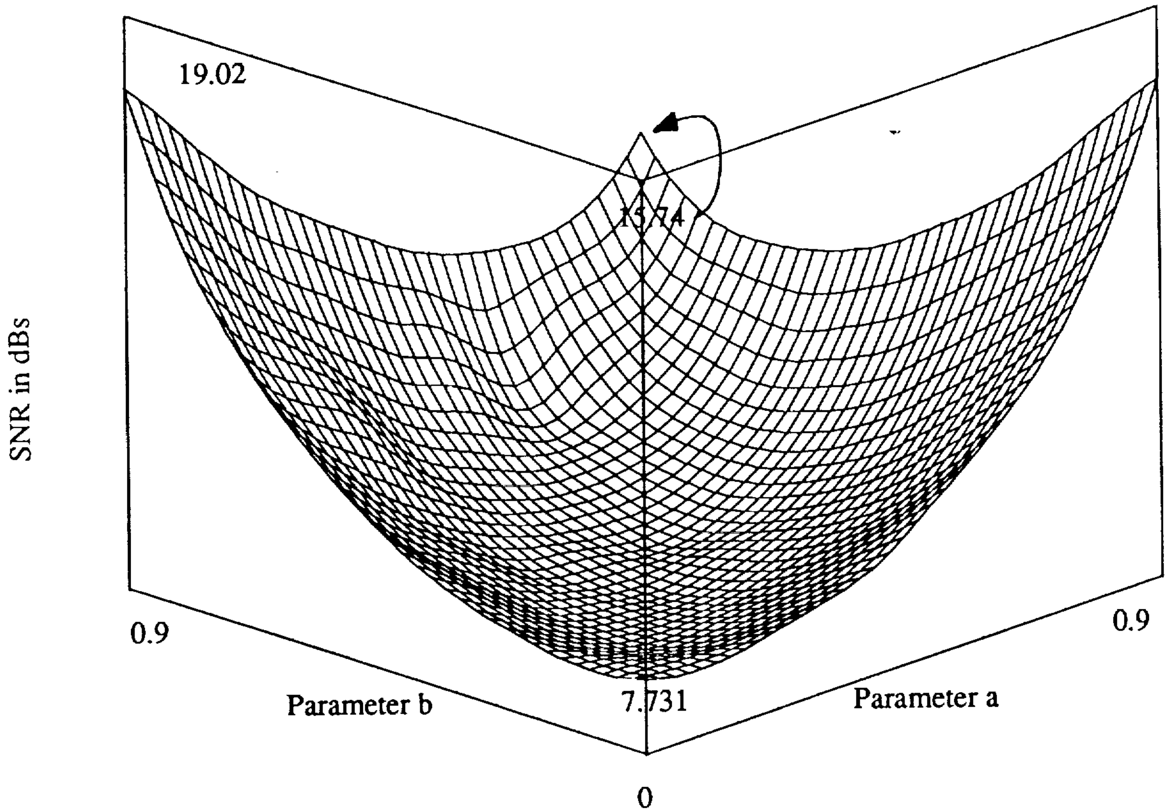


FIGURE 6.7

Performance of Coded System 2: with simple cascaded receiver



Probability of Error = .001

Corresponding Contour plot

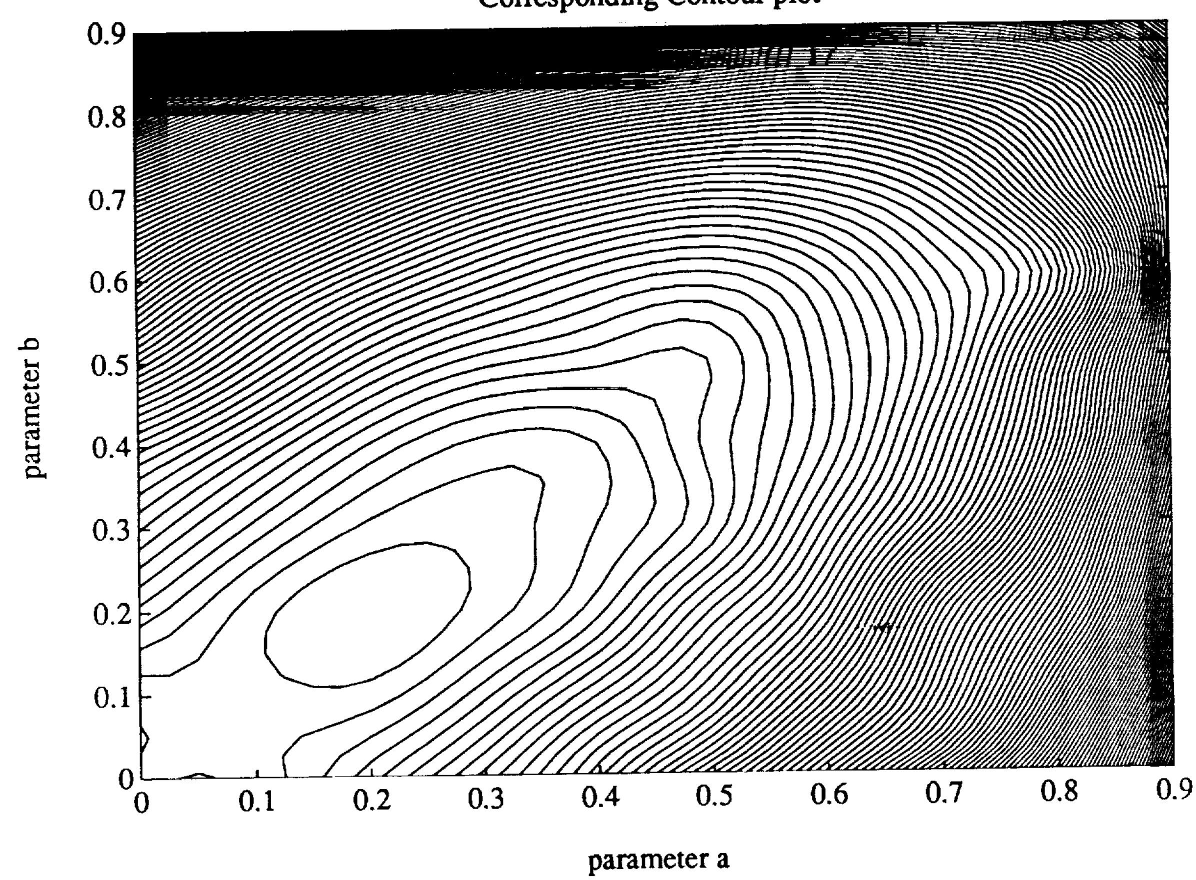
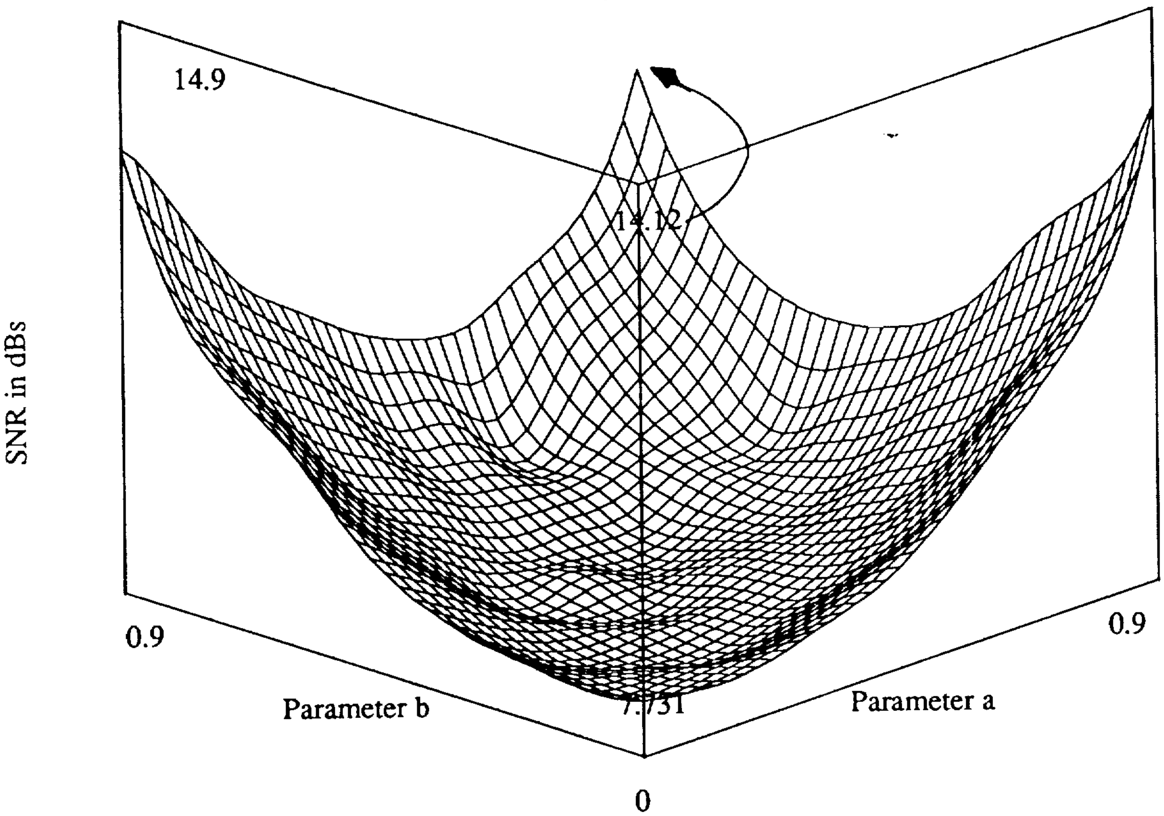


FIGURE 6.8

Performance of Coded System 3: Co-designed receiver



Probability of Error = .001

Corresponding Contour plot

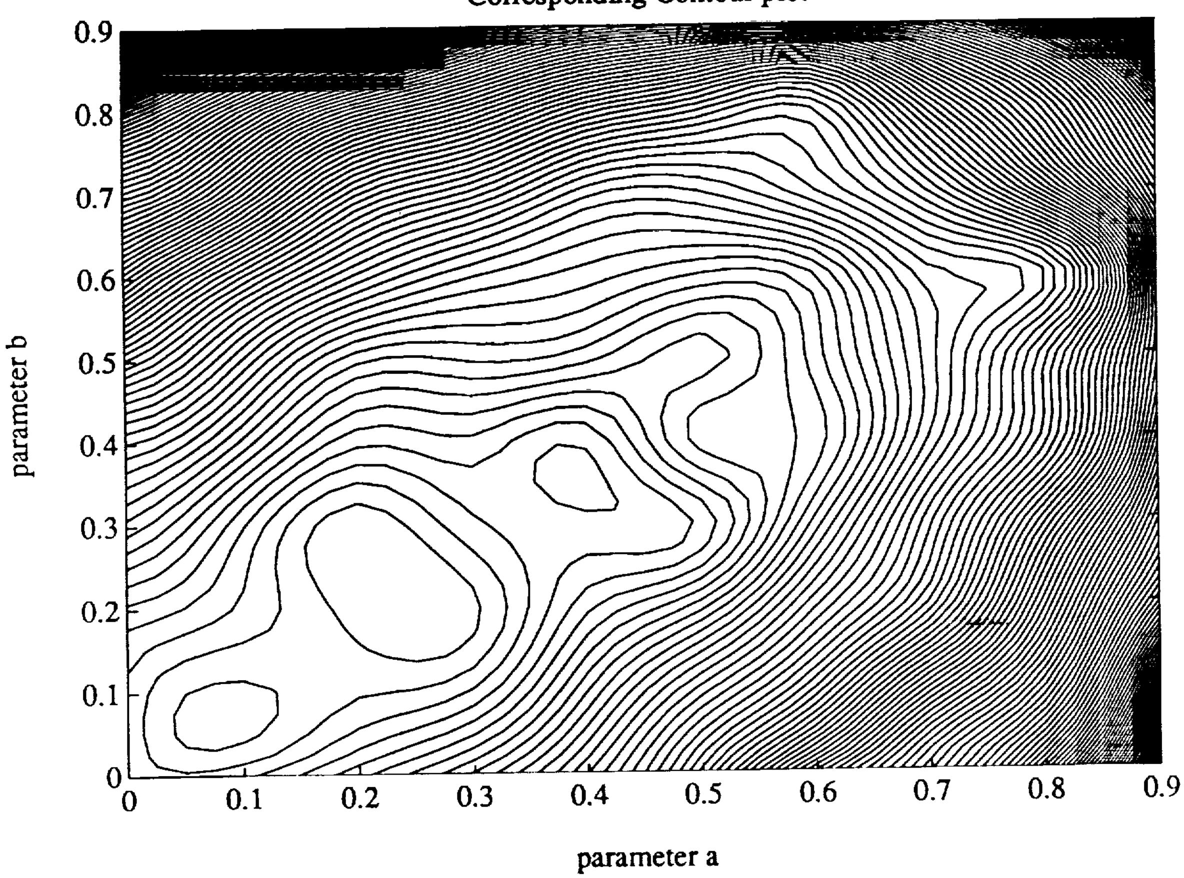


FIGURE 6.9

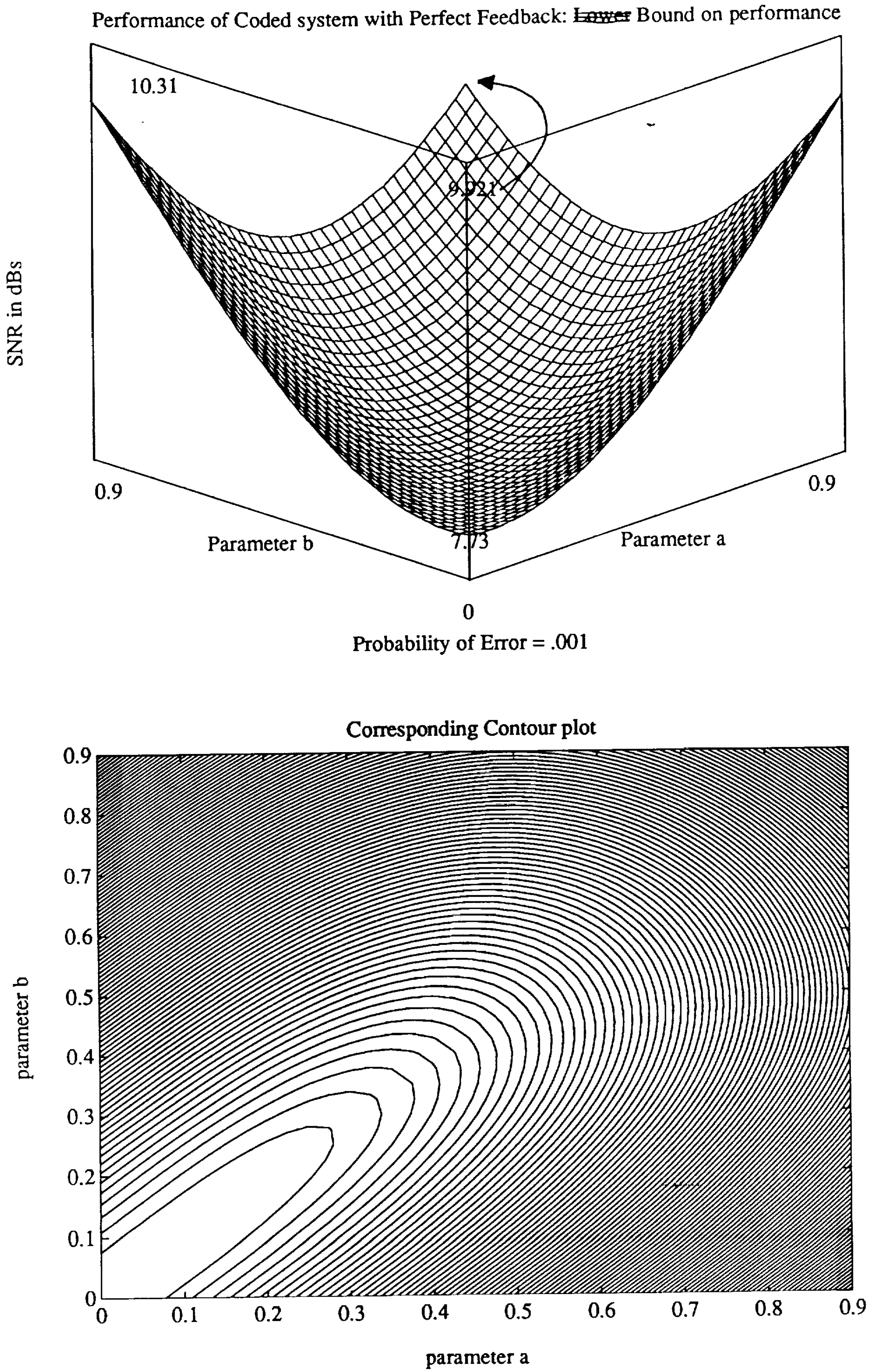
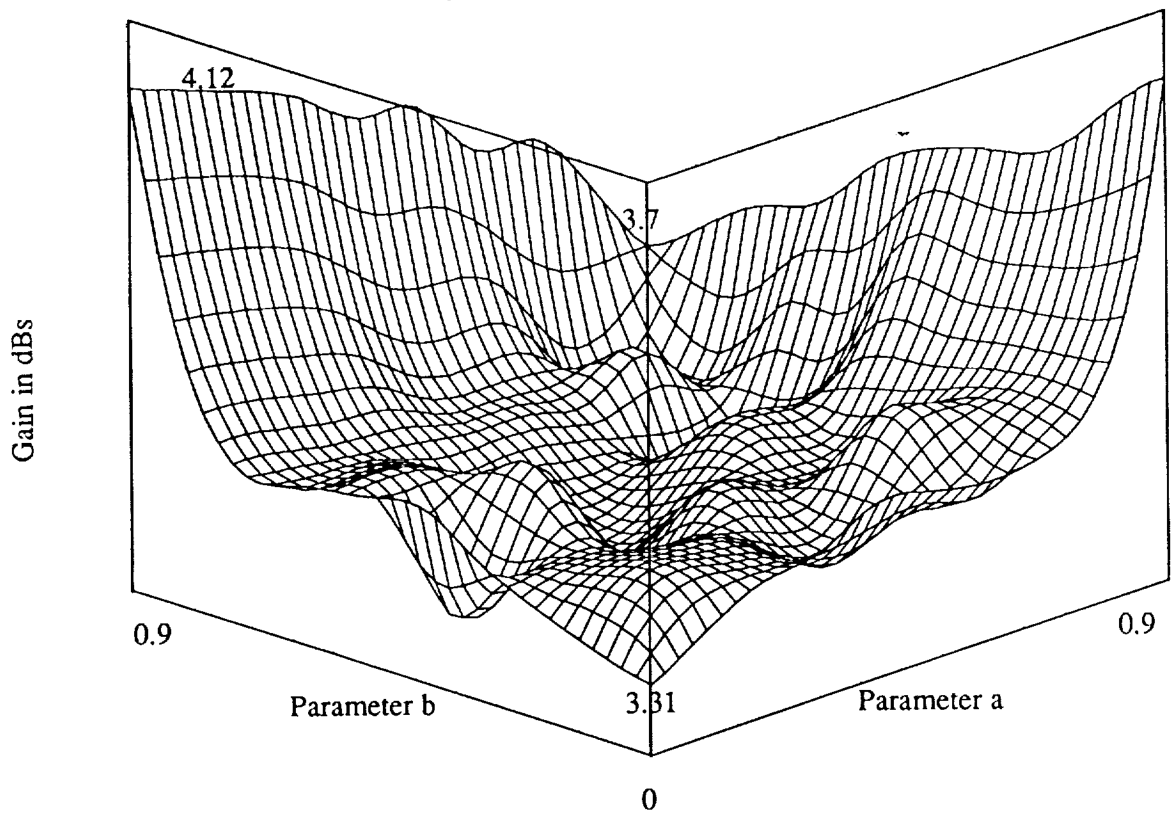

Performance of Coded system with Perfect Feedback: ~~Lower~~ Bound on performance
10.31
9.921
SNR in dBs
0.9
Parameter b
7.73
Parameter a
0.9
0
Probability of Error = .001
Corresponding Contour plot
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
parameter b
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
parameter a

FIGURE 6.10

Gain of System 2 relative to System 1: Coding Gain



Probability of Error = .001

Corresponding Contour plot

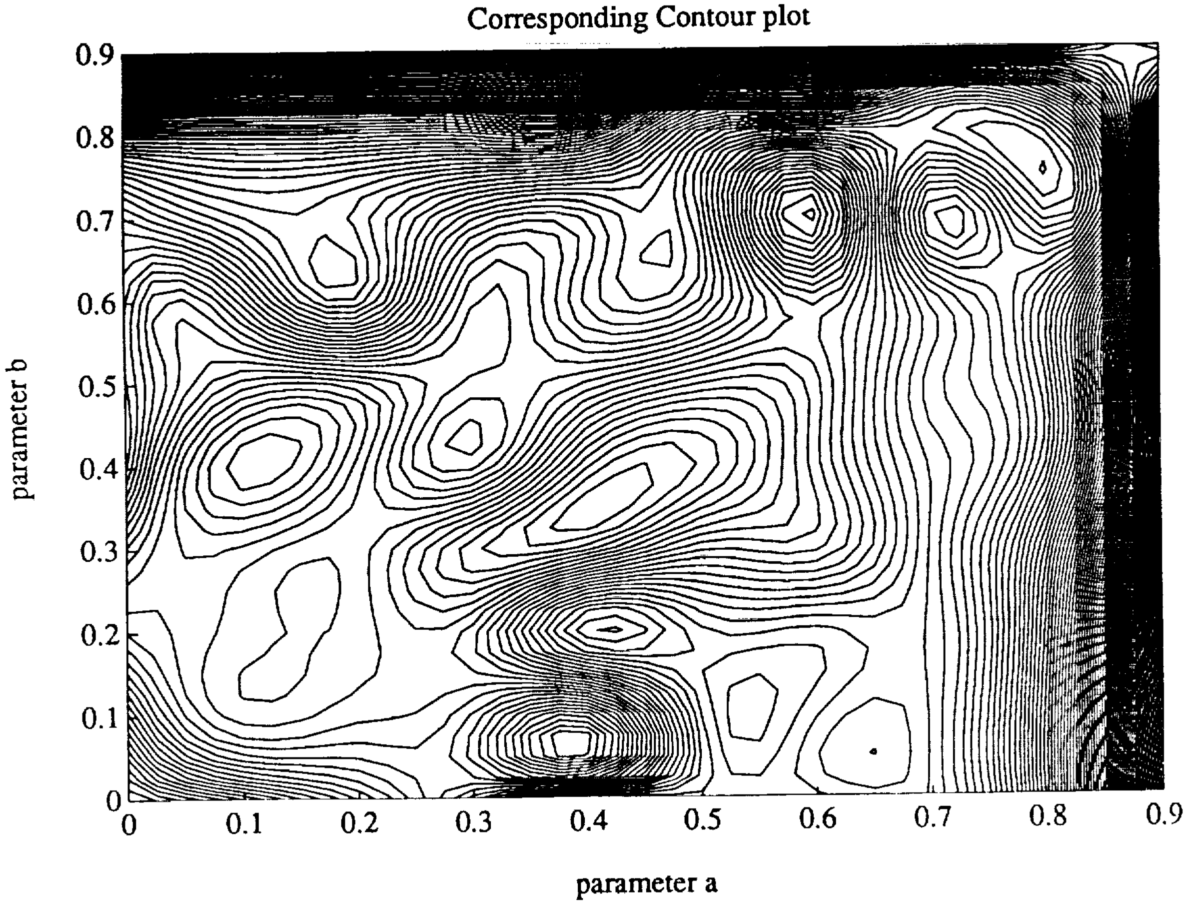
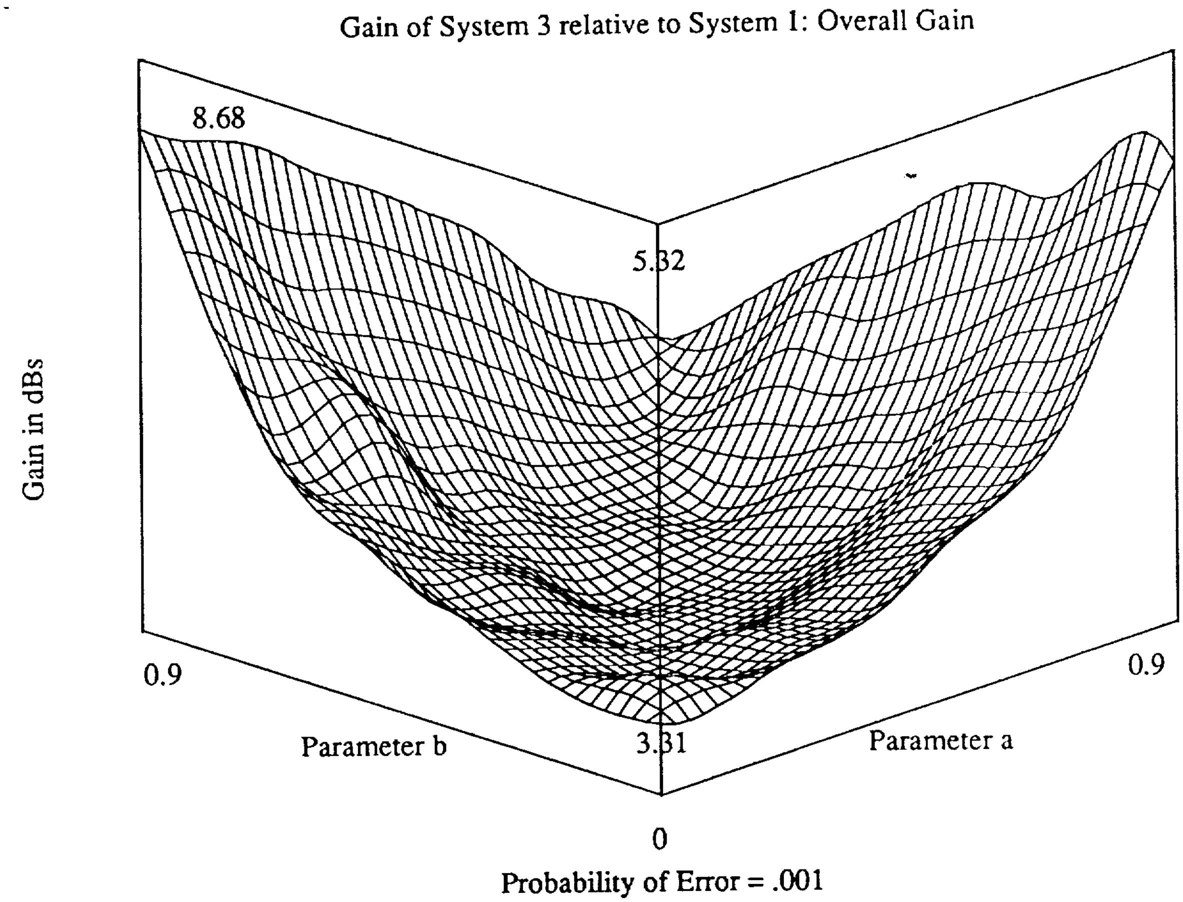


FIGURE 6.11


Gain of System 3 relative to System 1: Overall Gain
8.68
5.32
Gain in dBs
0.9
Parameter b
3.31
Parameter a
0.9
0
Probability of Error = .001

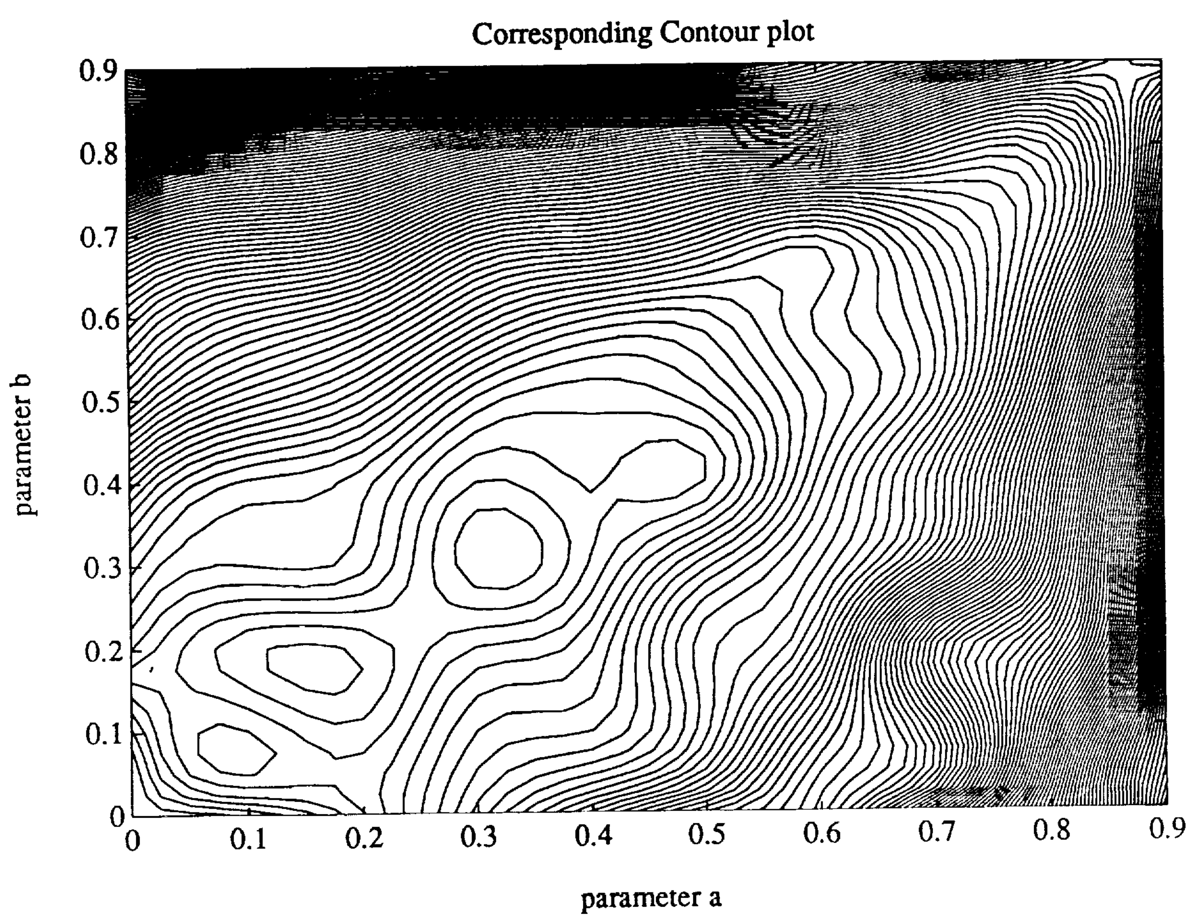
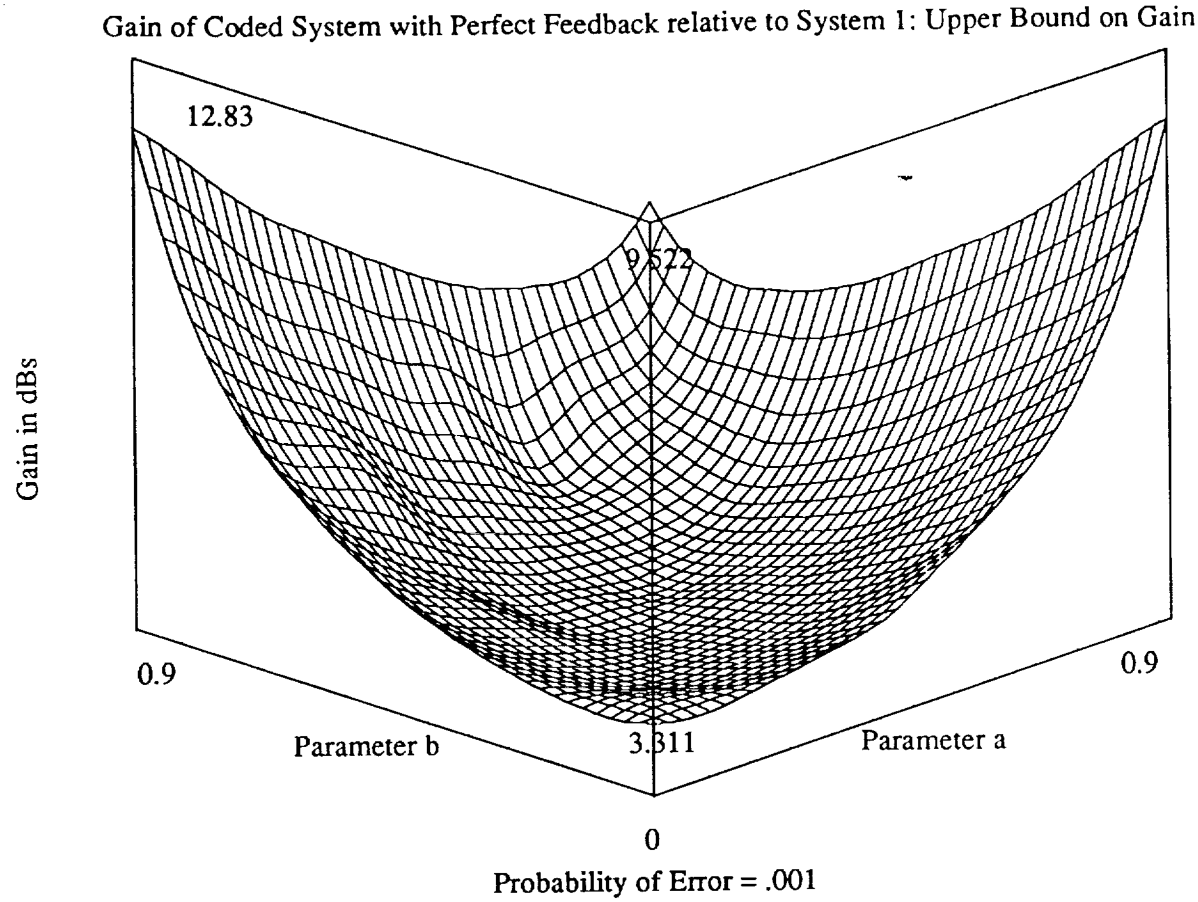
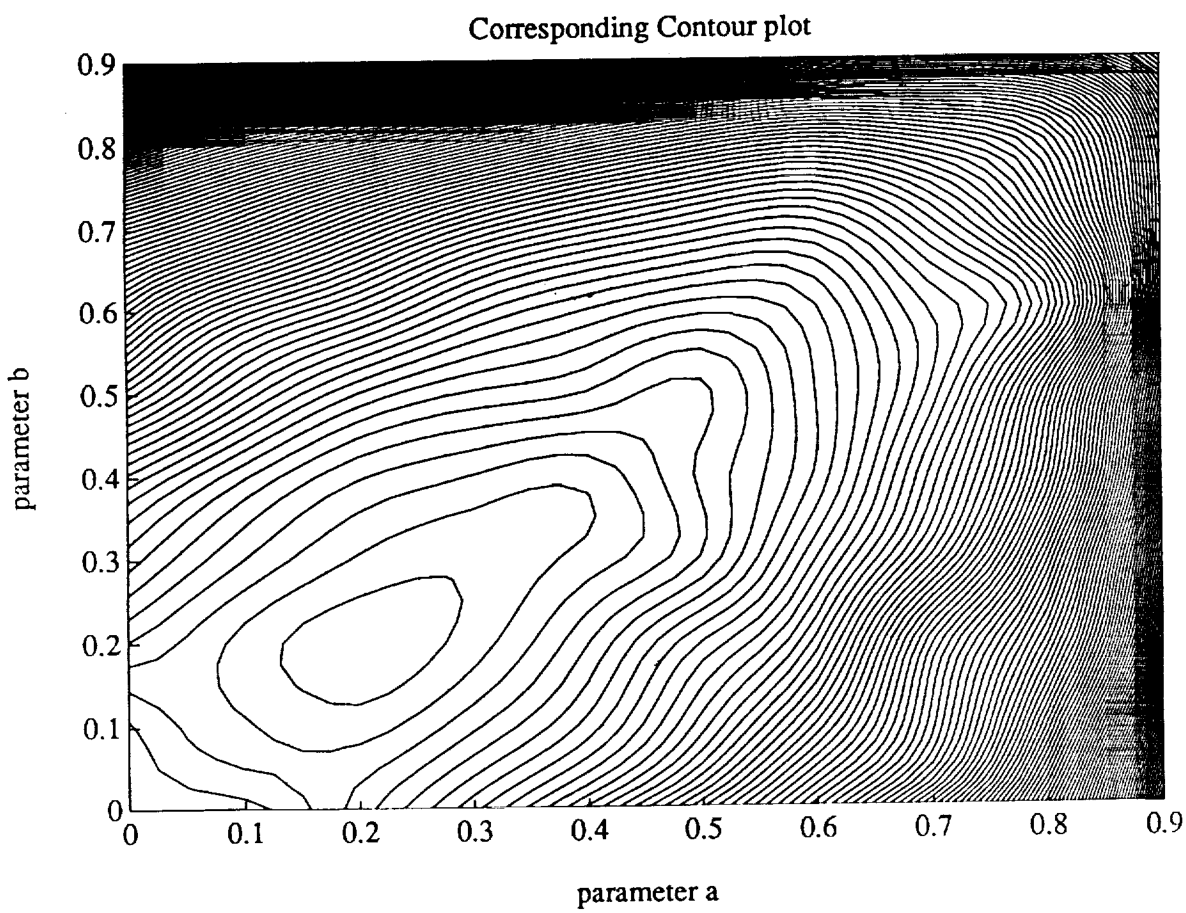

Corresponding Contour plot
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
parameter b
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
parameter a

FIGURE 6.12



Gain of Coded System with Perfect Feedback relative to System 1: Upper Bound on Gain
12.83
9.522
Gain in dBs
0.9
Parameter b
3.311
Parameter a
0.9
0
Probability of Error = .001



Corresponding Contour plot
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
parameter b
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
parameter a

FIGURE 6.13

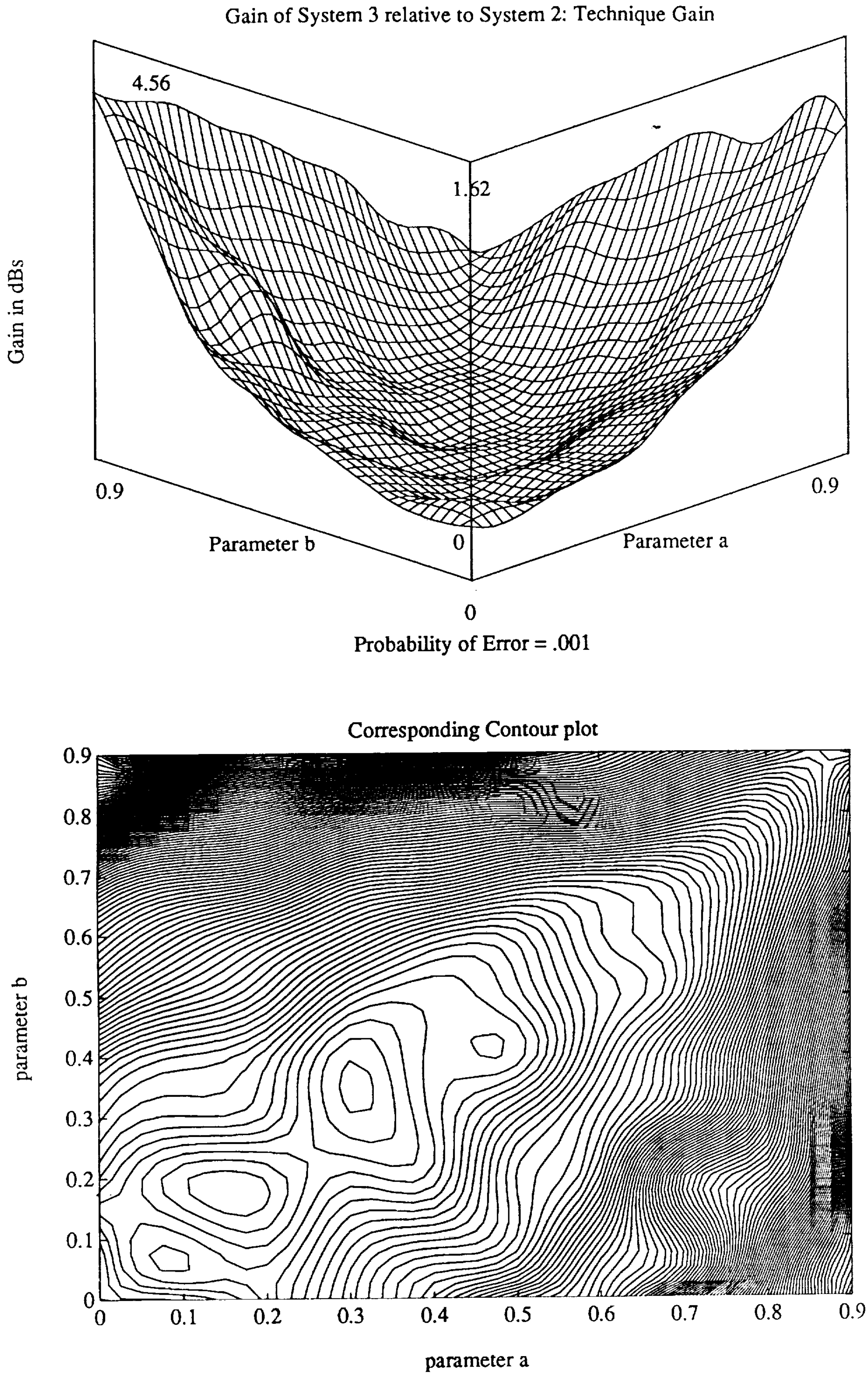
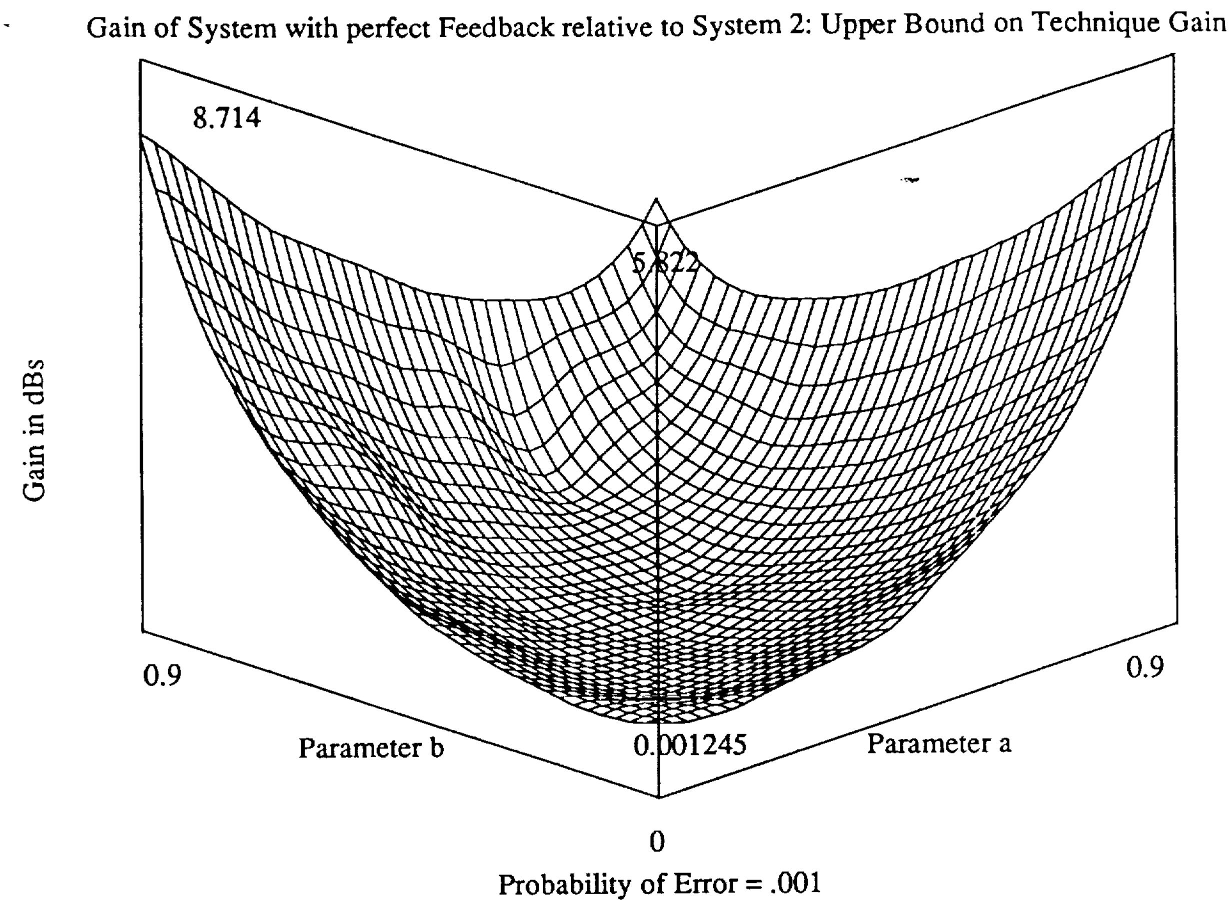
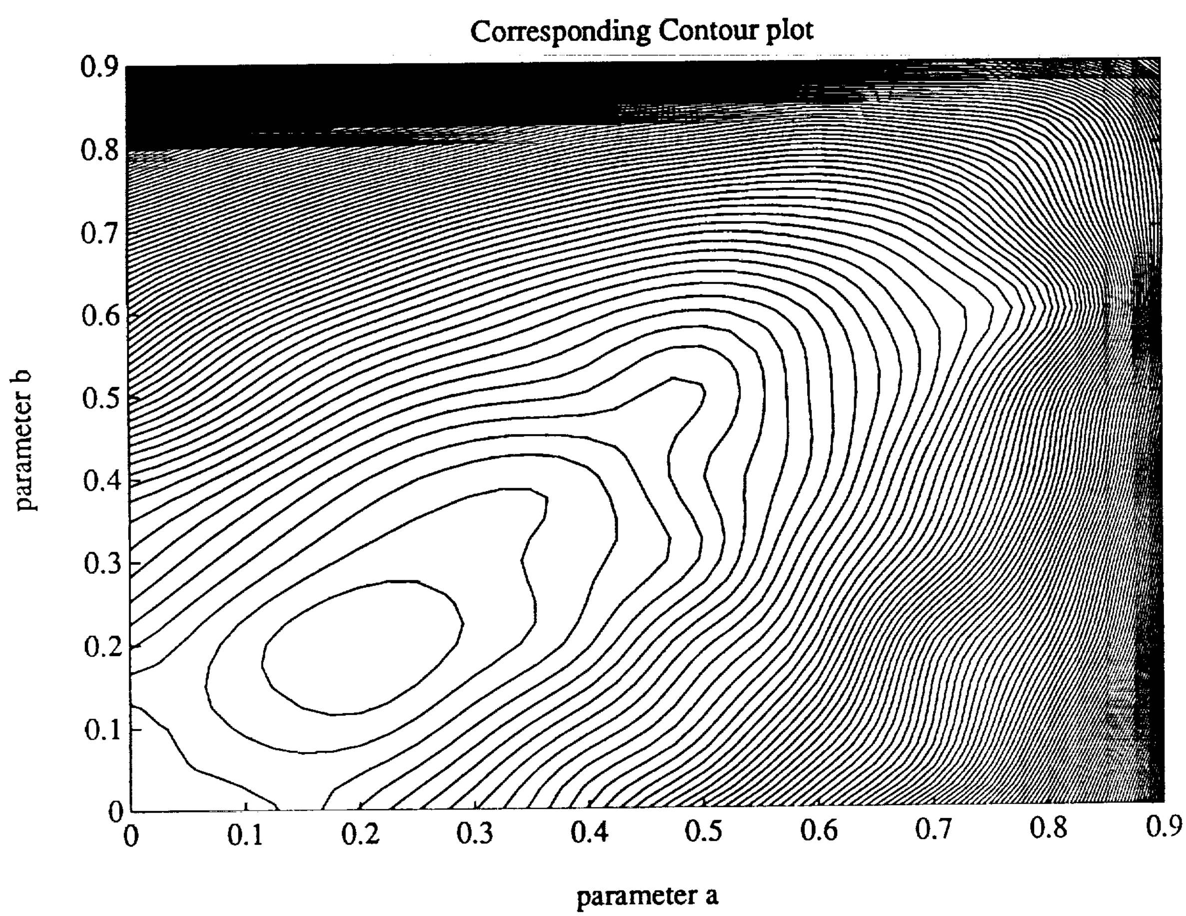

Gain of System 3 relative to System 2: Technique Gain
4.56
1.62
Gain in dBs
0.9
Parameter b
0
0.9
Parameter a
0
Probability of Error = .001
Corresponding Contour plot
parameter b
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
parameter a

FIGURE 6.14.



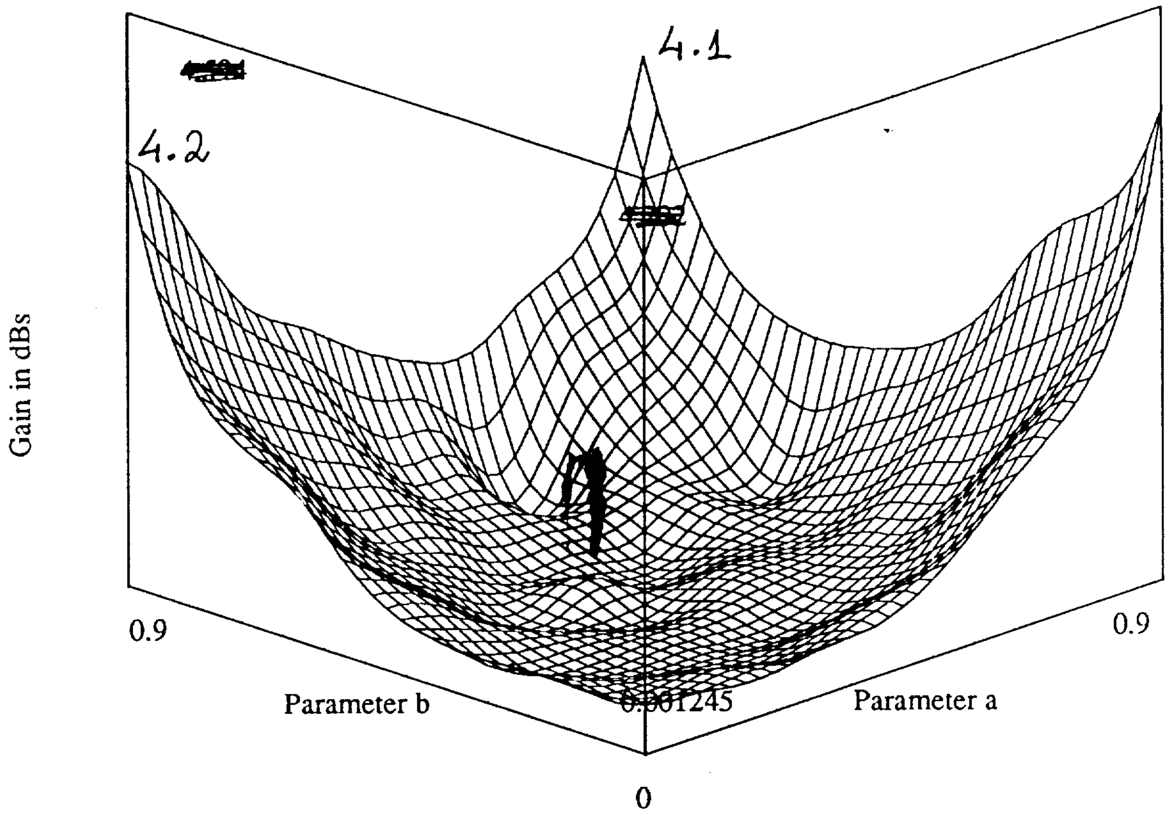
Gain of System with perfect Feedback relative to System 2: Upper Bound on Technique Gain
8.714
5.822
Gain in dBs
0.9
0.9
Parameter b
0.001245
Parameter a
0
Probability of Error = .001



Corresponding Contour plot
parameter b
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
parameter a

FIGURE 6.15

Gain of System with perfect Feedback relative to System 3



Probability of Error = .001

Corresponding Contour plot

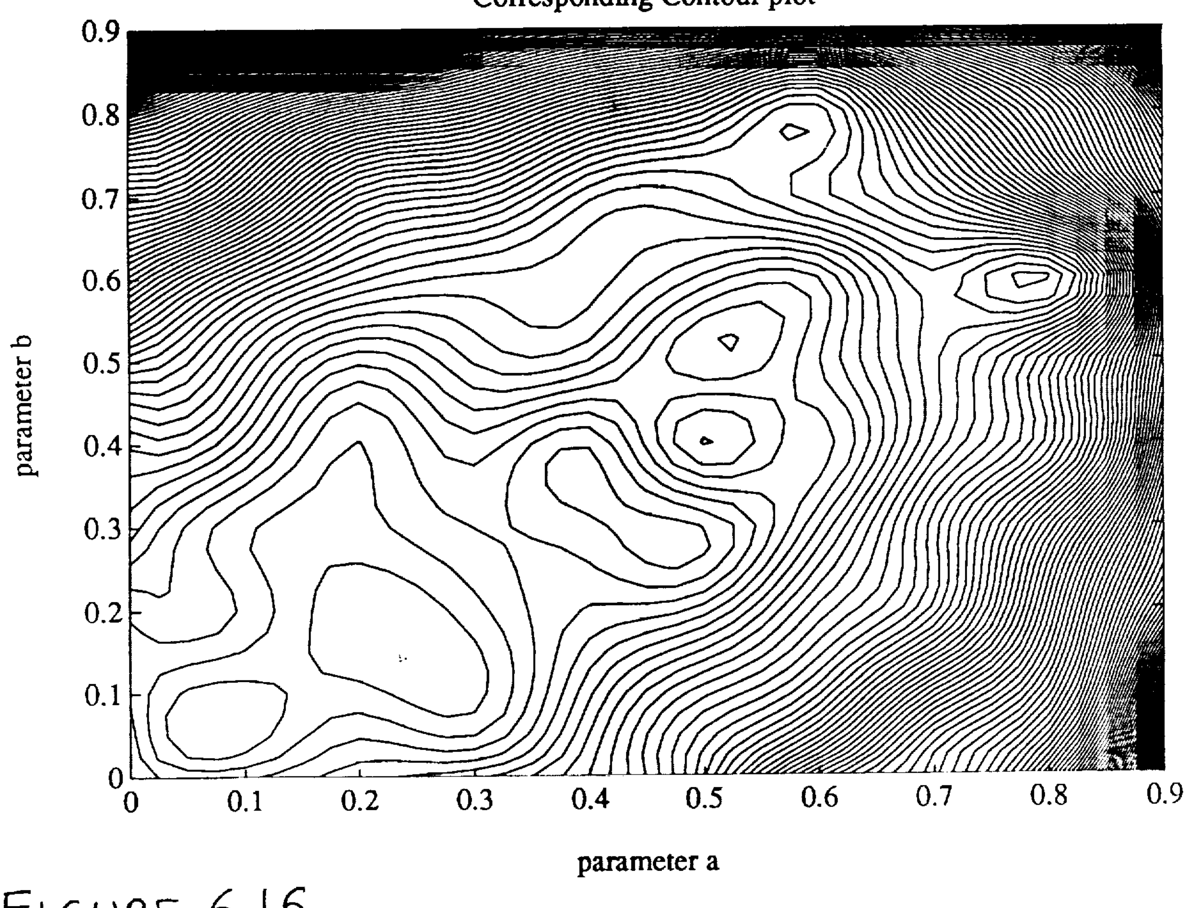


FIGURE 6.16