N92-22009

A CO-DESIGNED EQUALIZATION, MODULATION AND CODING SCHEME[1]

Robert E. Peile
University of Southern California
Los Angeles, CA 90089-0272

1) Outline

This article reports on work being performed under a Nasa-Lewis Research grant (No. NAG3-982) into a scheme that attempts to combine the advantages of certain coding, modulation and equalization techniques in order to remedy well-known disadvantages with the individual techniques. The work should be seen in the context of much of the recent research and progress made in the bandlimited channel over the last ten years. The commercial impact and technical success of Trellis Coded Modulation seems to illustrate that, if Shannon's capacity is going to be neared, the modulation and coding of an analogue signal ought to be viewed as an integrated process. More recent work, including this project, has tried to press beyond the gains obtained for Average White Gaussian Noise and combine the coding/modulation in with adaptive equalization. The motive is to gain similar advances on less perfect or idealized channels.

This work has split into several different areas. One important area is for a single point-to-point communications link for which it is assumed that the channel changes slow enough for the transmitter to be aware of the channel response function. Under these conditions, the transmitter can apply "pre-equalization" and pre-filtering to the signal prior to transmission. It seems clear that, under these conditions, the least complexity solution to this problem is not the traditional solution of placing all the processing at the receiver [refs. 1-6]. Another important area, that of this paper, looks at the case of multipoint transmission over variable quality, diverse channels, i.e. when the transmitter can not simultaneously pre-process a signal to be optimal at the input to each and every modem. In fact, both areas are encouraged by results due to Price [ref. 7]. This proved that, under high-power conditions, an "ideal" Decision-Feedback Equalizer followed by a code designed for the AWGN channel could achieve capacity. This result raises two immediate questions:

i) What is the difference between "ideal" and "non-ideal" Decision-Feedback Equalizers?

ii) How can the performance of "ideal" Decision-Feedback Equalizers be better approximated?

Section 2 reviews the definition of a DFE and the cost of non-ideal performance. In Section 3, we describe helical interleaving and a property of this scheme that basic to our system. In Section 4, we describe the coding and modulation schemes selected for the system. The target base constellations are 32 point PSK and 256 point QAM and the codes are derived from the Barnes-Wall lattice [refs. 14,18]. In Section 5, we pull the ingredients together and describe the entire system. It must be stressed that that the entire system is seen as more than the sum of the parts; the components reinforce each other. Section 6 discusses the results of simulations for simpler models. Section 7 presents conclusions.

2) Decision-Feedback Equalizers

The system to be described revolves around an improved used of coding in systems using a Decision Feedback Equalizer(DFE). In these equalizers, a channel baud is detected via a two-fold process:

1 [1] This work was supported under NASA-Lewis Research Grant No. NAG3-982.

i) by applying a transversal filter to channel samples taken at times up to and including the time of the present baud.

ii) by applying a transversal filter to the *decisions* as to the previously transmitted bauds.

The first process is known as the feedforward section and the latter as the feedback section. Figure 1 shows such an equalizer. For both sections the taps to the transversal filter are adjusted so that the combined effect of the system is to remove the effect of the channel transfer function and to produce decisions as to the transmitted bauds. For reasonably stationary channels the taps are adjusted (or "trained") and then left fixed. For more time-varying channels, various algorithms exist for continuous adjustment, ranging from the low complexity LMS algorithm to the more complex Kalman algorithm. The LMS algorithm is suitable for slowly varying channels (actually channels for which the eigenvalue spread of the covariance matrix is small) and the Kalman algorithm is suited for more adverse channels. Hybrid schemes have also been used. The importance of the feedback section has long been understood in relation to several classes of channels. It is of most value when the channel destroys information as well as distorts the signal, for example when a deep spectral null appears on the channel. The decision feedback section, being non-linear, is capable of re-introducing some amount of this lost information that a linear system, operating on the received signal, can not. Figures 2, 3 and 4 (taken from [ref. 8]) show some performance results of a DFE versus a linear equalizer that employs feedforward alone. Figure 2 shows the channel response of three channels. Figure 3 shows the performance of equalizers without decision feedback ("linear feedback" means that a section contains the unquantized outputs). Figure 4 shows the performance of a DFE on the two worst of the three channels. Clearly there are some channels for which the non-linear decision feedback section is critical for communication.

One recognized drawback with DFEs is that of error propagation; when the DFE does produce an error in the detection of the transmitted baud, this error is fed back through the feedback section, providing an erroneous input to several successive symbols and possibly producing further errors in the detected bauds. This can cause performance to be degraded relative to that of correct symbol feedback. In Figure 4, performance curves show the effect of using detected data rather than correct data can result in degradation of at least 2dBs. This is the issue that the proposed system attempts to address. Other authors have addressed this question [refs. 4,5,9] from different perspectives. One might consider solving this problem by adding error correction between the symbol detection and the feedback register (point A in Figure 1). 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 corrected data is available for input into the feedback section or the code selection is limited to codes of short length and, consequently, limited performance. Fortunately, helical interleaving can remove this dilemma. It is easier to describe helical interleaving as stand-alone technique prior to adapting it to the equalization problem. This is done in Section 3.

3) Helical Interleaving

In this system the above limitation is removed by using the time scrambling inherent in all interleaving and, in particular, a property of a type of interleaving known as helical interleaving, see Figure 5. This interleaving was invented by Tong and Berlekamp of Cyclotomics. It was used by them for forecasting correlated digital error bursts to a Reed-Solomon decoder [ref. 10]. The salient feature of helical interleaving is that every symbol of a codeword is immediately preceded by a symbol that is either in an earlier codeword or is a synchronization symbol. It remains to describe the interleaving. We will describe it by example.

Consider interleaving a code with a block length of eight symbols. To each codeword we attach another synchronization symbol, giving a total block length of nine symbols. Figure 5a shows several codewords with the attached symbol S. The numbering reflects the time ordering of the symbols leaving the encoder and prior to the action of the interleaver. (Note that the sync. character represents the symbol numbers divisible by nine). At the interleaver, the codewords are written down the columns of an array as shown in Figure 5b. When complete, the rows of the array are read out and transmitted. Figure 5c gives the transmit sequence of a few rows. Note that consecutive symbols of a codeword are separated by precisely 8 symbols. At the receiver the process is reversed, i.e., the rows are filled in with the received data and complete columns ending with 'S' are read out. In fact, the entire interleaver and de-interleaver process can be made symmetric.

A symbol c received from the channel is said to "have a prediction" if the proceeding channel symbol is either:

(i) a symbol from a codeword that is decoded earlier than the codeword to which c belongs;

(ii) a synchronization symbol.

Inspection of Figures 5b and 5c exposes a major advantage of Helical interleavings; every symbol has a prediction. For channels that have heavily correlated error statistics, this is ideal; if a symbol is in error, the following symbol is highly likely to also be in error. If a symbol is known to be in error, the following symbol can be treated as an erasure (for more detail see [ref. 11].) However, we will be adapting this property for a different purpose in Section 5.

4) The Barnes-Wall Lattice and System Coding

4.1 Introduction, encoding and synchronization

An overview of coding for the bandlimited channel is outside of the scope of this paper. A selective list of references are given [refs. 12-18]. There are, in essence, three basic elements that have to be examined prior to the addition of any coding. These are:

i) An infinite set of lattice points in the complex plane. We think of these as containing the coded signaling constellation if we were not limited to a peak power constraint.

ii) A finite set of these lattice points that lay within the region that we can signal. For example, on a non-linear channel, we have m regularly spaced points on a unit circle giving m-ary PSK. On a linear channel, we can select a finite set of m points placed on or within a circle of radius r. For example, Figure 6 shows 64-QAM, a set of 64 points arranged in a grid. The grid is obviously a subset of an infinite grid or checkerboard.

iii) A labeling of the finite set of points that reflects the squared Euclidean distance between pairs of points.

The exact way in which a finite subset of points is "punched out" of an infinite grid varies from case-to-case and can lead to greater or lesser performance gains. This parameter is often called "the shape gain". Since the principles of the present system do not relie upon the exact number of points or the exact shape, we have not optimized this aspect; this can, to a large extent, be done at a later date. This project is targeting two constellations. The first is 32-ary PSK. The second is 256 point QAM. For the QAM case, we have concentrated upon a *square* grid of 256 points (8 channel bits per modulation symbol). A circle containing 256 points from a square grid would produce additional gain in terms of average SNR. Given a base constellation, the next essential is to give it a labeling that reflects the Euclidean distance between the points. Although there have been many generalizations, Ungerboeck's [ref. 12] labeling by set-partitioning is the simplest to understand. For PSK, the labeling is relatively intuitive, amounting to a clock order around the circumference. For QAM, the labeling is only slightly harder, [ref. 14]. For example, in Figure 6, 64 QAM points are labeled 0-63. The labeling is constructed so that the least significant bit is the most error prone, the second most least significant bit is the next most etc. To be more precise, if the minimum squared Euclidean distance between any two distinct points in the square is $d^2_{\min}$ and two points agree in the first k least significant bits, $k \geq 0$, then the minimum distance between the points is $\geq 2^k.d^2_{\min}$. Alternatively, the constellations selected by specifying the least k bits of a modulation baud have superior distance and performance properties as k increases, the gain being at least 3dBs per increment of k.

With the constellation and labeling fixed, we are ready to add coding. We are going to use a particular example derived from the Barnes-Wall lattice of complex dimension 8 and real dimension 16 and written as $BW(16)$, [refs. 14,16,18]. By definition, the Barnes-Wall lattice is a an application to a square grid constellation. However, the same coding scheme can be used for PSK, i.e. encoded integers can be used as to select PSK points using mapping by set-partitioning. Data is entered in a n by 8 array, where $n \geq 3$. The first row only has 1 data bit, the second row only has 4 data bits and the third has 7 bits. The remaining n-3 rows are fully occupied by data bits. The first row is completed by setting all the row entries equal to the data bit in that row. This is a (8,1;8) codeword. The second row is completed by taking the 4 bits in that row, encoding them into a (8,4;4) Reed-Muller codeword and placing the codeword into the second row. The third row is completed by adding an eighth bit that makes the third row entries add to an even entry, a codeword from a (8,7,2) code. An example is given in Figure 7. The columns of the array are regarded as integers. Hence the digital encoding gives a sequence of eight integers. Each of these integers selects a unique constellation point using a set-partitioning labeling. For a square grid labelled by set-partitioning we get points from $BW(16)$, the Barnes-Wall lattice [refs. 14,18]. For a PSK system, the encoded integers can still be used to select constellation points but the process does not give Barnes-Wall lattice points. In either case, a codeword can be regarded as sequence of 8 complex constellation points and every sequence of eight constellation points is not necessarily a codeword.

A $BW(16)$ lattice using n rows contains $(n\text{-}3).8 + 12 = 8n\text{-}12$ bits of data spread over 8 modulation symbols, e.g. 52 bits for $n = 8$. This allows the nominal gain of the lattice (for large n and high SNR) to be estimated and normalized for both the redundancy and dimensionality. Unfortunately, the definition of gain differs from source to source. [ref. 16, p. 74] gives the gain as 5.491dBs while Forney [ref. 14] gives the gain as $10\log_{10}(2^{1.5})$= 4.5118dBs. $BW(16)$ is the best known lattice of complex dimension 8 and there is strong evidence [ref. 16] this is the best possible lattice of this dimension. It was selected for this project as a compromise between ease of decoding and return in gain.

In this system, we add a ninth synchronization symbol to each lattice point, giving a combined block-length of nine modulation symbols per block. The n_s least most significant bits of the synchronization symbol are set to a fixed pattern and the n - n_s most significant bits are used for data communication. The value of n_s is a parameter to be optimized. Obviously, small values are better in terms of throughput. Higher values are better in terms of training and synchronization performance. This symbol is used for block synchronization and equalizer training.

4.2 Decoding/demodulation

The description of the decoding/demodulation for the PSK constellation application is virtually identical to that of the QAM Barnes-Wall lattice and we concentrate the description upon the latter.

There are various methods that can be used for the demodulation/decoding of the Barnes-Wall lattice. The method being implemented is simple but sub-optimal. The process consists of the sequential soft-decision decoding of the (8,1;8) (8,4;4) and the (8,7;2) codes. In more detail, suppose that we receive eight complex points, $(r_1, r_2, r_3, r_4, r_5, r_6, r_6, r_8)$ and we wish to estimate the point from $BW(16)$ that was transmitted. The following process takes place:

i) For each received point r_i, $1 \le i \le 8$, we form $d(i) = d_{0,i}^2 - d_{1,i}^2$, where $d_{\delta,i}$ is the squared Euclidean distance from r_i to the nearest point ending in a lsb = δ, for $\delta = 0$ or 1.

ii) We compute $d = \sum_{i=1}^{i=8} d(i)$. If d is positive, we decode the lsbs to (0,0,0,0,0,0,0,0). If d is negative, we decode the lsbs to (1,1,1,1,1,1,1,1). Suppose that the decoding decision is that $t_{0,i}$ was sent where $t_{0,i}$ is a constant (0 or 1) for all $i : 1 \le i \le 8$.

iii) For each received point r_i, $1 \le i \le 8$, we form $c(i) = c_{0,i,t}^2 - c_{1,i,t}^2$, where $c_{\delta,i,t}$ is the squared Euclidean distance from r_i to the nearest point that has its two least significant bits equal to $(\delta, t_{0,i})$, for $\delta = 0$ or 1. More intuitively, we look for the metrics within the subconstellation of points selected by placing the lsb equal to the decoded value.

iv) Using the $c(i)$ as metrics, we apply soft-decision decoding to the (8,4;4) code. (This can be accomplished, for example, by the Hadamard transform [ref. 19]). Suppose that the decoder decides the secondmost least significant bits are equal to $(t_{1,i} | 1 \le i \le 8)$, where these form a (8,4;4) codeword.

v) For each received point r_i, $1 \le i \le 8$, we form $b(i) = b_{0,i,t}^2 - b_{1,i,t}^2$, where $b_{\delta,i,t}$ is the squared Euclidean distance from r_i to the nearest point that has its three least significant bits equal to $(\delta, t_{1,i}, t_{0,i})$, for $\delta = 0$ or 1.

vi) We place $t_{2,i}$ equal to 0 if $b(i)$ is positive and 1 otherwise. If the sum of the $t_{2,i}$ is odd, we locate $i_{\min}$, the value of i for which $b(i)$ is smallest in absolute value, and invert $t_{2,i_{\min}}$.

vii) For each received point r_i, $1 \le i \le 8$, we find the constellation point $m(i)$ having the least squared Euclidean distance from r_i subject to the point having its three least significant bits equal to $(t_{2,i}, t_{1,i}, t_{0,i})$. The sequence of $m(i)$ is the demodulated signal, or, equivalently, the decoded point of the $BW(16)$ lattice.

5) System Description

From Sections 2, 3 and 4, we are in a position to describe the system and to understand the motivations for the various combinations of techniques. In 5.1 we describe the process at the transmitter. In 5.2 the receiver process is described. The notation as to symbol and sample order is defined in Section 5.1. A system block diagram is shown in Figure 8.

5.1 Transmitter process and channel notation

The data is assumed independent and identically distributed (iid) with a probability = 0.5 of a 0 or 1, i.e. it has been encrypted. Data is encoded using the binary codes described in Section 4. We concentrate the description on the case of 256 QAM. This takes, for a targeted 8 bit per modulation symbol system, 54 bits and encodes them into 8 labels where each label contains 8 points. A ninth label is added to each BW(16) lattice point. This ninth label contains n_s pre-set synchronization bits and $8 - n_s$ data bits. Thus, if $n_s = 2$, a total of 60 bits are encoded into a total of nine labels. When we refer to label number 13, we mean the 13-th label to leave the coding device. Labels are integers. (Conceptually, we could dispense with labels and regard the encoder output as a sequence of complex constellation points. However, it is implementationally easier to interleave integer labels rather than complex numbers.) We now helically interleave the labels in the exact fashion described in Section 3 and Figure 5. The interleaved stream of labels are used to select complex constellation points and these points are transmitted. Point 13 corresponds to the point that is selected by the 13-th label leaving the encoder. Points are members of the constellation, i.e. they are complex numbers.

5.2 Receiver process

The transmitted points are subject to a convolution with the channel response function and the addition of Gaussian noise. The resulting signal is sampled at the transmitter at the signaling rate (this article does not consider fractionally spaced equalizers). The numbering convention reflects the same numbering of the underlying sequence of constellation points. For example, sample number 13 refers to the sample of the received signal that would, in the absence of any intersymbol interference and additive noise, correspond to the constellation point selected by the 13-th symbol leaving the encoder. Sample numbers are complex numbers that might not belong to the constellation. Note that there is a difference between sample number 13, point number 13 and label number 13. The processing is described assuming that block synchronization (which is a study in its own right) has been achieved. The broad idea is that there a bank of eight slow DFE equalizers and one fast DFE equalizer. The combined complexity, in terms of equalizer operations per bit, is approximately equal to that of two fast equalizers. The fast equalizer does not enjoy any advantage over that of a conventional DFE (it is also not trusted to be as correct as the slow equalizers). This equalizer processes the input signal samples in a conventional process. When a non-synchronization symbol is being processed, the equalizer tap updates use data-directed mode, i.e. the difference between the predicted symbol and the received signal sample is used as the error signal to re-adjust the tap values. The re-adjustment could use any of a wide variety of algorithms but, in fact, we are using the simple LMS gradient algorithm. The Kalman Algorithm and its various derivatives would be a (high complexity) alternative. Upon receiving a synchronization symbol, the fast equalizer still uses data-directed mode but the difference is between the predicted symbol and the closest constellation point that has a label ending with the pre-set synchronization pattern. Since the sub-constellation of possible synchronization points has superior distance properties than the entire constellation, the update on synchronization points is more accurate. The process is seen as two-fold; the synchronization symbols "snaps the equalizer in" and the other updates try to track the changes.

The slow equalizers work in an iterative fashion. Throughout this description we assume that the system is giving correct data. In Figure 9, we show the equalizers at the start of a cycle. We will describe the process for codeword 73-80. The registers to the left are the feedforward part of the equalizers. They hold channel samples. The labels on the feedforward registers refer to the sample times with the above notation. Note that each register contains the samples received immediately prior to and including the symbol that it is attempting to predict. The registers to the right are the decision feedback registers. They hold decisions as to the immediately preceding constellation points. It is important to note that the decision registers hold two types of decisions. The shaded areas of memory refer to the decisions made by the fast equalizer. These are, given the correct decoder assumption, relatively unreliable. The unshaded areas in the decision registers refer to constellation points that have either been decoded prior to this codeword or which are synchronization symbols. These are relatively reliable. Consider the iterative procedure. The slow equalizers predict the constellation points of the codeword. These predictions are corrupted versions of the original sequence and the decoder is called upon to correct the noise (a non-linear process). The output of the decoder is, by assumption, the correct sequence of constellation points. For each register, the decoded constellation point can be compared to the predicted point to get an error signal. Note that this will be more reliable than the normal data-directed error signal (for example, one of the equalizers might diverge completely and still be re-trained with the decoded data-directed error signal; this is unlikely to happen with a single data-directed equalizer). The taps for each slow equalizer are re-trained using the decoded data-directed error signals. The decoded constellation points are stepped into the decision feedback registers, the entire contents and taps of each register are rotated up and new channel samples are entered into the feedforward sections. This, except for the lowest register, gives the next stage of the iteration, as shown in Figure 10. The taps for the lowest register are obtained from the fast equalizer. After the fast equalizer has processed the synchronization symbol, the taps and contents of the equalizer are transferred to the bottom equalizer. With this transfer, the iteration is closed and we are ready to repeat the process for codeword 82-89, as shown in Figure 10. There are several points to note:

i) The most important part of the decision feedback cancellation process are the decisions immediately preceding the present sample. These are the most reliable in this scheme. Note that the top registers are more reliable than the low registers as they hold less undecoded decisions and that the undecoded decisions are farther back in time relative to the predicted symbol.

ii) If the channel is varying in time, the taps values of each equalizer can vary from each other. The decoding process does not imply correlation between register tap values.

iii) In terms of implementation, the registers and taps are not "rotated up" The registers are arranged as a circular area of memory with a pointer to each area. The "rotation" consists of decrementing each register's pointer modulo 8. The register with the pointer equal to zero has the contents and tap values discarded and replaced by those in the fast equalizer. (Note that, with this approach, the inputs to the decoder have to be cycled before they are in correct order). Similarly, the equalizer that is going to be discarded is not updated.

iv) There are only seven slow equalizer updates per codeword and only one codeword per nine channel symbols. There is one fast equalizer update per channel symbol.

6) Model Results

Before embarking upon this design the principle of helical interleaving and DFEs was investigated using simpler models. In these experiments, one-dimensional PAM was used for modulation. The channels were assumed stationary and the tap values pre-computed. The channels were taken from a magnetic recording source [ref. 8] and are shown in Figure 11. They are similar except in severity. They are referred to as Channel 1 and 3 (Channel 2 was intermediate in definition and results). The experiments used a (16,5;8) Reed-Muller decoded with soft-decision information via a Hadamard transform. The performance curves show four lines: the matched filter bound (i.e. the performance if no intersymbol interference or coding is present), conventional DFE (no coding and interleaving), co-designed receiver with corrected feedback (i.e. the full scheme with corrected data in the decision feedback registers whenever possible) and a receiver that uses interleaving and decoding but which does not attempt to replace the DFE estimates in the decision registers. The coding results were penalized by $10\log_{10}(17/5) \equiv 5.31$dBs, i.e. for the utilized redundancy. Figure 12 shows the results for Channel 1. The straight addition of coding with no corrected feedback does little more than Conventional DFE once the normalization is added. However, the addition of corrected feedback makes a considerable difference, even after normalization. The improvement to conventional DFE is approaching 2dBs with improving SNR. Figure 13 shows the curves for the much harsher Channel 3. The straight use of coding does not justify the redundancy; the normalized figures are worse than DFE alone. The addition of coding with corrected feedback and helical interleaving is worthwhile; even with normalization, there are significant power gains over conventional DFE. Without the normalization penalty, the scheme is working at much lower SNR. Note that the total gain can be considered as the sum of the coding gain plus the co-designed gain.

In other experiments, it was observed that the co-designed gain depends on the channel; different channels cause greater or lesser degrees of error propagation. At worst you get the coding gain, at best you get considerably more.

7) Conclusions

This is work in progress and present efforts are concentrating upon refining the design and to implement it on a Motorola DSP56000 Digital Signal Processing system. It is a pleasure to acknowledge the support of Comdisco Inc. in this regard.

However, it seems that the technique will provide superior performance when:

i) The channel has significant non-linearities and/or spectral nulls;

ii) The throughput delay is not a critical parameter;

iii) The transmission is broadcast.

In summary, a technique is presented that appears to offer significant benefits over a black-box approach to equalization and modulation. The technique is closest in spirit to a technique introduced by Eyuboglu [ref. 4] and fits into several other attempts to improve upon DFE [refs. 2,4-6,7,9].

References

[1] Kasturia, S., J. Asalanis and J. Cioffi, "Vector Coding for Partial Response Channels," submitted to *IEEE Trans. Info. Theory.*

[2] Forney, G.D., Jr. and A.R. Calderbank, "Coset Codes for Partial Response Channels," submitted to *IEEE Trans. Info. Theory.*

[3] Peile, R.E., "Coding in Correlated Gaussian Noise," pre-print.

[4] Eyuboglu, M.V., "Detection of Coded Modulation Signals on Linear, Severely Distorted Channels using Decision-Feedback Noise Prediction with Interleaving," *IEEE Trans. Comm. Theory*, Vol. 34, No. 4, April 1988.

[5] Eyuboglu, M.V. and S.U. Qureshi, "Reduced-State Sequence Estimation for Trellis-Coded Modulation on Intersymbol Interference Channels," submitted to *IEEE SAC*, Special Issue on Band-Limited Channels.

[6] Eyuboglu, M.V., "Coding and Equalization," CSI/ARO Workshop, May 1989.

[7] Price, R., "Nonlinearly Feedback Equalized PAM versus Capacity for Noisy Filter Channels," *Proc. of ICC '72*, Philadelphia, PA, June 1972.

[8] Proakis, J.G., *Digital Communications*, McGraw-Hill, 1983.

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

[10] Tong, P. and E.R. Berlekamp, "Improved Interleavers for Algebraic Block Codes," US patent 4 559 625, December 17, 1985.

[11] Peile, R.E., "Error Correction, Interleaving and Differential Pulse Position," *International Journal of Satellite Communication*, Vol. 6, 1988, pp. 173-187.

[12] Ungerboeck, G., "Channel Coding with Multilevel/Phase Signals," *IEEE Trans. on Info. Theory*, Vol. 28, No. 1, January 1982.

[13] Calderbank, A.R. and N.J.A. Sloane, "New Trellis Codes Based on Lattice and Cosets," *IEEE Trans. on Info. Theory*, Vol. 33, No. 2, March 1987.

[14] Forney, G.D., Jr., "Coset Codes I: Introduction and Geometrical Classification," *IEEE Trans. Info. Theory*, 1989.

[15] Sloane, N.J.A., "Binary Codes, Lattices and Sphere-Packings," in *Combinatorial Surveys: Proceedings of the Sixth British Combinatorial Conference*, Academic Press, 1977 (Ed. P.J. Cameron).

[16] Conway, J.H. and N.J.A. Sloane, *Sphere Packings, Lattices and Groups*, New York, Springer-Verlag, 1987.

[17] Forney, G.D., Jr., R.G. Gallager, G.R. Lang, F.M. Longstaff and S.U. Qureshi, "Efficient Modulation for Band-Limited Channels,"*IEEE J. Selected Areas Commun.*, Vol. SAC-2, 1984, pp. 632-647.

[18] Barnes, E.S. and G.E. Wall, "Some Extreme Forms Defined in Terms of Abelian Groups," *J. Australian Soc.*, Vol. 1, 1959, pp. 47-63.

[19] MacWilliams, F.J. and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, 1977.

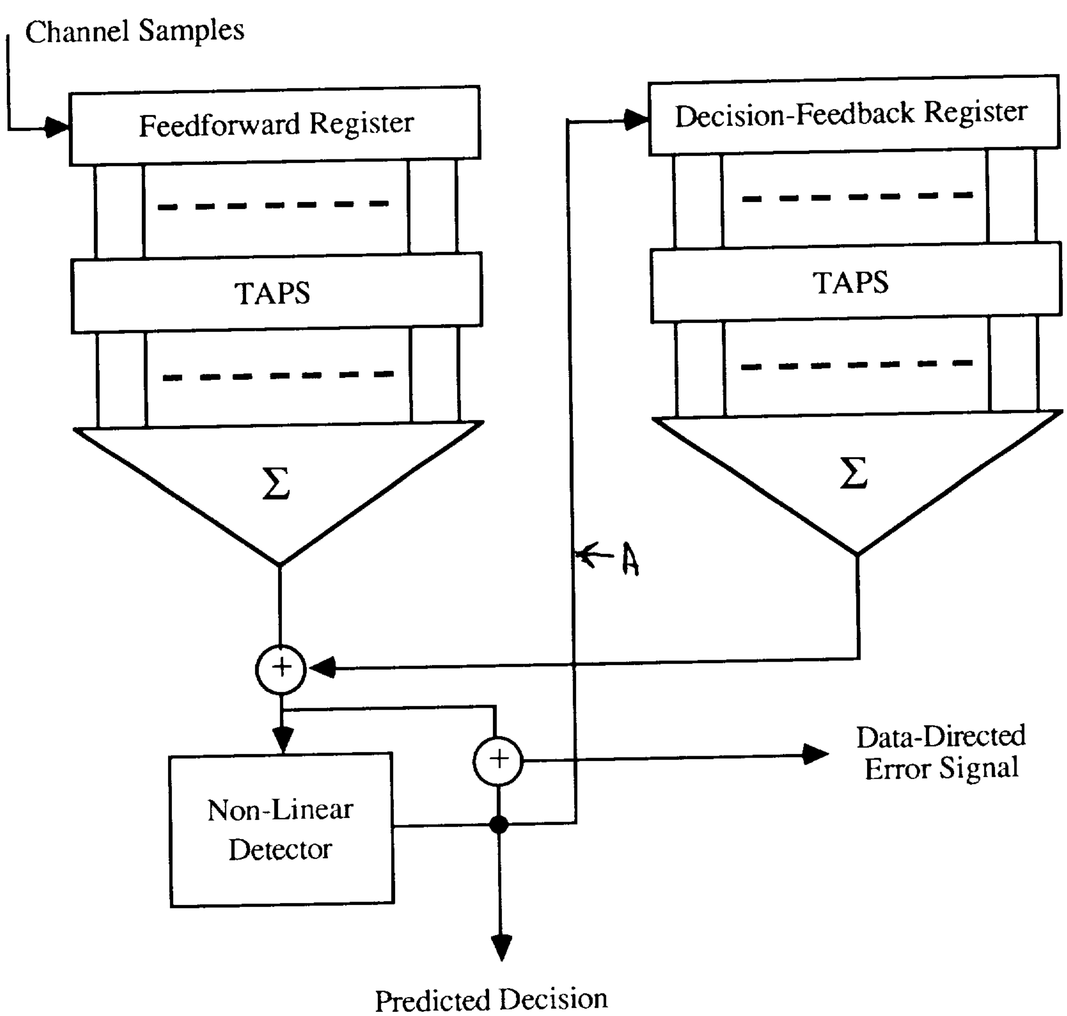
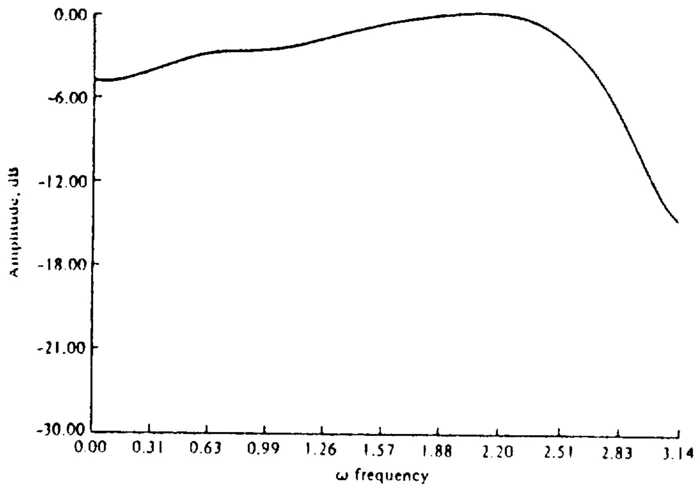
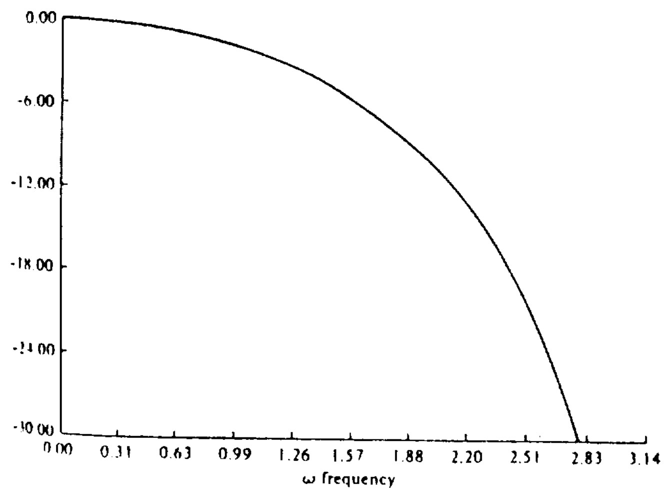


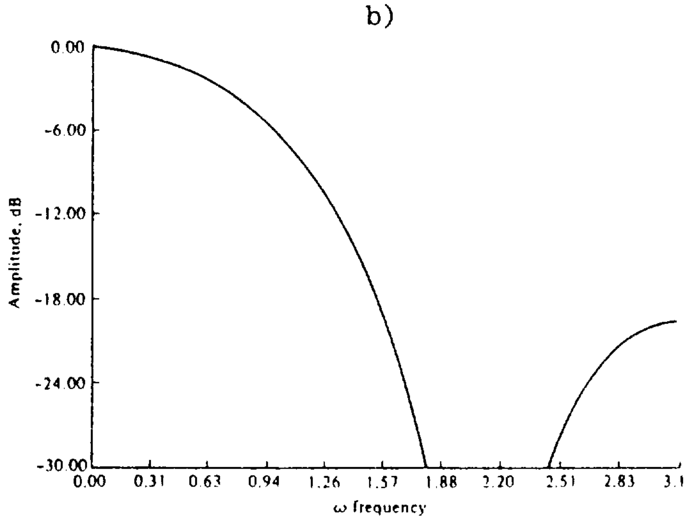
Figure 1: Decision Feedback Equalizer Schematic



a)



b)



c)

Figure 2: Channels

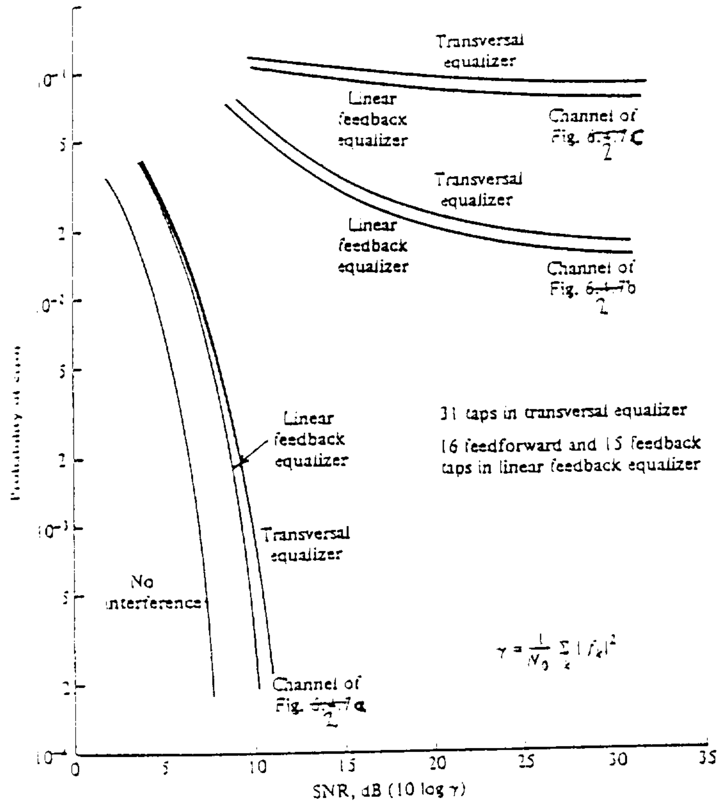


Figure 3: Error Rate Performance of Linear MSE Equalizer

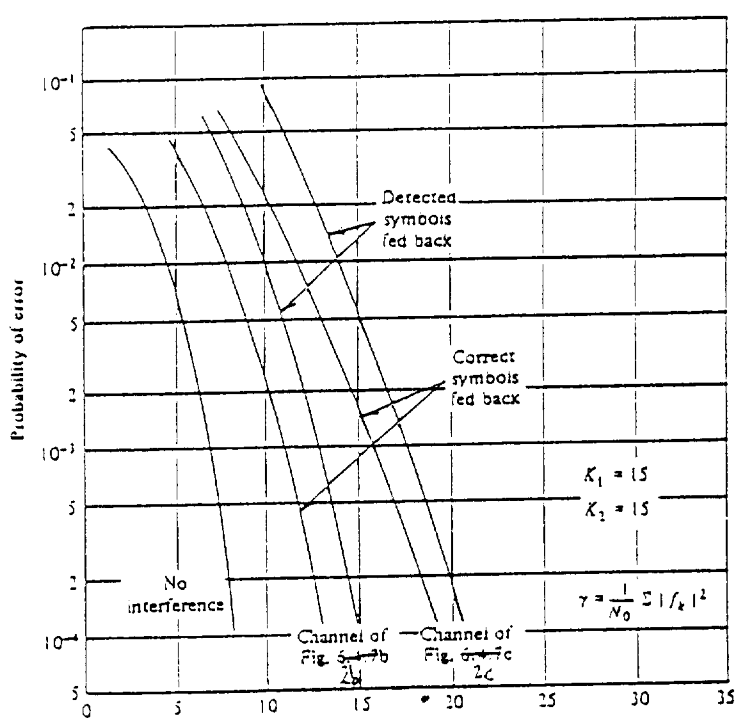


Figure 4: Performance of decision-feedback equalizer with and without error propagation.

5a: Encoded Data plus Sync Stream

1,2,3,4,5,6,7,8,	S
10,11,12,13,14,15,16,17,	S
19,20,21,22,23,24,25,26,	S
28,29,30,31,32,33,34,35,	S
37,38,39,40,41,42,43,44,	S
46,47,48,49,50,51,52,53,	S
55,56,57,58,59,60,61,62,	S
64,65,66,67,68,69,70,71,	S
73,74,75,76,77,78,79,80,	S
82,83,84,85,86,87,88,89,	S
91,92,93,94,95,96,97,98,	S
100,101,102,103,104,105,106,107,	S

5b: Interleaving Array

1,							
2,	10,						
3,	11,	19,					
4,	12,	20,	28,				
5,	13,	21,	29,	37,			
6,	14,	22,	30,	38,	46,		
7,	15,	23,	31,	39,	47,	55,	
8,	16,	24,	32,	40,	48,	56,	64,
S,	17,	25,	33,	41,	49,	57,	65,
73,	S,	26,	34,	42,	50,	58,	66,
74,	82,	S,	35,	43,	51,	59,	67,
75,	83,	91,	S,	44,	52,	60,	68,
76,	84,	92,	100,	S,	53,	61,	69,
77,	85,	93,	101,	109,	S,	62,	70,
78,	86,	94,	102,	110,	118,	S,	71,
79,	87,	95,	103,	111,	119,	127,	S,
80,	88,	96,	104,	112,	120,	128,	136,
S,	89,	97,	105,	113,	121,	129,	137,

5c: Transmitted Stream (Partial)

8,16,24,32,40,48,56,64,S,17,25,33,41,49,57,65,73,S,26,34,42,50,58,66, 74,82,S,35,43,51,59,67,75,83,91,S,44,52,60,68,76,84,92,100,S,53,61, 69,77,85,93,101,109,S,62,70,78,86,94,102,110,118,S,71,79,87,95,103, 111,119,127,S,80,88,96,104,112,120,128,136,

Figure 5: Helical example

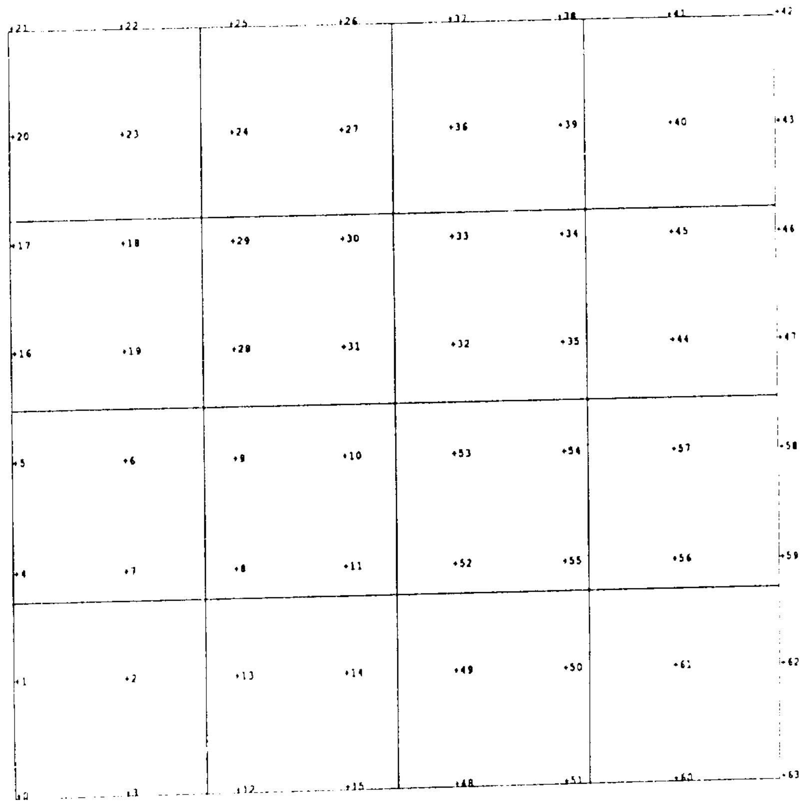


Figure 6

1	1	1	1	0	1	1	1	
0	0	1	0	0	0	1	1	Data-block
0	0	0	1	0	1	1	1	
1	0	1	1	1	1	0		
1	0	1	1					
1								

↓ Encoding

	1	1	1	1	0	1	1	1
	0	0	1	0	0	0	1	1
	0	0	0	1	0	1	1	1
	1	0	1	1	1	1	0	1
	1	1	0	0	0	0	1	1
	1	1	1	1	1	1	1	1
Integer:	37	35	53	45	5	45	49	63

(8,7,2) codeword
(8,4,4) codeword
(8,1,8) codeword

Figure 7: Encoding Example

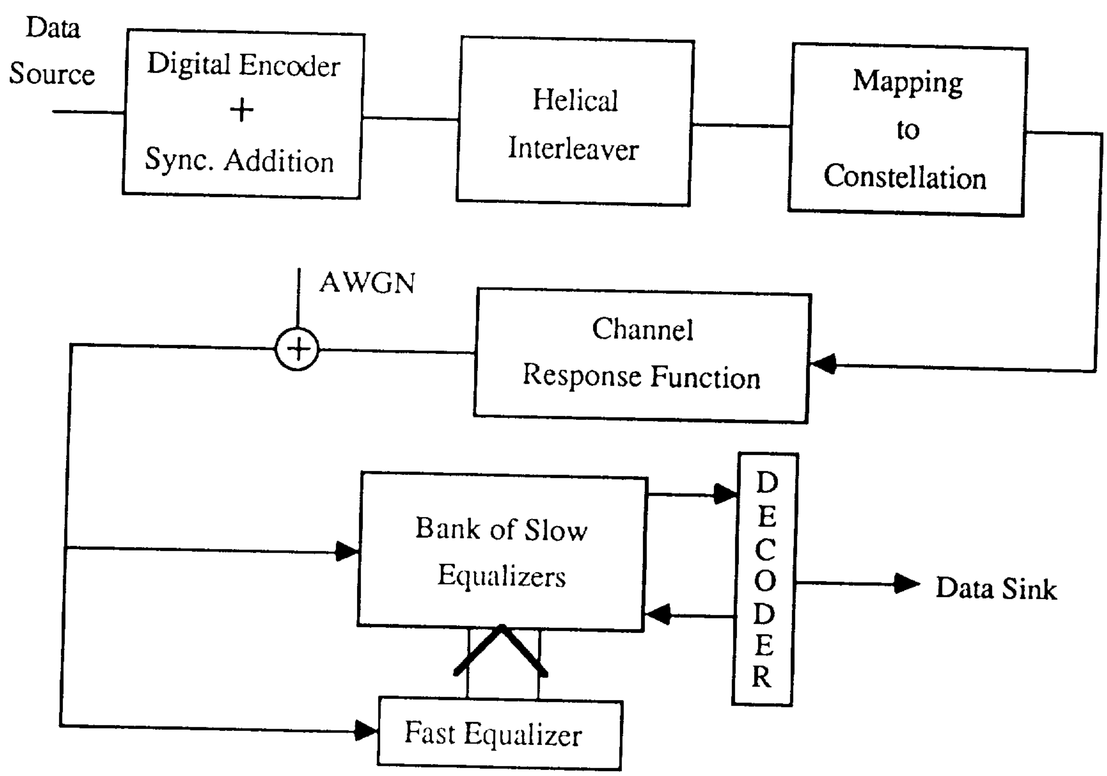
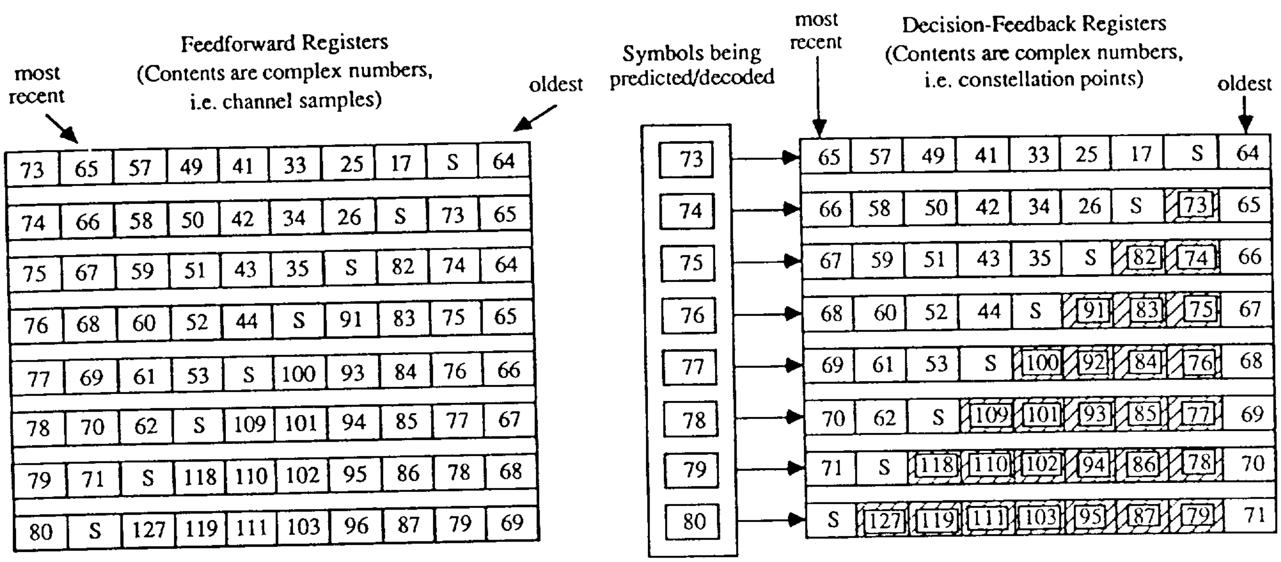
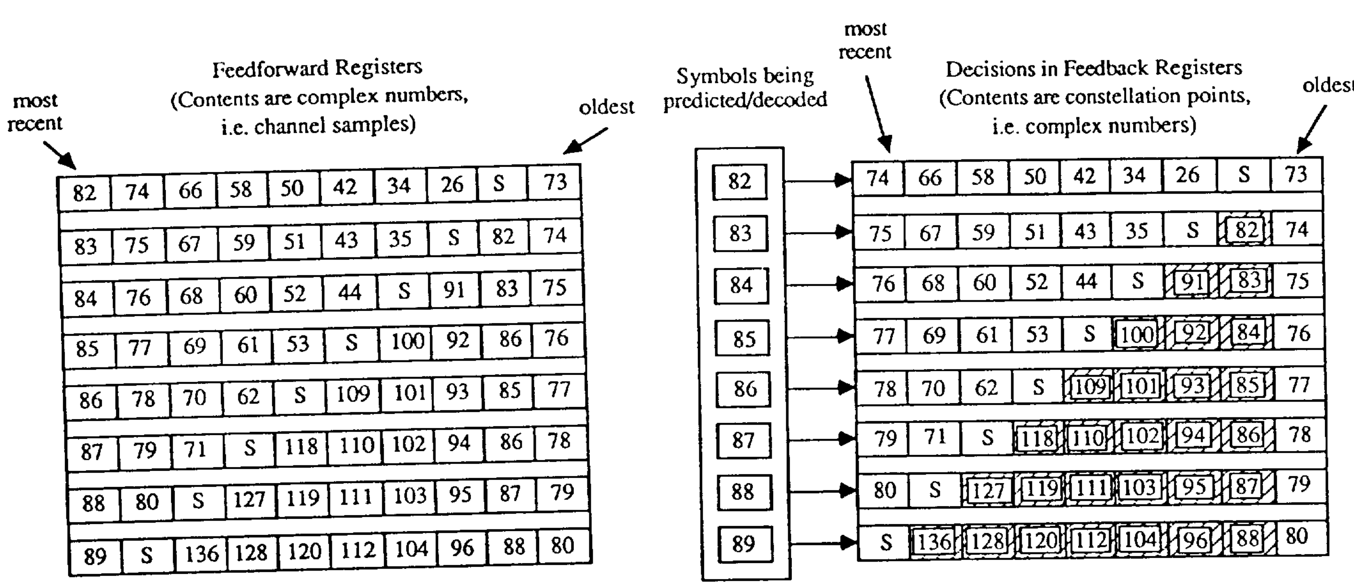


Figure 8: System Description



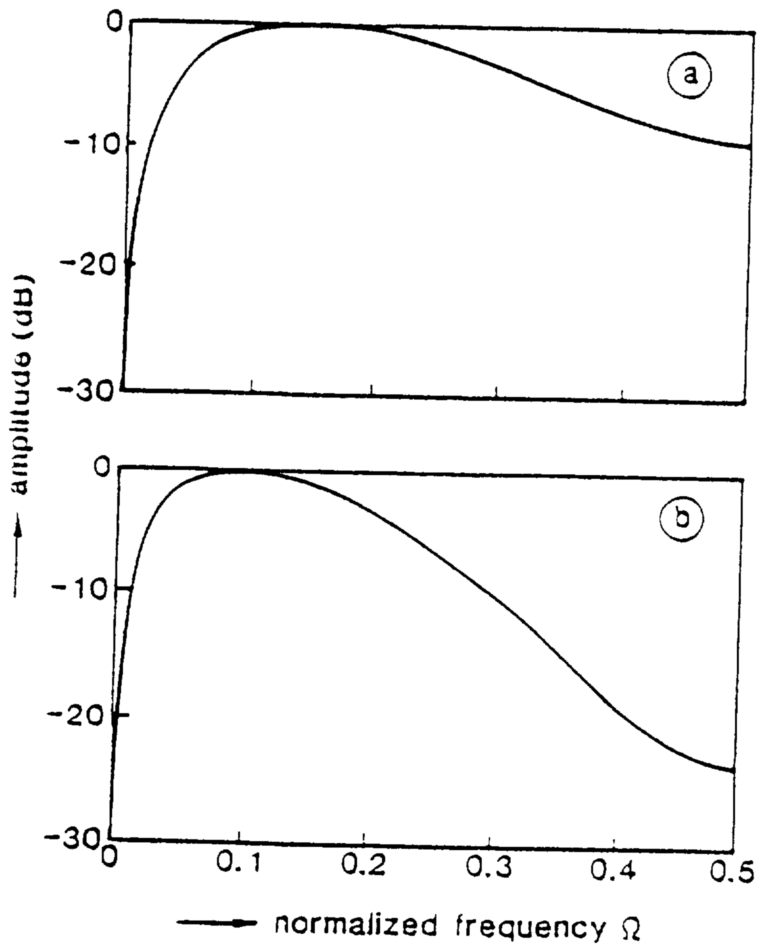
Relatively unreliable decision,
i.e. fast equalizer output

Figure 9: Contents of Slow Registers at Initial Prediction



Relatively unreliable decision,
i.e. fast equalizer output

Figure 10: Contents of Slow Registers at Next Codeword



Channel 1

Channel 3

Figure 11

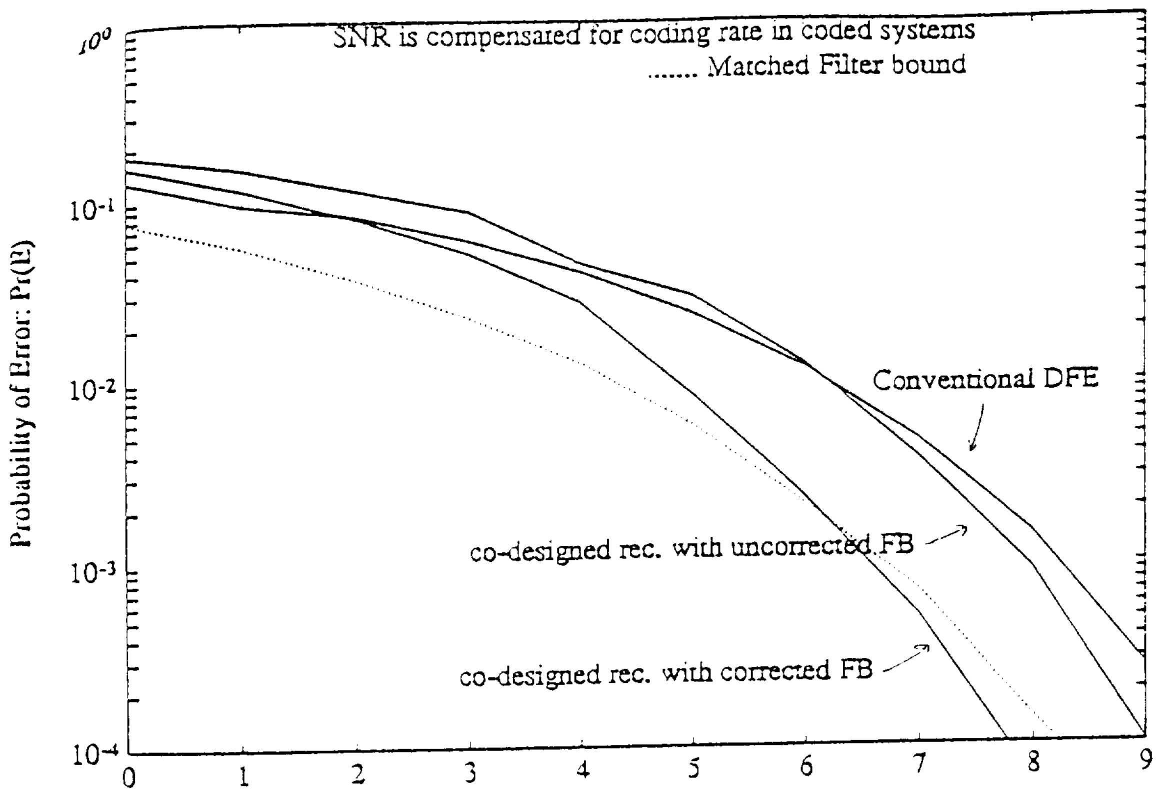


Figure 12: SNR vs Pr(E) plot for Channel 1

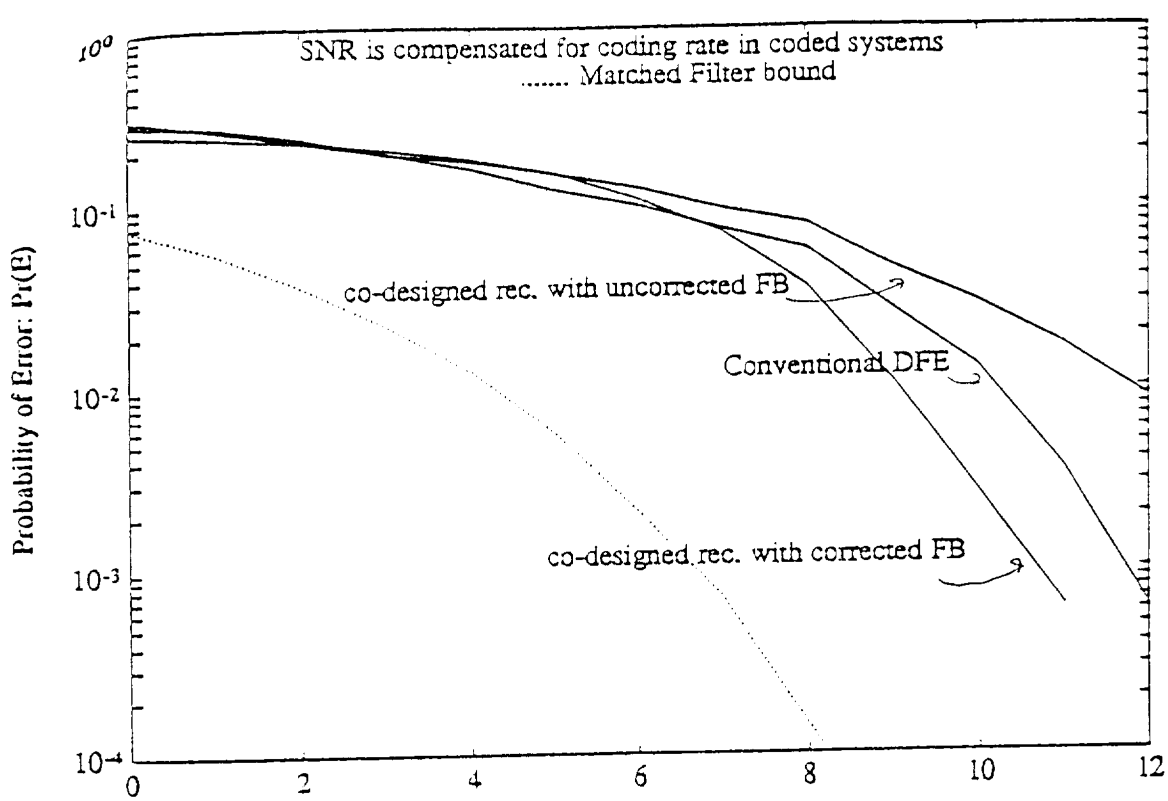


Figure 13: SNR vs Pr(E) plot for Channel 3