POWER, BANDWIDTH AND COMPLEXITY

IN

MAXIMUM LIKELIHOOD SEQUENCE ESTIMATION

POWER, BANDWIDTH AND COMPLEXITY
IN
MAXIMUM LIKELIHOOD SEQUENCE ESTIMATION

by

Cheung Woo Charles Wong, B. Sc.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree of

Master of Engineering

McMaster University

June 1980

MASTER OF ENGINEERING (1980)
(Electrical Engineering)

MCMASTER UNIVERSITY
Hamilton, Ontario.

TITLE: POWER, BANDWIDTH AND COMPLEXITY IN MAXIMUM LIKELIHOOD SEQUENCE ESTIMATION

AUTHOR: Cheung Woo Charles Wong, B.Sc.

SUPERVISOR: Professor J. B. Anderson

NUMBER OF PAGES: xi, 167

ABSTRACT

This thesis develops a two dimensional Viterbi Algorithm for the maximum likelihood sequence estimation over band limited baseband channels with intersymbol interference. Degradation, decision depth, 99% energy bandwidth and the channel cost are used as the performance measures for the comparisons of different channels. The four measures are extensively evaluated for channels with length up to four signalling intervals. The results of each measure are presented in contour form.

Error events analysis shows that the degradation contours are governed by elliptical equations. Maximum degradation results from state path merge at a depth equal to the channel length plus one. By analysing periodic state sequences, we found that catastrophic error propagation contours are mainly governed by linear equations. Generally, channels with longer length have narrower minimum bandwidth but higher degradation.

A channel cost similar to Shannon capacity equation is proposed to jointly minimize both degradation suffered and bandwidth required for signalling over a channel. According to the equation, the channel cost is influenced more by the bandwidth than by the degradation and thus the regions of low channel cost lie on the regions of narrow bandwidth. Also low channel cost regions are found to be on the regions of long decision depth and thus require higher complexity for maximum likelihood sequence estimation. In addition, it is found that minimum channel cost decreases with increasing channel length.

ACKNOWLEDGEMENT

The author wishes to express his sincere thanks to his supervisor, Dr. J. B. Anderson for support and guidance throughout this study. His inspiring suggestions and constructive criticisms have been most valuable. His careful reviewing of the manuscript is much appreciated.

This study was supported by the National Science and Engineering Research Council under grant no. A8828.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In a polar digital communications system, each data symbol gives rise to a baseband pulse every symbol period. Over a band limited channel, the pulses interfere with successive ones, resulting in intersymbol interference. Various receiver's structures with different complexities have been proposed to "disentangle" intersymbol interference but with different successes. For those receivers which cannot fully nullify the effect of intersymbol interference, degradations in the form of lowering of signal to noise (S/N) ratio and/or increasing the bit error rate, result. This thesis focusses on the usage of an encoding system that correlates the amplitude levels among pulses for spectral shaping. This system, known as a partial response signalling system, may result in bandwidth efficient coding. On the receiver's side, the Viterbi algorithm, which is an optimum way of implementing maximum likelihood sequence estimation, will be used.

Kabal and Pasupathy [8] gave a comprehensive study of partial-response signalling (PRS) systems. They modelled a PRS system as a transversal digital filter in cascade with a Nyquist filter, and the decoder is simply a decision feedback equalizer. Speed tolerances are considered, taking into account multi-level outputs and the effect of sampling phase. Eye width is introduced as a performance measure in

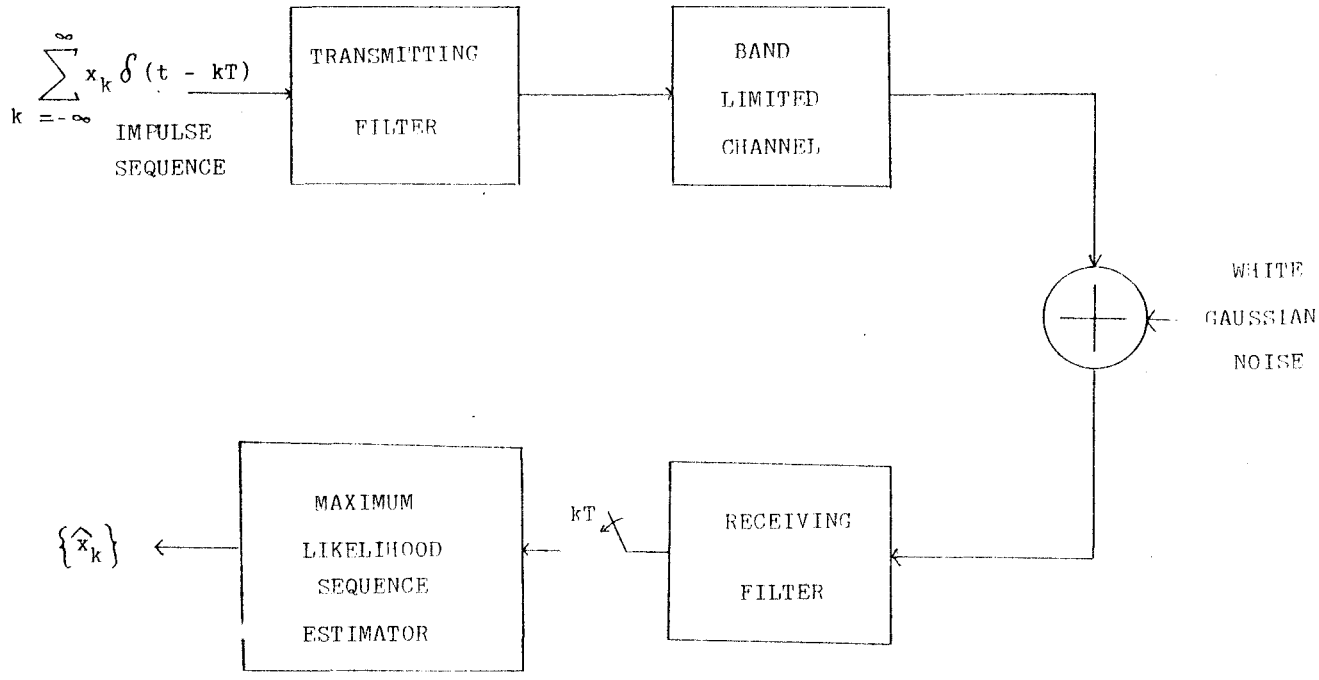


Fig. 1.1 A Digital Communications System Model

comparing PRS systems. The effect of an increase in the number of output signal levels is measured by SNR degradation over ideal binary transmission. Probability of symbol error including the effects of error propagation is considered using Markov chain models. The minimum bandwidth of the filter is always considered as 1/T, where T is the signalling interval. The codes being considered are formed from filters 1 - D and 1 + D.

Kobayashi [10] applied a non-linear processor known as the Viterbi algorithm to achieve maximum likelihood decoding of correlative level codes of the form $1 \pm D^k$, and analysed the performance by calculating the probability of symbol error using the minimum Euclidean distance concept. He pointed out the duality of codes $1 + D^k$ and $1 - D^k$, where k is an integer.

Forney [5] designed a maximum likelihood receiver for estimating digital sequences in the presence of intersymbol interference. It consists of a whitened matched filter followed by the Viterbi algorithm. The channel is modelled as a linear discrete time channel with memories similar to a PRS system. The error events concept was introduced in his performance analysis. Finally, Forney asserted that the probability of symbol error P(e) was bounded by $K_L Q(d_{min}/2\sigma) < P(e) < K_U Q(D_{min}/2\sigma)$, where K_L, K_U are small constants, Q(x) is the Gaussian error probability function, σ^2 the noise variance, and d_{min} is the minimum Euclidean distance between any two allowable signal sequences.

He asserted that, by using a preemphasis filter for the input sequences before transmission over the channel, any degradation due to intersymbol interference can be avoided. The preemphasis filter is in fact a PRS system with $F(D) = \mathcal{E}_x(D)$, where $\mathcal{E}_x(D)$ is the input error sequence.

This assertion is only true in the context of getting optimum performance in degradation alone. When both degradation and 99% power bandwidth are to be jointly optimized, we show that the preeamphasis filter will require a wider bandwidth and thus a worse overall performance. Also the complexity of the Viterbi algorithm necessarily increases in preemphasis as the order of the linear channel is increased.

Qureshi and Newhall [2] proposed a sub-optimum receiver for the time-dispersion channel. They showed that for a channel with a memory of order one, the error probability with maximum likelihood sequence estimation approaches the lower bound. They went on and showed that the quantity $d^2_{min}/R_o = 0.58$ for a channel with memory of order two, where R_o is the impulse response energy of the channel. Hence, the performance of maximum likelihood sequence estimation cannot approach the no-inter-symbol interference lower bound for codes with memory of order two.

Messerschmitt [6], [7] developed a linear space geometric theory for intersymbol interference. An equivalence between the theories of intersymbol interference and wide-sense stationary discrete random processes was shown, and was used to demonstrate the equivalence of zero-forcing (decision-feedback) equalization to minimum mean-square linear interpolation (prediction) of a random process. He showed that a canonical relationship exists between the minimum Euclidean distance d_{min} and the decision feedback error $||e_o^+||$ and that $d_{min} > ||e_o^+||$. He asserted that the amount by which S/N ratio of maximum likelihood decoding exceeds that of decision feedback equalization is governed by the coarseness of the best approximation to the projection by an element with restricted manifold coefficients, which for binary transmission is

1 and 0; the poorer the approximation, the better the S/N ratio of the maximum likelihood decoder

Magee and Proakis [15] estimated the worst case error probability for maximum likelihood sequence estimation on channels with different memory order. The bound follows that of Forney [5], and estimating the upper bound on error probability becomes essentially finding the d_{min}. They used the fact that the minimum d^2_{min} for channels with unity energy constraint on the pulse response is the minimum eigenvalue of the matrix of a suitable quadratic form. For channels of length greater than three, they consider input error sequences of various degree and select one that may give rise to minimun d^2_{min}. A positive definite matrix is then formed depending on the chosen error sequence. The minimum eigenvalue and the corresponding eigenvector then will be the minimum squared distance and the minimum distance channel respectively for that particular error sequence. This elimination method was used for all channel lengths up to 10. No procedure has been given to find the minimum d^2_{min} for all channel lengths.

Anderson and Foschini [22] provided a procedure for finding the minimum d^2_{min} for classes of systems of moderate complexity, that is, up to a few hundred states. Their way of expressing d^2_{min} originated from a functional analysis computer search approach. The determination of a closed-form expression for d^2_{min} proceeds by selecting the crucial error patterns from the full tree. They found the necessary and sufficient error sequence patterns to cause d_{min} for binary, three-level and four-level signals with memory of order 2, 3, and 4. They also listed the

above error sequence pattern for channels with a null at the band edge for memory of order 4. Specifically, they found that $\epsilon_x(D) = 1 \pm D$ gives the minimum d^2_{min} for all channels with length up to six, which previous work [15] did not show.

The purpose of this work is to evaluate the performance of the maximum likelihood receiver structure as proposed by Forney in the presence of intersymbol interference for baseband channels. This task in turn is equivalent to finding the performance of the receiver for channels having finite duration impulse response. These channels can be modelled as partial response signalling systems with real number tap-gains.

In this way, the effect of the band-limitation induced intersymbol interference can be considered to reside entirely in the encoder instead of the baseband channel, hence allowing us to vary solely the encoder and the transmitting filter for the optimization of the overall performances of the communication system. In our case, the encoder and transmitting filter are formed from a partial response signalling system. In this study, a partial response signalling system is used interchangeably with a channel with finite impulse response.

The second chapter of the thesis explains the above rationales in full details. In addition, the application of dynamic programming for maximum likelihood sequence estimation in memoryless noise is developed. The third chapter goes on to quantify the performances of different optimum linear receivers. Their similarities and differences with a non-linear receiver is highlighted. Also the Viterbi algorithm for the detection of partial response signals is formulated.

In chapter four, the error event concept is introduced and used to develop the bounds on the probability of symbol error following Forney's approach. In addition, "double" dynamic programming is developed for finding the free distances of partial response signalling systems.

The rest of the thesis concentrates on developing the appropriate performance measures for the comparisons of different partial response signalling systems with different tap-gains and constraint lengths. This is for the searching of channels with good impulse responses for maximum likelihood sequence estimation.

Four parameters are selected as the performance measures for different channels. Channels with length up to four are considered. The four parameters are degradation, decision depth, 99% energy bandwidth and the channel cost. Degradation is a measure of the effective decrease in the energy of a signal sequence in distinguishing itself from another in a channel with impulse response over a number of signalling interval, compared with that of an isolated pulse.

Decision depth is a variable which indicates the complexity, in terms of memories and computations required, for maximum likelihood sequence estimation using "double" dynamic programming. 99% energy bandwidth is the frequency band within which 99% of the energy of the channel's impulse response lies. It gives a guide of the practical bandwidth provided by the channel.

Finally, in a venture to jointly optimize both degradation and bandwidth, a channel cost function similar to Shannon capacity equation is proposed and evaluated.

CHAPTER 2

DYNAMIC PROGRAMMING AND PARTIAL RESPONSE SIGNALLING SYSTEMS

This chapter introduces the concept of dynamic programming and its application to maximum likelihood decoding of Markov process in memoryless noise. Furthermore, this chapter provides some rationale of utilizing partial response signalling systems as the transmitting filter in our communications system model.

2.1 Maximum Likelihood Decoding

In digital communications, a noisy channel distorts the transmitted sequences in a stochastic manner. A channel with input sequences from an input alphatet I and output sequences from an output alphabet Y can be described by a conditional probability distribution $P(\underline{y}/\underline{x})$, where $\underline{x}$, $\underline{y}$ are input and output vectors.

A decoder is a device that instruments a decoding rule for choosing the transmitted sequence among all possible sequences on the basis of the received sequence. Consider the decoding rule which minimizes P(e), the probability of code word error. Let $\hat{\underline{x}}$ be the estimated input vector or codeword. Then $P(e/\underline{y}) = 1 - P(\hat{\underline{x}}/\underline{y})$. This is equivalent to maximizing $P(\underline{y}/\underline{x})P(\underline{x})/P(\underline{y})$, by Bayes' rule. As only $P(\underline{y}/\underline{x})$ depends on the channel, we define a maximum likelihood decoder (MLD) as a decoder which, given the received vector $\underline{y}$, sets $\underline{x} = \hat{\underline{x}}$ such that

$P(\underline{y}/\underline{x})$ is maximized.

Main advantage of this decoder includes:

1) The MLD depends only on the channel and is independent of how the codewords are chosen.

2) The MLD is optimal in the sense of minimizing P(e) in the important case when the information symbols are statistically independent and equally likely.

3) The MLD sets $\underline{x} = \hat{\underline{x}}$ about which the received vector $\underline{y}$ gives the most information. The mutual information of a data symbol 'a' supplied by 'b' is $I(a/b) = \log P(b/a)/P(b)$. Hence, $I(\underline{x}/\underline{y}) = \log P(\underline{y}/\underline{x})/P(\underline{y})$, and maximizing $P(\underline{y}/\underline{x})$ in the MLD is the same as maximizing the mutual information between y and x.

4) The MLD gives a constant low P(e) independent of the actual selection of the codewords.

Although the MLD is an optimal decoding rule, it is often not used in practice due to complexity involved in its implementation. Consider a block code of length N and size M, where M is the number of distinct sequences called codewords. Each data symbol belongs to some alphabet I = 0, 1, 2, ..., m - 1 so that $M \leq m^N$. The rate of the code is defined as $R = (\log_m M)/N$. A brute-force application of the maximum-likelihood decoding requires m^{RN} calculations of the conditional probability distribution P(y/x). If the noise in the channel is not too large then a suboptimum rule such as bounded-distance decoding can be used instead.

2.2 Dynamic Programming Concept and the MLD

The most commonly used MLD is the Viterbi Algorithm (VA), first introduced as an optimum decoder for convolutional codes. As it is based on dynamic programming concepts, let us have a clear understanding of these principles.

We conceive of a system as a state vector consisting of K state variables. The smallest set of variables which determine the state of the system are termed the state variables. The state has the property that the knowledge of these variables at $t = t_o$ together with the inputs for $t > t_o$ determine the behaviour of the system for any time $t > t_o$; i.e., $s(t) = (x_1(t),....x_K(t))$. One important point is that the state of the system at $t > k$ depends only on s_k; we do not require the past history of the system to determine the future. The future is uniquely determined by the present.

Suppose we have sufficient influence over the system so that at each stage i we can choose a variable $q_i \in Q$, where Q is the set of allowable decisions and $s_{i+1} = \beta(s_i, q_i)$ for all i [1]. A decision among the q_i forces a change of state. The process is deterministic if a decision causes a unique change of state. The process is discrete if there are only a finite number of decisions. An N-stage discrete deterministic process is denoted by the set of vectors

$$\mathcal{D}_N = \{(s_o, s_1,...s_N), \ (q_o, q_1,...q_N)\},$$

with $s_{i+1} = \beta(s_i, q_i)$ for each i. We are concerned with processes in which q_i are chosen so as to maximize or minimize prescribed scalar

function of the state and decision variable $R(\mathcal{D}_N)$.

We want to evaluate the "goodness" of various sequences of decisions. We need a criterion possessing a structure which permits us to concentrate solely upon the past and present history of the process in the search for values of q_i. Thus we restrict to functions of form $q_k = \gamma(\mathcal{D}_{k-1})$. This function is called a policy function. A policy function is any rule for making decisions which yield an allowable sequence of decisions; the policy which maximizes or minimizes the criterion or return function is called an optimal policy.

The above policy function is too general and we wish to have the policy of form $q_k = \gamma(s_k)$, a function only of the current state. A subpolicy refers to a sequence of connected decisions which form part of a policy. The theorem of optimality [19] states: An optimal policy must contain only optimal subpolicies .

Proof:

Consider a subpolicy extracted from an optimal policy. If such a subpolicy were not optimal then there exists a better one which if added to the remaining portion of the policy under consideration would improve the latter, a deduction contrary to the hypothesis that the latter is the optimal. Q.E.D.

As an illustration, consider a return function possessing Markovian nature; after any number of decisions k, the effect of the remaining (N - k) stages of the decision process upon the total return depends only on the states s_k of the system at the end of the k-th decision and the subsequent decisions. Let $R[\mathcal{D}_N] = \sum_{k=0}^{N-1} g(s_k, q_k)$ and

$f_N(s_o) = \max \mathcal{R}[\mathcal{D}_N]$; i.e., it is the maximum total N-stage return starting in state s_o using the optimal policy.

$$f_N(s_o) = \max_{q_o} \max_{q_1} \ldots\ldots \max_{q_{N-1}} \{g(s_o, q_o) + \ldots . g(s_{N-1}, q_{N-1})\}$$

$$= \max_{q_o} \max_{[q_1, \ldots q_{N-1}]} \{g(s_o, q_o) + \ldots\ldots + g(s_{N-1}, q_{N-1})\}$$

$$= \max_{q_o} \{g(s_o, q_o) + \max_{[q_1, \ldots q_{N-1}]} \{g(s_1, q_1) + \ldots$$

$$+ g(s_{N-1}, q_{N-1})\}\},$$

as $g(s_o, q_o)$ is independent of $[q_1, \ldots\ldots q_{N-1}]$. Consequently,

$$f_N(s_o) = \max_{q_o} g(s_o, q_o) + f_{N-1}(s_1) \text{ with } s_1 = \beta(s_o, q_o). \qquad (1)$$

This is the dynamic programming approach.

The principle of optimality can now be stated in another form [1], "An optimal policy has the property that regardless of the initial state and the initial decisions are, the remaining decisions must contribute an optimal policy with regard to the state resulting from the first decision".

The backward dynamic programming approach is similar. Let $f_N(s_N)$ = maximum total N-stage return terminating in state s_N using an optimal policy. As $\max_{[q_o, \ldots q_{N-1}]} \mathcal{R} = \max_{[q_{N-1}, \ldots q_o]} \mathcal{R}$, accordingly,

$$f_N(s_{N-1}) = \max_{q_{N-1}} \max_{[q_{N-2},\ldots q_0]} \{g(s_{N-1}, q_{N-1}) + \ldots . g(s_0, q_0)\}$$

$$= \max_{q_{N-1}} \{g(s_{N-1}, q_{N-1}) + \max_{q_{N-2},\ldots q_0} \{g(s_{N-2}, q_{N-2}) + g(s_0, q_0)\}\}$$

$$= \max_{q_{N-2}} \{g(s_{N-1}, q_{N-1}) + f_{N-1}(s_{N-2})\} ,$$

with $s_{N-1} = \beta(s_{N-2}, q_{N-2})$. (2)

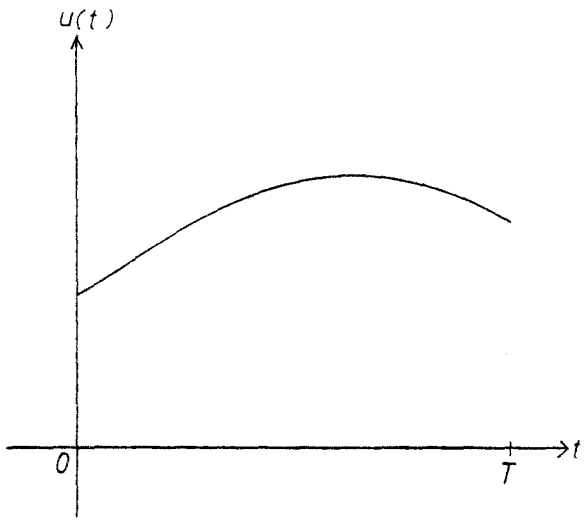
Both approaches form recurrence relations.

As noted by Bellman [1], "the decomposition of the problem of choosing a point in N-dimensional space into N choices of points in one-dimensional phase space is of utmost conceptual importance".

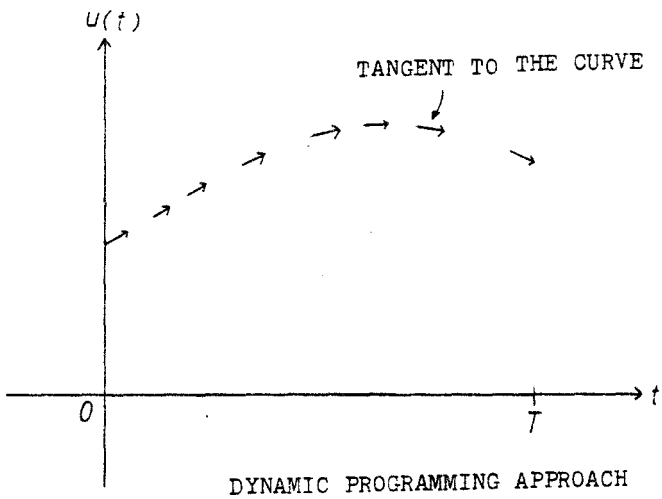
The comparison of the dynamic programming approach with the classical treatment of variational problem is summarized by Bellman [1]:

> "Classically one seeks a curve u = u(t) over (0, T) which maximizes. The unknown u is regarded as a point in function space. In our approach, at every point, we seek a direction which is optimal; the solution is obtained in the form of a policy, a set of instructions to carry out the process. In geometry parlance, we can say that the classical view of a curve as a locus of points, while dynamic programming considers a curve to be an envelope of tangents....Hence the two theories are dual to each other... This duality and equivalence remains valid, however, only for deterministic processes" See Fig. 2.1

A solution in the dynamic programming context can be given in terms of $f_N(s_i)$, the sequence of return functions, or $q_N(s_i)$ the sequence of policy functions. Each sequence uniquely determines the other. Another point is that the presence of constraints simplifies the determination of the solution; by means of constraints we are able to cut down on the allowable choices of policies at each stage. Thus, the search process



VARIATIONAL CALCULUS APPROACH



DYNAMIC PROGRAMMING APPROACH

Fig. 2.1 Two different principles for the maximization of a quantity in time.

is easier to carry out.

2.3 Tree Structure in Discrete Processes

A discrete deterministic process can be represented as a tree starting at the root of level 0. Levels and nodes in a tree structure correspond to stages and states in a process. The root gives rise to a finite number of nodes which in turn are roots of other trees. Thus a tree is able to represent exhaustively all states and stages of a process. At every stage, a decision is made which causes a specific branch to be followed. Thus by assigning return values to different branches of the tree in a corresponding way, an optimal policy can be regarded as a path through the tree with minimum or maximum values.

To give a dynamic programming formulation to a process requires:

1) Characterizing a physical system by a set of state variables and defining the allowable states at every stage.
2) Defining the appropriate return or criterion function.
3) Deriving a recurrence relation connecting the members of the sequence of return functions $f_N(s)$.
4) Setting up the appropriate boundary conditions or constraints.

In the dynamic programming approach, the maximum return is uniquely determined, but there may be more than one optimal policy which yields this return. Also both forward and backward recurrence relation formulations are feasible to most problems, so the choice depends on the ease of programming. In general, the tree or path representing a process is unstructured in the following senses. Refer to Fig. 2.2.

1) The number of allowable states at each stage is different.

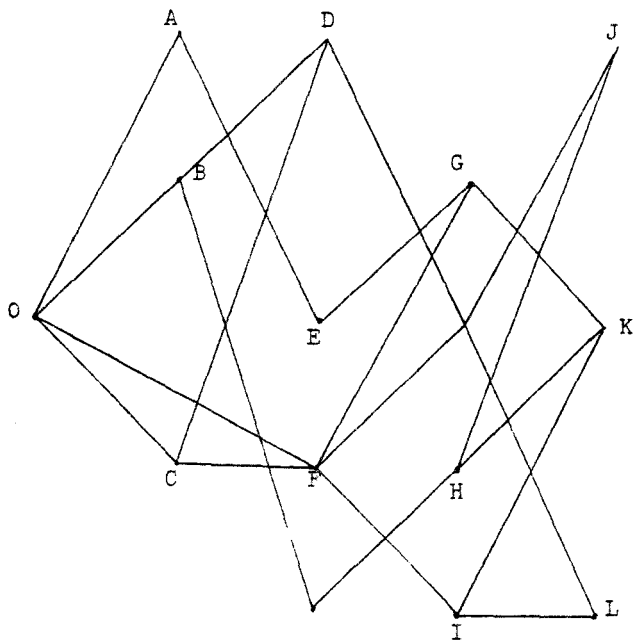


Fig. 2.2 An example of an unstructured graph

2) The number of allowable decisions in a given state at different stages varies.

3) The number of allowable transformations from different states to a given state differs at different stages.

2.4 Application of Dynamic Programming to a Discrete Markov Process

Consider a problem posed by Forney [4]: Given a sequence Z of observations of a discrete-time finite state Markov process in memoryless noise, find the state sequence S for which the a posteriori probability P(S/Z) is maximum. The process is modelled as:

Time is discrete. The state space is {0, 1,..., M-1}. Assume the process runs from time 0 to time N; i.e., the state sequence is $S = (s_0, s_1, \ldots, s_{N-1})$. As the process is Markov, the future is independent of the past conditioned on the present; i.e., the probability

$$P(s_{k+1}/s_0, s_1, \ldots, s_k) = P(s_{k+1}/s_k) \tag{4}$$

A transition α_k occurs at time k when the process changes from state s_k to s_{k+1} and is denoted by $\alpha_k = (s_k, s_{K+1})$.

Since the channel is memoryless, the sequence Z of observations z_k depends probabilistically only on the transition α_k at time k:

$$P(Z/S) = P(Z/\alpha) = P(z_k/\alpha_k), \tag{5}$$

with $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{N-1})$. We use the fact that the mapping from state sequence to transition sequence is one to one.

In some special cases,

1) z_k depends only on s_k, $P(Z/S) = \prod_{k=0}^{N-1} P(z_k/s_k)$.

2) z_k depends probabilistically on an output y_k of the process at time k, which in turn is a deterministic function of the transition α_k.

Now consider a type of Markov process called a shift-register process [4] shown in Fig. 2.3. An input sequence $X = (x_0, x_1, \ldots)$, consisting of x_k generated independently according to some probability distribution $P(x_k)$, can take on one of a finite number of values m. This input sequence is used to drive the sequential machine to generate a signal sequence $Y = (y_0, y_1, \ldots)$ in which each y_k is some deterministic function of the present state and input; i.e., $y_k = f(s_k, x_k)$. Define the state

$$s_k = (x_{k-1}, x_{k-2}, \ldots . x_{k-L}), \tag{6}$$

where L is the number of memory units in the shift-register. This is a L-th order m-ary Markov process. Note that y_k is not observable, but that the observed sequence Z is the output of a memoryless channel whose input is Y, i.e.,

$$\{z_k\} = \{y_k\} + \{n_k\}, \tag{7}$$

where $\{n_k\}$ is the additive noise due to the channel.

Maximum a posteriori (MAP) is a rule such that given a sequence Z of observations of a discrete-time process in a noisy channel, the signal sequence Y will be found for which the a posteriori probability P(Y/Z) is maximum. We only consider the case of a discrete-time finite-state Markov

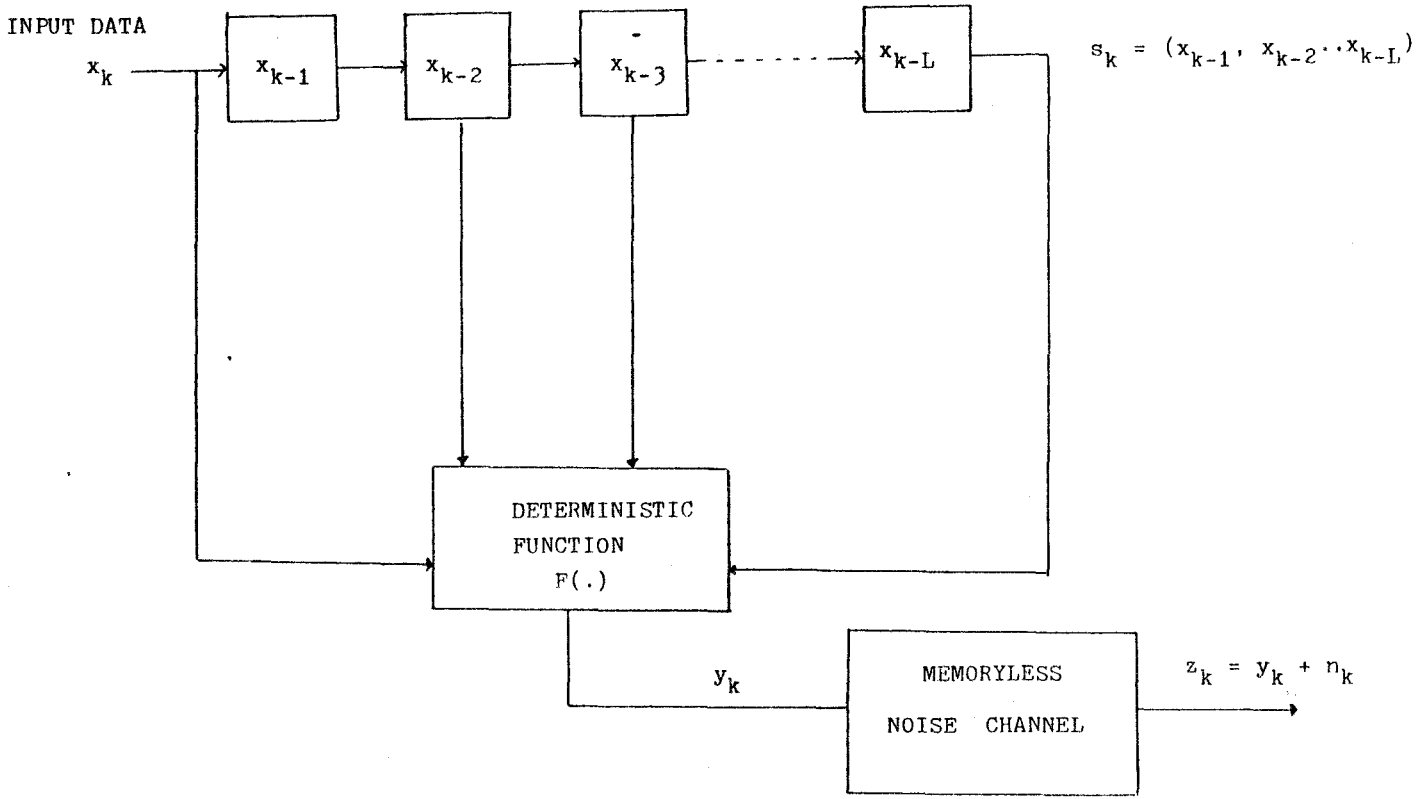


Fig. 2.3 A shift-register model

process in memoryless noise. For a m-ary shift-register process, finding the probable signal sequence Y is equivalent to finding the most probable state sequence S or the most probable input sequence X. The tree representing the above process is really a m-ary tree. Every node represents a state but not all states at a given level are distinct. The Viterbi Algorithm (VA) is a dynamic programming solution to the MAP rule for a finite-state Markov process in memoryless noise.

Maximum Likelihood sequence estimation (MLSE) is defined as the choice of Y for which the probability density p(Z/Y) is maximum. It can be shown [25] that the ML estimates correspond mathematically to the limiting case of MAP estimates in which the a priori knowledge approaches zero. MLD is usually used in the context for block codes while MLSE is used for sequences.

2.5 Generalized Partial Response Signalling Systems

A Partial Response Signalling (PRS) system is based on the shift-register process mentioned in section 2.4. The system consists of a digital transversal filter with impulse response

$$F(D) = \sum_{i=0}^{L} f_i D^i \qquad (8)$$

in cascade with a filter with frequency response $G(\omega)$, where D is the Huffman's delay operator and $\{f_i\}$ is the tap-gains of the transversal filter, with L being the number of T-delay units [5], [8]. See Fig. 2.4.

The filter is a finite-state machine with m^L states, the state-space is

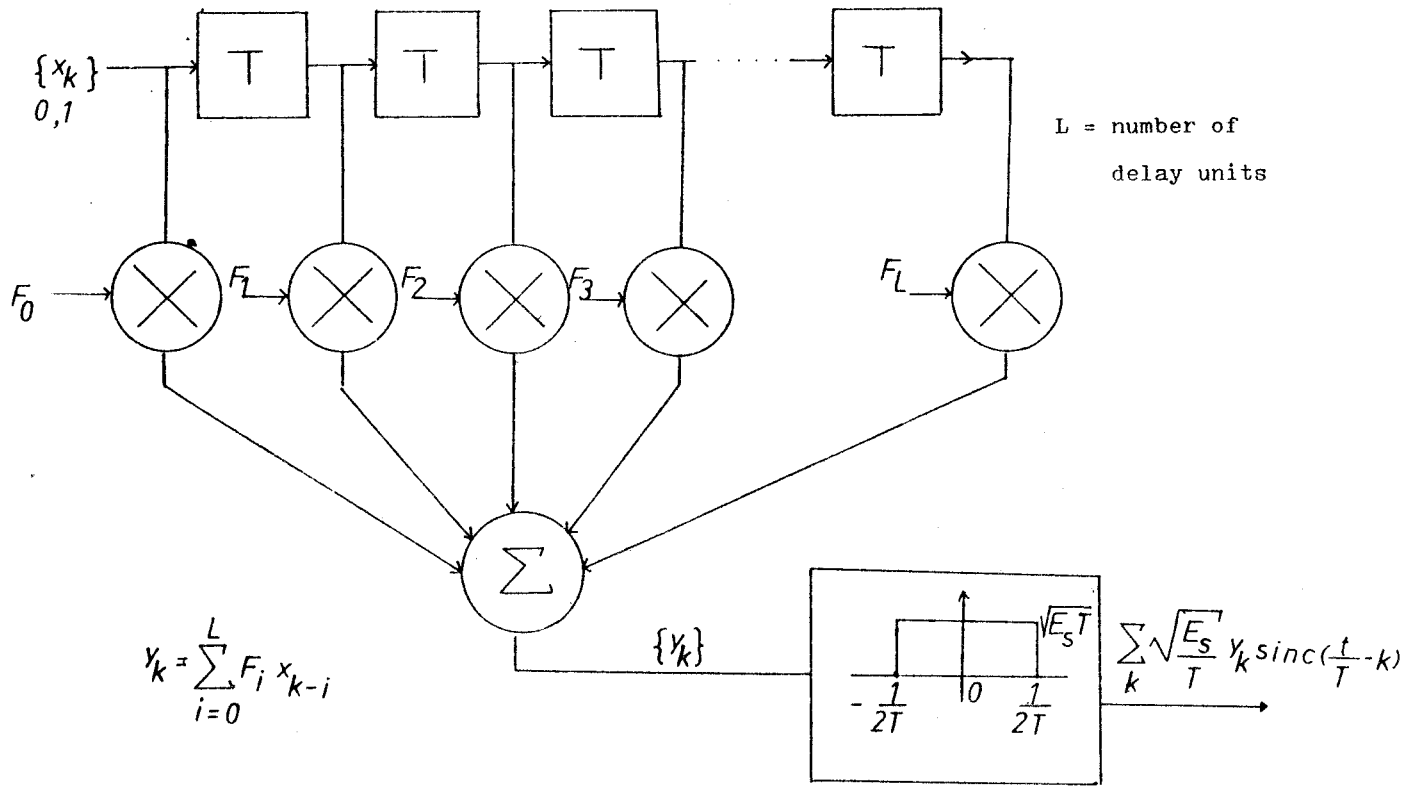


Fig. 2.4 A Partial Response Signalling System with K = L + 1 taps

$$S = \{0, 1, 2, 3, \ldots m^L - 1\}, \tag{9}$$

and the inputs are

$$X = \{0, 1, 2, \ldots m - 1\}, \tag{10}$$

As the machine is able to remember its L previous inputs, the state is defined as in eq. (6). Similarly, the output depends only on the present state and input; i.e.,

$$y_k = \lambda(s_k, x_k). \tag{11}$$

Thus the input sequence is

$$X(D) = \sum_{i=0}^{\infty} x_i D^i, \tag{12}$$

and the output sequence is

$$Y(D) = \sum_{i=0}^{\infty} y_i D^i. \tag{13}$$

The system is linear, therefore the output sequence Y(D) is the convolution of the input sequence with the impulse response F(D). Thus in D transform,

$$Y(D) = X(D)\, F(D) \tag{14}$$

with

$$y_k = \sum_{i=0}^{L} f_i x_{k-i} \quad . \tag{15}$$

The above representation is in the time domain.

In the frequency domain, the digital transversal filter has periodic frequency response of period 1/T given by [8],

$$F(\omega) = F(D)\Big|_{D = \exp(-j\omega T)}$$

$$= \sum_{k=0}^{L} f_k \exp(-jk\omega T). \tag{16}$$

The impulse response of the whole PRS system has sample values $\{f_k\}$ if and only if $G(\omega)$ satisfies Nyquist's first criterion, which requires that its impulse response has zeroes at uniformly spaced intervals except for a centreal peak [20]; i.e.,

$$g(kT) = 0, \; k = 0, \; g(0) = 1. \tag{17}$$

In the frequency domain, this implies [8]

$$\sum_{k=-\infty}^{\infty} G(\omega - 2k\pi/T) = T. \tag{18}$$

Any filter which satisfies Nyquist's first criterion is known as a Nyquist filter, and its bandwidth as the Nyquist bandwidth. There are several filters with different bandwidths which satisfy Nyquist's

first criterion. The minimum Nyquist bandwidth refers to the minimum bandwidth of any filter which satisfies the criterion [12]. The filter turns out to be an ideal low-pass filter of bandwidth $2W = 1/T$.

Conceptually, $F(\omega)$ forces the desired sample values $\{f_i\}$ but is periodic, while $G(\omega)$ preserves the samples and is used to bandlimit $F(\omega)$ [8]. $G(\omega)$ preserves the samples in the sense that it does not cause any overlapping of pulses at sampling time. $G(\omega)$ also can be thought of as converting the discrete-sample values into a continuous waveform.

2.6 Minimum Nyquist Bandwidth Filter and the Raised Cosine Filter

The impulse response of an ideal low pass filter, of bandwidth

$$W = 1/2T \tag{19}$$

and unity gain, is a sinc pulse

$$2W(\sin 2\pi Wt/2\pi Wt). \tag{20}$$

A sinc pulse is not desirable for signalling because of the precise timing required. The pulse response decreases as $1/t$ for large t. Any slight deviation in symbol rate, filter cut off frequency, or sampling instant would cause failure, as the overlapping tails represent a divergent series, and can add up to large values resulting in mis-interpretation of sample values [20].

With a more gradual roll-off of the low-pass characteristic, the oscillatory nature of the pulse is reduced and the tail decay is faster than $1/t$. One class of Nyquist filter called raised cosine [12] consists

of a flat amplitude portion and a truncated sinusoidal roll-off. It has the advantage of tolerating more deviation in the sampling instants than the ideal filter, because the response falls off faster and fewer pulse tails contribute significantly. Its spectrum is defined in terms of a parameter which specifies the amount of bandwidth used in excess of the minimum Nyquist bandwidth. In this work, we only consider the minimum Nyquist bandwidth because we want to find the maximum theoretically obtainable data rate in a channel of given bandwidth. Also for systems with bandwidth larger than π/T, aliasing will result when the outputs of such systems are sampled at the symbol rate. If a non-minimum bandwidth PRS system is used to equalize a channel characteristics, aliasing can cause nulls or near nulls in the Nyquist equivalent channel where non is intended, resulting in the degradation of the performance of the equalizer as noise enhancement will occur in compensating for these unintentional nulls. In addition, non-sinc pulses like the raised cosine filter will introduce analytical complexities beyond the scope of this work. Throughout this study the impulse response of a PRS system with an ideal low-pass filter of minimum Nyquist bandwidth and unity gain is given by

$$h(t) = \sum_{i=0}^{L} f_i \, \mathrm{sinc}(2Wt - i). \qquad (21).$$

2.7 Intersymbol Inteference and PRS Systems

In digital transmission through a linear, band limited analog channel, the input sequence X, in discrete time and value, is transmitted through a pulse-shaping filter to modulate some continuous waveform.

After passing through the baseband channel and the receiving filter, the waveform is being sampled. Ideally, the received samples z_k should be equal to the corresponding x_k or same simple functions thereof. But the samples z_k are perturbed by noise and some neighbouring inputs x_i, $i \neq k$. The latter effect is known as intersymbol interference (ISI).

To eliminate bandwidth-induced ISI, the impulse response h(t) of the transmitting filter $G_T(\omega)$, the channel $C(\omega)$, and the receiving filter $G_R(\omega)$ in cascade must satisfy the Nyquist's first criterion [12]. For h(t) with sample values

$$(h_0, h_1, \ldots h_L) \tag{22}$$

ISI due to the L-th order memory system $G_T(\omega)G_R(\omega)$, with input sequence $X = \{x_i\}$, can be represented by the convolution

$$y_k = \sum_{i=0}^{L} h_i x_{k-i}$$

$$= h_0(x_k + (1/h_0) \cdot \sum_{i=1}^{L} h_i x_{k-i}). \tag{23}$$

The second term is the ISI.

The correlation introduced between successive sample values is of discrete type, in the sense that a data symbol can be disturbed only in a finite number of ways by adjacent symbols. If symbols of a data sequence are so correlated, a method better than symbol-by-symbol decisions is to base decisions on the entire sequence received. Were the data encoded by a convolutional code, we would argue the same. Comparing the

above convolutional form with the convolution for PRS system, we note the similarity in both cases: namely that the sample values $\{h_i\}$ in ISI could correspond to the $\{f_i\}$ of a PRS code. Thus ISI can be regarded as an unintended form of PRS coding.

It is the band limitation of $G_T(\omega)C(\omega)G_R(\omega)$ that introduces memory into the system and stretches the original waveform over a time period longer than the symbol duration resulting in ISI.

The controlled amount of correlation between samples in PRS systems can be used for spectral shaping [8]. This allows practical systems to transmit at the Nyquist rate which is not possible with ordinary PAM systems. Violations of the correlative coding format in PRS systems can be used to monitor error performance or even to correct it [21]. For example, in the ambiquity zone decoding method, the quantizer makes a soft decision including rejection levels. Most of the digits in the ambiquity levels are replaceable with correct values by using the redundancy of the sequence. In addition, a PRS spectrum might be selected to complement a non-ideal channel spectrum in order to reduce the ISI. This idea will be further exploited in this thesis.

The disadvantage of traditional PRS systems lies in using symbol by symbol detection. Reduced noise margin results because the superposition of waveforms causes the number of output levels to be larger and more dense than the number of input levels. By exploiting its spectral shaping property and by using MLSE, we may get improved performance out of PRS systems.

2.8 99% Energy Bandwidth of PRS Systems

For any minimum Nyquist bandwidth PRS systems, the one-sided bandwidth is 1/2T where T is the signalling interval. But we note that for some PRS systems, more energy is concentrated in the low-frequency portion of their spectrum, while others have DC nulls causing energy to be concentrated in the high frequency portions of their band instead. Thus the minimum Nyquist bandwidth of 1/2T of all PRS systems does not convey any information concerning the energy distribution of the coded systems.

Our aim is to obtain the minimum bandwidth required by a PRS code but still be able to signal at a rate of 1/2T symbols/sec with a given amount of ISI. To facilitate this, we define the 99% energy bandwidth of a PRS system as the frequency band within which 99% of the energy of the response is confined.

We choose a model in which the transmitting filter $G_T(\omega)$ is a PRS system and the channel is assumed to be an ideal low-pass filter of single-sided bandwith W. Essentially, one intentionally causes a band limitation with a known spectrum and transmits through a channel that causes no further limiting; that is, no further unintended ISI is introduced by the channel. The premise is that since the ISI is known, being caused by the PRS system, its effect can be removed. We shall consider from now on that the bandwidth of a PRS code is its 99% energy bandwidth.

CHAPTER 3

LINEAR AND NON-LINEAR RECEIVERS

In this chapter, we introduce linear and non-linear receiver structures designed to combat ISI, whether intentional or otherwise. We then show step by step how to formulate the maximum likelihood decoding of partial response signalling systems using the Viterbi Algorithm.

3.1 Linear Receiver Structures

Before we can perform symbol-by-symbol decoding or maximum likelihood sequence estimation, we first have to make the transition from continuous waveform to discrete samples. Simple application of a sampler without a matched filter is information lossy in general. A matched filter receiver is optimum in the sense of maximizing the S/N ratio when there is no ISI [13].

A receiver for synchronous data symbols in the presence of ISI consists of a linear filter, a symbol-rate sampler and a quantiser for establishing symbol-by-symbol decisions. A decoder, possibly with error-detection and/or error correction capability, may follow. The purpose of the receiver filter is to eliminate ISI, at the same time maintaining a high S/N ratio.

For various performance criteria such as minimum mean square error and minimum average error probability [12], [23], the optimum linear

filter can be factored as a matched filter (MF) and a transversal filter (TF) with tap-spacings equal to the sampling interval. The MF sampled outputs are a set of sufficient statistics for estimation of the input sequence, and this filter maximizes the SNR ratio without regard to the residual ISI at its output. The TF eliminates or at least reduces ISI at the expense of enhancing noise and lowering the S/N ratio.

3.2 The Zero-Forcing Equalizer and the Decision Feedback Equalizer

We consider the above filters in a linear space context, which requires that the impulse response of $G_T(\omega)C(\omega)$ mentioned in chapter 2 has finite energy, that is, it is square integrable. Let the impulse response by h(t), thus

$$\int_{-\infty}^{+\infty} h^2(t)\, dt \; < \infty. \tag{24}$$

Define the inner product in this linear space as

$$\langle x, y \rangle \; = \; \int_{-\infty}^{+\infty} x(t)y(t)\, dt, \tag{25}$$

hence,

$$||x||^2 \; = \; \langle x, x \rangle \; . \tag{25}$$

An example of a symbol-by-symbol decisions receiver is the zero-forcing equalizer (ZFE); a ZFE is a filter with impulse response $g_k(t)$, which does not respond to any translate of h(t) except h(t - kT) [6]:

$$\int_{-\infty}^{+\infty} h(t - mT)g_k(t)\, dt = 0, \quad m \neq k, \tag{26}$$

but

$$\int_{-\infty}^{+\infty} h(t - kT)g_k(t)\, dt \neq 0, \tag{27}$$

so that there is an output signal on which to base the decision.

In terms of inner product notation, we have

$$\langle h_k, g_0 \rangle = 0, \qquad k \neq 0, \tag{28}$$

where

$$h_k = h(t - kT)$$

and

$$\langle h_0, g_0 \rangle \neq 0 . \tag{29}$$

Refer to Fig. 3.1.

As noted earlier, in order to exploit the correlation between discrete samples in ISI, sequence estimation rather than symbol-by-symbol decision should be used. One method is to feedback previous symbol decisions and the decision feedback equalizer (DFE) represents the earliest step in this direction; the linear forward filter is allowed to respond to all past data symbols, and the residual interference from past symbols using past decisions is subtracted out prior to the decision threshold. The past decisions may not all be correct, which affects the present decision and causes error propagation. The DFE as a whole is inherently non-linear; only the forward filter consisting of a MF in

cascade with a TF is linear. In inner product form, the forward filter is shown as

$$\langle h_k, g_o \rangle = 0; \qquad k \neq 0 \tag{30}$$

$$\langle h_o, g_o \rangle \neq 0. \tag{31}$$

Refer to Figure 3.2.

Following Messerschmitt's geometric approach to equalization [6], we denote $M(h_k, k > 0)$ as the linear subspace of the past translates of the basic pulse $h(t)$ and $P[h_o, M(h_k, k > 0)]$ as the projection of h_o to the subspace $M(h_k, k > 0)$. The projection is, by definition, the minimum distance between the element of the subspace $M(h_k, k > 0)$ and h_o. The prediction error vector e_o^+ and the interpolation error vector e_o are, respectively,

$$e_o^+ = h_o - P[h_o, M(h_k, k > 0)] \tag{32}$$

and

$$e_o = h_o - P[h_o, m(h_k, k \neq 0)], \tag{33}$$

See Figure 3.3.

The necessary and sufficient conditions for DFE (ZFE) to exist are that e_o^+ (e_o) must be positive. Physically, this means that $h(t)$ must not be representable as an infinite weighted sum of a subset of its own translates.

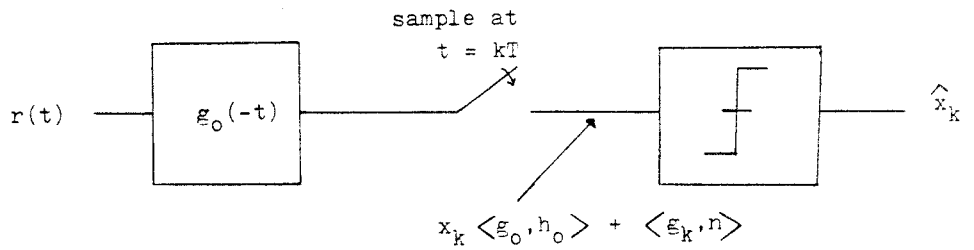


Fig. 3.1 A zero-forcing equalizer

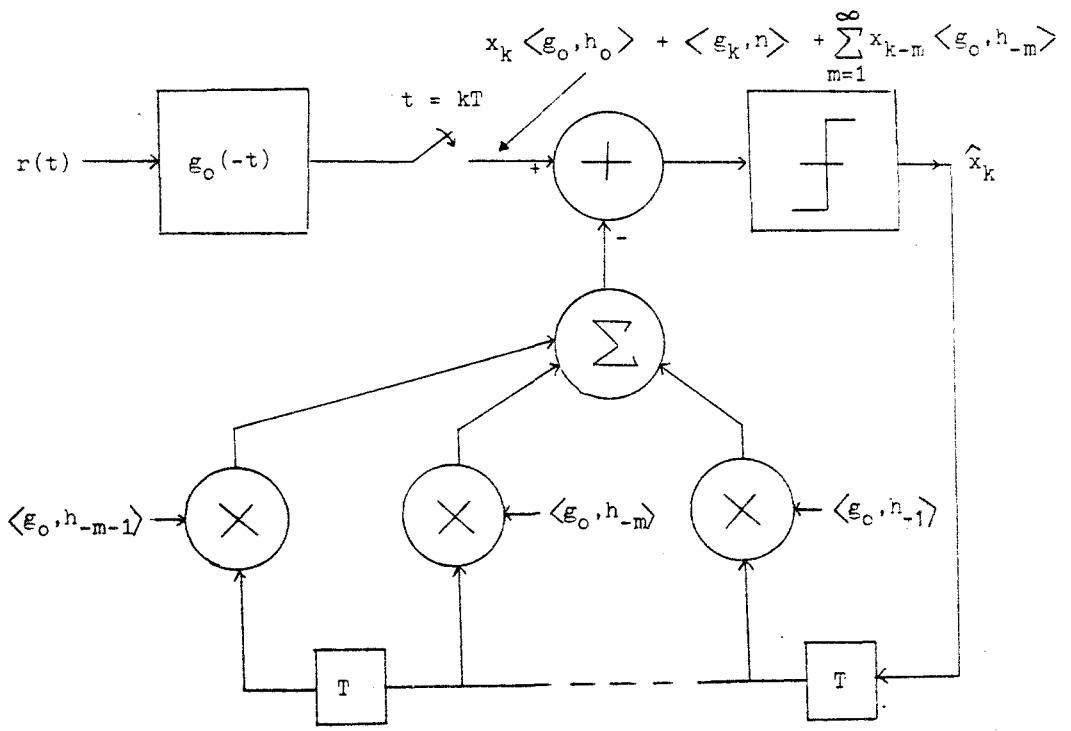


Fig. 3.2 A decision-feedback equalizer

3.3 Performances of DFE and ZFE

Let us consider the reception of

$$r(t) = \sum_k x_k h(t - kT) + n(t)$$

by a DFE represented by g_o. The output of the filter is

$$\langle g_o, r\rangle = x_k\langle g_o, h_o\rangle + \langle g_o, n\rangle. \tag{34}$$

For additive white Gaussian noise (AWGN) with zero mean, the variance of $\langle g_o, n\rangle$ is

$$E[n_o^2] = N_o/2 \cdot ||g_o||^2, \tag{35}$$

with N_o being the one-sided power spectral density of AWGN. The probability of symbol error is a monotonically decreasing function of S/N ratio

$$\langle g_o, h_o\rangle^2 / ||g_o||^2. \tag{36}$$

For DFE, $\langle h_o, g_o\rangle \neq 0$ and $\langle h_k, g_o\rangle = 0$ for $k > 0$; the latter equation means that g_o is orthogonal to $M(h_k, k > 0)$. Hence, g_o is orthogonal to $P[h_o, M(h_k, k > 0)]$. By definition, we have $h_o = e_o^+ + P[h_o, M(h_k, k > 0)]$ and the SNR ratio becomes

$$\langle g_o, (e_o^+ + P[h_o, M(h_k, k > 0)]\ )\rangle^2/||g_o||^2$$

$$= \langle g_o, e_o^+\rangle^2/||g_o||^2,$$

as $\langle g_o, P[h_o, M(H_k, k > 0)]\rangle = 0$. Using the Schwarz inequality, we have

$$\text{S/N ratio} \overset{\leq}{=} ||e_o^+||^2 , \tag{37}$$

with equality if and only if $g_o = ke_o^+$, where k is a multiplicative constant.

The above derivation is true for ZFE also. In the DFE case, it is assumed that the decision feedback mechanism correctly eliminates the ISI.

From eq. (32) and (33), we note that the prediction error vector e_o^+ and the interpolation error e_o can both be written in the form of a convergent sum of translates of h_o [6]; i.e.,

$$e_o^+ = h_o - \sum_{k>o} a_k^+ h_k \tag{38}$$

and

$$e_o = h_o - \sum_{k\neq o} a_k h_k , \tag{39}$$

where a_k^+, a_k are the appropriate coefficients. It is clear that the two vectors can be represented as a matched filter matched to h_o followed by a transversal filter. Using the results of eq. (37), we note that the DFE and ZFE that maximizes the S/N ratio is a MF followed by a TF in the presence of AWGN.

3.4 Non-Linear Receiver Structure

Receivers that truly perform sequence decisions or exploit the discreteness of ISI, exhibit highly non-linear structures. Forney devised a new receiver structure consisting of a MF followed by a TF, a

symbol-rate sampler and a recursive non-linear processor known as the Viterbi Algorithm (VA) [5]. The sampled outputs of the MF provides a set of sufficient statistics for the estimation of input sequences. Whitening of the noise is essential because the VA requires that the noise samples be statiscally independent. This can be provided by a TF characterized by $1/F(D^{-1})$ with $R(D) = F(D)F(D^{-1})$. Accordingly, $\sigma^2 R(D)$ is the autocorrelation function of zero mean colored Gaussian noise due to the channel. This decorrelating property of a MF and a TF (MFTF) follows from the fact that the successive least mean square prediction errors of a random process are uncorrelated random variables [6]. Letting $e_k^+ = e_o^+(t - kT)$, the noise sequence at the output of MFTF is $\langle e_k^+, n \rangle$. The noise terms in the sequence will be uncorrelated if and only if

$$\langle e_k^+, e_j^+ \rangle = 0; \quad k \neq j \tag{40}$$

Note that e_k^+ is orthogonal to $M(h_m, m > k)$ but $M(h_m, m > k)$ contains e_j^+ for $j > k$, thus $\langle e_j^+, e_k^+ \rangle = 0$ for $j > k$. By symmtry, $\langle e_j^+, e_k^+ \rangle = 0$ for $k < j$. Hence, eq. (40) is satisfied. The MFTF of the DFE forward filter is idential to the "whitened matched filter" employed by Forney for MLSE.

As stressed before, a band limited channel may be simulated by a PRS code in the sense that a PRS code can be used as the transmitting filter that band limits the signal before the channel. In effect, the channel causes no further ISI. Assuming this, let the system polynomial of the transmitting filter be $F(D) = \sum_{i=0}^{L} f_i D^i$.

Thus the unit impulse is

$$h(t) = \sum_{i=0}^{L} f_i \, \mathrm{sinc}(t - iT), \tag{41}$$

where L is the number of T-delay units.

Define $R(k) = \langle h_m, h_{m+k} \rangle$ with $h_m = h(t - mT)$.

Thus,

$$R(k) = \left(\sum_{i=0}^{L} f_i \, \phi_{m-i} \right) \cdot \sum_{J=0}^{L} f_j \, \phi_{m+k-j}$$

$$= \sum_{i=0}^{L} \sum_{j=0}^{L} f_i f_j \phi_{m-i} \, \phi_{m+k-j}$$

$$= \sum_{i=0}^{L} f_i f_{i+k}, \quad L < k < L, \tag{42}$$

zero otherwise due to the orthonormality of ϕ_ℓ, where

$\phi_\ell = \mathrm{sinc}(t/T - \ell)$ and ℓ = integer.

The energy of the impulse response of the filter is

$$R(0) = \sum_{i=0}^{L} f_i^2 \tag{43}$$

From eq. (15) the received signal at the input of the whitened matched filter is

$$z(t) = \sum_{k=0}^{\infty} x_k h(t - kT) + n(t) \tag{44}$$

The signal at the output of the whitened matched filter with $1/F(D^{-1})$ being the transversal digital filter is

$$z_k = f_o x_k + \sum_{m=1}^{\infty} f_m x_{k-m} + n_k , \qquad (45)$$

where n_k is an i.i.d. Gaussian random variable with variance σ^2 (i.i.d. denotes independent and identically distributed). See Fig. 3.6

The DFE forms the quantity

$$z_k' = z_k - \sum_{m=1}^{\infty} f_m \hat{x}_{k-m} , \qquad (46)$$

and applies it to the decision threshold set at $\pm f_o$ to determine the estimated symbol $\hat{x}_k$; while the maximum likelihood sequence estimator assumes the sum in eq. (45) is truncated to M terms and determines the estimated sequence $\{\hat{x}_k\}$ so as to minimize

$$\sum_{k=1}^{N} \{z_k - \sum_{m=0}^{M} f_m \hat{x}_{k-m}\} , \qquad (47)$$

for any integer N. Thus the two receivers perform similar functions on the same sufficient statistics $\{z_k\}$; the difference being that the DFE extends a single sequence while the MLSE exhaustively searches over all allowable sequences and selects the one which is closest to the received sequence in the N-dimensional Euclidean space.

3.5 Maximum Likelihood Sequence Estimation for PRS Systems

For PRS systems, the maps from input sequences X(D) to state sequences S(D) then to signal sequences Y(D) are one to one. To show that the eq. (47) is true, we define the maximum likelihood sequence

estimation (MLSE) of

$$Z(D) = \sum_{k=0}^{\infty} z_k D^k \tag{48}$$

as maximizing p(Z(D)/X(D)), where p(.) denotes the probability density function.

Maximizing p(Z(D)/X(D)) is the same as maximizing ln(p(Z(D)/Y(D))), as ln(.) is a monotonically decreasing function. This in turn is equal to maximizing $\ln \prod_k p_n(z_k - y_k)$ where $P_n(.)$ is the probability density function of AWGN samples n_k; y_k and n_k are statistical independent. The above maximization is equivalent to maximizing

$$\sum_k \ln p_n(z_k - y_k),$$

is equivalent to maximizing

$$\sum_k \ln \left[1/\sqrt{2\pi\sigma^2} \cdot e^{-(z_k - y_k)^2/2\sigma^2}\right].$$

is equivalent to maximizing

$$\sum_k \left[- 1/2 \cdot \ln 2\pi\sigma^2 - (z_k - y_k)^2/2\sigma^2\right],$$

is equivalent to maximizing

$$\sum_k - (z_k - y_k)^2/2^{\ 2},$$

as the 1st term is a constant. Thus, it amounts to the minimization of

$$\sum_k (z_k - y_k)^2.$$

It can be shown [18] that the MLSE rule is equivalent to the MAP rule for infinite SNR ratio. The MAP rule offers a significant advantage only at low S/N ratio.

3.6 Formulation of Viterbi Algorithm for PRS Systems

From eq. (47) and eq. (49), we note that Y(D) will be chosen that is nearest in terms of squared Euclidean distance to the received sequence Z(D) in maximum likelihood sequence estimation. In formulating the VA, we will exploit the tree and trellis structure of the real number convolutional codes. Any convolutional code can be represented as a tree, but if two nodes at the same level of the tree represent the same state then they can be merged to a single node, since they will produce the same output sequence for the same future input sequence. Thus, it should be represented as a collapsed tree Forney called a trellis. It shows the state transitions versus time for all distinct states. It has the important property that to every possible state sequence S(D), there corresponds a unique path through the trellis and vice-versa. We associate with every branch of a trellis at depth K a quantity $(z_k - y_k)^2$ of eq. (49). Every path of N depth will have a Euclidean distance of $\sum_{k=1}^{N} (z_k - y_k)^2$. A typical section in a trellis refers to the section where m^k transitions occur. The first typical section of a trellis starts at depth K and ends at depth N-k for a trellis of length N. See Fig. 3.4.

For a trellis of depth N, we have m^N possible state sequences for m-ary inputs to the PRS system. A brute force method of finding the minimum distance path requires the computation of the distance of m^N paths followed by $m^N - 1$ binary comparisons.

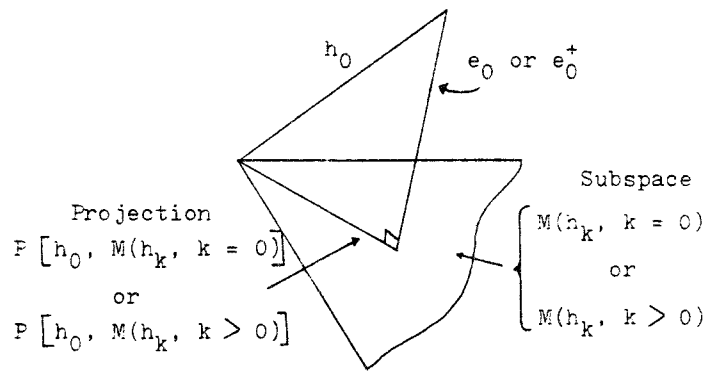


Fig. 3.3 Geometric interpretation of the ZFE and DFE

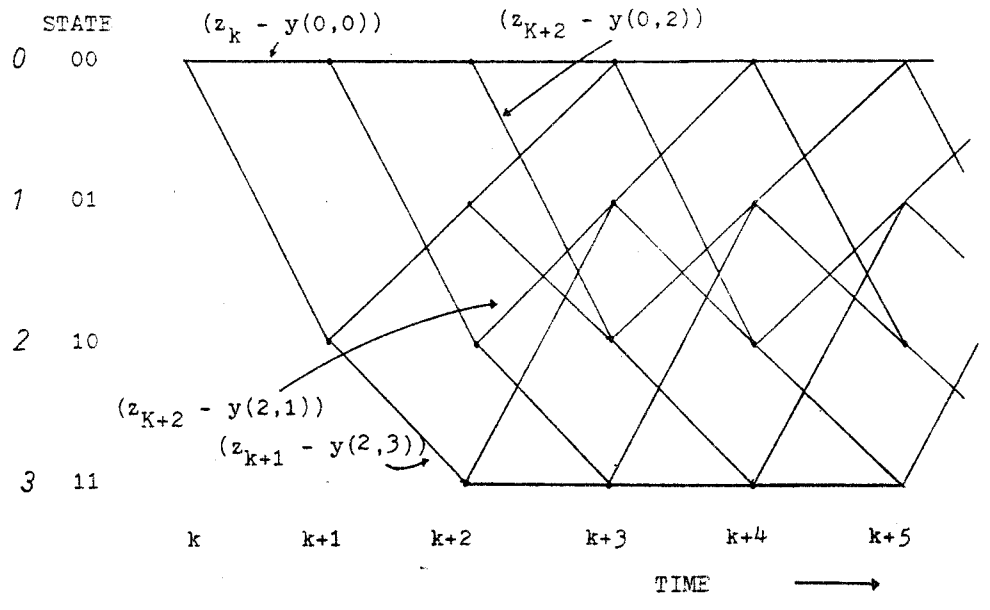


Fig. 3.4 Metric assignment for a state trellis of a two delay units PRS system in Viterbi decoding

The VA is a dynamic programming approach, that compares the distances of all possible paths through a trellis in an optimum way, to minimize the number of operations, and to select the path which has the minimum distance. Define the optimal value function $F_n(u)$ as the minimum squared Euclidean distance from time t = 0 to t = n between any signal sequence $Y_i(D)$ and the received sequence Z(D) terminating in state u. Thus,

$$F_n(u) = \min\{ h_i^2 : h_i^2 = \min ||Z(D) - Y_i(D)||^2 \quad \text{for each } 1 \leq i \leq m^n$$

$$\text{such that } s_{i_n} = u\}. \tag{50}$$

The state sequence $S_j(D)$ is known as the survivor at time n for state u. s_{i_n} denotes the state of state sequence $S_i(D)$ at time n.

Using the theorem of optimality in dynamic programming, we assert that of all the m^L survivors at every depth, one for each state, at least one will constitute the initial segment of the state sequence denoted by

$$S(D) = \sum_{i=0}^{N} s_i D^i \tag{51}$$

which minimizes the squared Euclidean distance. For if S(D) does not contain any survivors at a given depth n in state u then we can replace its initial segment by $S_j(D)$, a survivor at time n in state u, to obtain a shorter squared Euclidean distance.

The above assertion allows us to act as follows: for a trellis of depth N, instead of choosing one path through the trellis that gives

the minimum squared distance, we decompose the task into N choices. For every depth $n \geq K$ and for every state u in S, we select among the m paths terminating at a given state u, a path that gives the minimum squared Euclidean distance as the survivor. The recurrence relation for the optimal value function is thus

$$F_{n+1}(u) = \min\{ L^2(v,u) + F_n(v): \text{ for every v in S such that } \delta(v,x_n) = u, x_n \text{ in X}\}. \quad (52)$$

Refer to section 2.5. The incremental length $L^2(v,u)$ is computed as follows

$$L^2(v,u) = (z_k - y(\alpha_k))^2, \quad \text{where the transition } \alpha_k = (v,u),$$
$$= (z_k - \lambda(v, x_k))^2, \text{ where the state } u = \delta(v, x_k) \quad (53)$$

The state transition function $\delta(v, x_k)$ gives the state at t = k + 1, given the present state v and the present input x_k.

From eq. (6), a state is denoted by

$$s_k = (x_{k-1}, x_{k-2}, \ldots x_{k-L}).$$

In m-ary representation, s_k is equal to the integer

$$x_{k-1}\, m^{L-1} + x_{k-2}\, m^{L-2} + \ldots\ldots x_{k-L}\,. \quad (54)$$

Similarly, s_{k+1} is denoted by

$$(x_k, x_{k-1}, \ldots x_{k+1-L}),$$

which in m-ary notation is equal to the integer

$$x_k m^{L-1} + x_{k-1} m^{L-2} + \ldots\ldots x_{k+1-L} \tag{55}$$

By comparing eq. (54) and (55), the integer representing s_{k+1} is equal to

$$s_{k+1} = \delta(s_k, x_k) = x_k m^{L-1} + \underline{\lfloor s_k/m \rfloor}, \tag{56}$$

where $\underline{\lfloor\ \rfloor}$ denotes modulo division. Thus, the δ function is a function of the present state and input:

$$\delta(v, i) = i m^{L-1} + \underline{\lfloor v/m \rfloor}, \tag{57}$$

for all $v \in S$, $i \in X$. See Fig. 3.5.

The recurrence relation of eq. (52) is in the form of the backward dynamic programming approach [11]. The formulation utilizes the state transition function which is an implicit function of state u. We define a function $T: S \times P \to S$, which explicitly gives all states v at time n merging to state u at time n+1. Define P as the set of data symbols that shift out from the digital transversal filter in the transition from $t = n$ to $t = n+1$, due to various input data symbols $x_n \in X$. Thus,

$$P = \{0, 1, \ldots\ldots m-1\} .$$

Consider s_{k-1}, denoted by

$$(x_{k-2}, x_{k-3}, \ldots\ldots x_{k-L-1}),$$

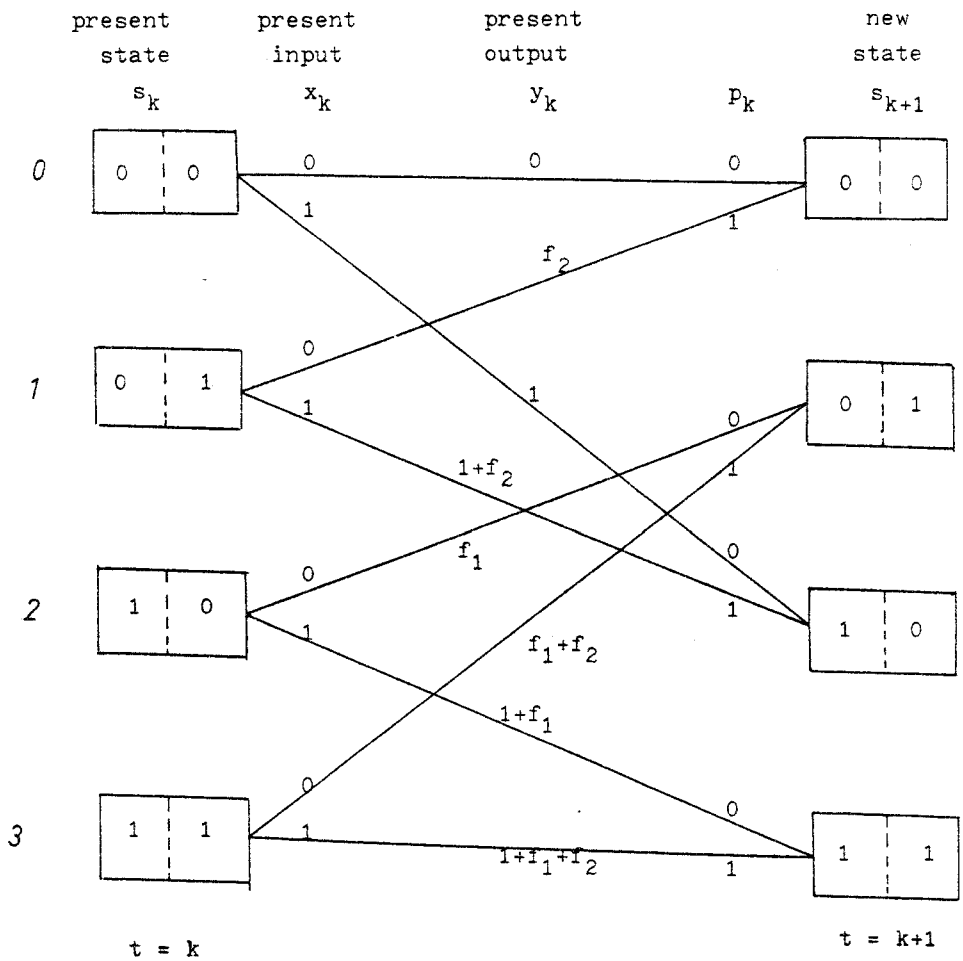
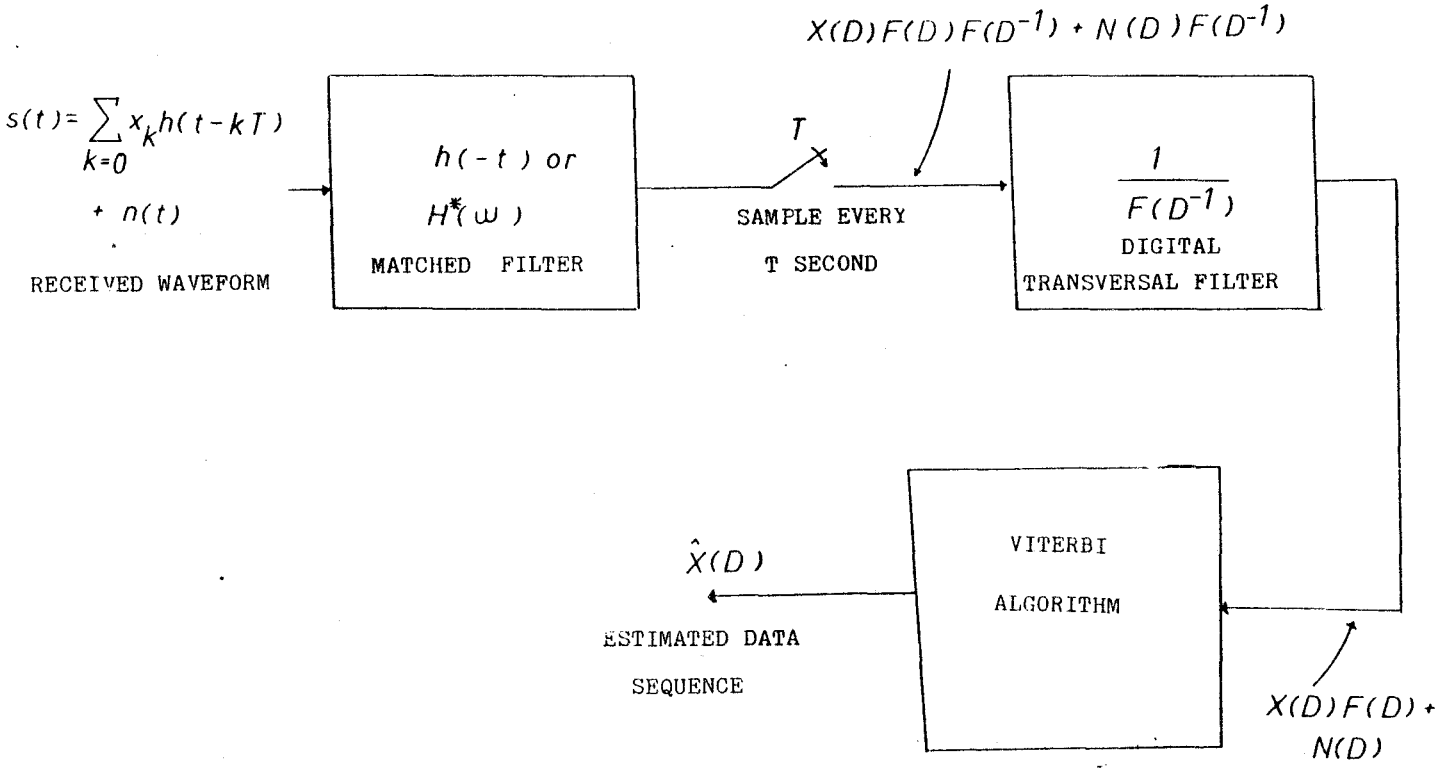


Fig. 3.5 A typical section of a state trellis of a two delays PRS system showing the present state s_k, the present input x_k at time t = k and the present output y_k in the transition to state s_{k+1} at time k + 1. p_k is the data bit that shifted out of the shift-register in the transition (s_k, s_{k+1}).



N(D) --- STATISTICALLY INDEPENDENT IDENTICALLY DISTRIBUTED NOISE SAMPLES

X(D) --- DATA SEQUENCE

Fig. 3.6 The model of a maximum likelihood receiver for transmitting filter using partial response signalling system of Fig. 2.4.

and equal to the integer

$$x_{k-2}\, m^{L-1} + x_{k-3}\, m^{L-2} + \cdots x_{k-L-1} \,. \tag{58}$$

From eq. (54) we have

$$x_{k-1} = \lfloor s_k / m^{L-1} \rfloor .$$

Hence,

$$x_{k-1} m^{L-1} = \lfloor s_k / m^{L-1} \rfloor \, m^{L-1} \,.$$

As $\quad x_{k-2} m^{L-2} + \cdots + x_{k-L} = s_k - x_{k-1} m^{L-1}$,

so $\quad x_{k-2} m^{L-1} + \cdots + x_{k-L} m = m(s_k - x_{k-1} m^{L-1})$,

therefore,

$$s_{k-1} = m(s_k - \lfloor s_k / m^{L-1} \rfloor \, m^{L-1}) + x_{k-L-1} . \tag{59}$$

Consequently, the T function is given by

$$T(u, p) = m(u - \lfloor u/m^{L-1} \rfloor \, m^{L-1}) + p \,. \tag{60}$$

The recurrence relation of eq. (52) becomes

$$F_{n+1}(u) = \min\{ \lfloor^2(v,u) + F_n(v): \text{ for each } p \text{ in } P \text{ such that}$$

$$v = T(u,p)\} \,. \tag{61}$$

This is an explicit form for computing the minimum squared Euclidean distance corresponding to the survivor at time n + 1 for state u.

3.7 Viterbi Algorithm

A functional form of the VA using structured programming is now given.

/COMMENT/ Set boundary condition for the non-typical sections of the trellis choose any $s_0 \in S$.

For each $u \in S$ do

if $u = s_0$ then $F_0(u) \leftarrow 0$ else $F_0(u) \leftarrow \infty$;

/COMMENT/ Trellis of depth N

For i = 1 step 1 until N do

for each $u \in S$ do

begin

for each $p \in P$ do

/COMMENT/ To get the survivor for each state at every depth

$$\text{Temp}(p) \leftarrow F_n(T(u,p)) + \lambda^2(T(u,p),u) \ ;$$

$$F_{n+1}(u) \leftarrow \min \{\text{Temp}(p)\}$$

end;

From Eq. (53), the values of $\lambda^2(.....)$ can be computed by a subtraction and a squaring operation, after getting the corresponding values of $\lambda(.,.)$ From eq. (15), we note that there are m^K values for $\lambda(.,.)$ and they constitute the set of all output levels. An efficient procedure is to calculate the above set and store them in random access memory for table look-up whenever needed. The depth N of the trellis required depends on the decision depth of the particular PRS code. The decision depth will be defined in Chapter 4.

The complexity of the VA is estimated as follows: in terms of memory, m^{L+1} locations are required to store the output levels of all transitions in a typical section. It requires m^{L} locations to store the optimal value functions $F_n(u)$. As $F_n(u)$ is computed by recursions, in practice twice the amount of memory is needed: m^{L} locations are used to store the previous optimal value functions $F_{n-1}(u)$ and another m^{L} locations are needed for the present $F_n(u)$.

A state sequence of input sequence for each survivor at every depth has to be stored. In terms of computation, in each symbol period, there are m^{L+1} operations each involving a subtraction, a squaring, and an addition followed by $(m - 1)m^{L}$ binary comparisons.

CHAPTER 4

DOUBLE DYNAMIC PROGRAMMING

We have sacrificed simplicity in using non-linear processors such as the Viterbi Algorithm in estimating the received data. We expect better performances in terms of both SNR ratio and probability of symbol error in return. Before we can precisely estimate these performances, the idea of error events is necessary. After introducing this concept, we will show how to compare the performances of the linear and non-linear receivers discussed in Chapter 3. For this we need the minimum squared Euclidean distances, which in turn require "two-dimensional" dynamic programming for their evaluation. The latter part of this chapter shows the step-by-step process in the formulation of the "two-dimensional" dynamic programming. Finally, its implementation in the form of structure programming is listed.

4.1 Error Event Concept

Following Forney's definition [4], an error event is said to extend from time k_1 to k_2 if the estimated state sequence S(D) is equal to the correct state sequence S(D) at times k_1 and k_2, but none in between; i.e.:

$$s_{k_1} = \hat{s}_{k_1}, \quad s_{k_2} = \hat{s}_{k_2}$$

and

$$s_k \neq \hat{s}_k , \qquad k_1 < k < k_2$$

The length of the error event is defined as: $n \triangleq k_2 - k_1 - 1$.

For a linear channel, $Y(D) = X(D)F(D)$. Consider an error event in a linear channel of memory L; since a state is denoted as

$$s_k = (x_{k-1}, \ldots . x_{k-L}),$$

consequently,

$$\text{if } s_{k_1} = \hat{s}_{k_1} \quad \text{then} \quad x_k = \hat{x}_k ; \; k_1 - L < k < k_1 - 1.$$

Similarly,

$$\text{if } s_{k_2} = \hat{s}_{k_2} \quad \text{then} \quad x_k = \hat{x}_k ; \; k_2 - L < k < k_2 - 1.$$

As $s_{k_1+1} \neq \hat{s}_{k_1+1}$,

therefore

$$x_{k_1} \neq \hat{x}_{k_1} .$$

Similarly, since $s_{k_2-1} \neq \hat{s}_{k_2-1}$

therefore $x_{k_2-L-1} \neq \hat{x}_{k_2-L-1}$.

Mathematically, the input error sequence $\epsilon_x(D)$ can be defined as:

$$\mathcal{E}_x(D) \triangleq (x_{k_1} - \hat{x}_{k_1}) + (x_{k_1+1} - \hat{x}_{k_1+1})D + \ldots.$$

$$+ (x_{k_2-L-1} - \hat{x}_{k_2-L-1})D^{n-L}$$

$$= e_{x_o} + e_{x_1}D + \ldots. + e_{x_{n-L}}D^{n-L} \tag{62}$$

It is a polynomial with non-zero constant term and degree n-L. No L consecutive zero coefficients can appear in an input error sequence, since then $s_k = \hat{s}_k$ for some intermediate k, and there would be two distinct error events instead of one.

In the same way, the signal error sequence associated with the error event is defined as

$$\mathcal{E}_y(D) \triangleq (y_{k_1} - \hat{y}_{k_1}) + (y_{k_1+1} - \hat{y}_{k_1+1})D + \ldots.$$

$$+ (y_{k_2-1} - \hat{y}_{k_2-1})D^n. \tag{63}$$

It has non-zero constant term and degree n, where n is the length of the error event. As $Y(D) = X(D)F(D)$ and $\hat{Y}(D) = \hat{X}(D)F(D)$, therefore

$$\mathcal{E}_y(D) = \mathcal{E}_x(D)F(D) \tag{64}$$

4.2 Euclidean Weight of A Particular Error Event

The Euclidean weight $d^2(\mathcal{E})$ of an error event is defined as the energy in the associated signal error sequence.

$$d^2(\epsilon) \triangleq ||\epsilon_y(D)||^2$$

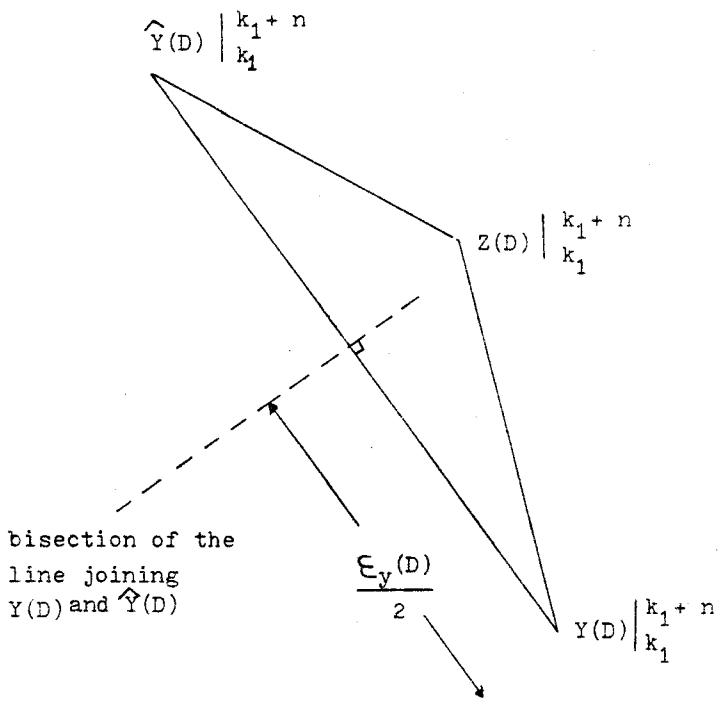
$$= [\epsilon_y(D)\epsilon_y(D^{-1})]_{D=0}$$

$$= [\epsilon_x(D)F(D)F(D^{-1})\epsilon_x(D^{-1})]_{D=0}$$

$$= [\epsilon_x(D)R(D)\epsilon_x(D^{-1})]_{D=0} \qquad (65)$$

where $[.]_{D=0}$ denotes the constant term of the polynomial. The energy depends on $\epsilon_x(D)$ and R(D), where R(D) is the autocorrelation function polynomial of F(D), and is independent of the factorization $R(D) = F(D)F(D^{-1})$. Essentially, the Euclidean weight $d^2(\epsilon)$ is identifiable as the energy of the signal error sequence after passing the input error sequence $\epsilon_x(D)$ through a PRS system F(D). The number of errors in the input error sequence $\epsilon_x(D)$ is defined as the Hamming weight $w_H(\epsilon)$ of the event.

The error event ϵ is due to the fact that in the n + 1 dimensional Euclidean space corresponding to times k_1 to $k_1 + n$, $\hat{Y}(D)$ is closer to the received sequence Z(D) than is Y(D). Since the noise is additive, white, and Gaussian with equal variance in all dimensions, it is spherically symmetric; by coordinate rotation, we see that the probability of an error event $P(\epsilon)$ is dominated by the probability that a single Gaussian random variable of variance σ^2 exceeds half the Euclidean distance between Y(D) and $\hat{Y}(D)$ [13]. Since $[Y(D) - \hat{Y}(D)]_{k_1}^{k_1+n} = \epsilon_y(D)$, this distance squared is the Euclidean weight $d^2(\epsilon)$ of the error event. See Fig. 4.1.



$$\varepsilon_y(D) = \left[Y(D) - \hat{Y}(D) \right]_{k_1}^{k_1+n}$$

$$P(\varepsilon) = Q(\varepsilon_y(D)/2\sigma)$$

Fig. 4.1 The plane containing the three points Y(D), $\hat{Y}(D)$ and Z(D) in the n-dimensional signal space.

4.3 Probability of Symbol Error

Following the analysis of Forney [4], let E be the set of all possible error events ϵ starting at time k_1. Using the union bound, which states that the probability of a union of events is less than or equal to the sum of their individual probabilities, the probability of an error event starting at time k_1 can be bounded as

$$P(E) \leq \sum_{\epsilon \in E} P(\epsilon). \tag{66}$$

Let D be the set of all possible distances $d(\epsilon)$, and for each $d \in D$, let E_d be the subset of all error events for which $d(\epsilon) = d$. Let P_1 be the probability that between k_1 and $k_1 + n - L$, the input sequence X(D) will be such that $X(D) + \epsilon_x(D)$ is an allowable sequence. Then the probability that a particular error event starting at time k_1 with Euclidean weight d is bounded as

$$P(\epsilon) \leq Q(d(\epsilon)/2\sigma)P_1. \tag{67}$$

Accordingly,

$$P(E) \leq \sum_{\epsilon \in E} Q(d(\epsilon)/2\sigma)P_1$$

$$\leq \sum_{d \in D} Q(d/2\sigma) \sum_{\epsilon \in E_d} P_1. \tag{68}$$

Due to the exponential decrease of the Gaussian distribution function Q(.) with d, the P(E) will be dominated by the free distance d_{free} of $E_{d_{free}}$ at moderate SNR ratio; i.e.

$$P(E) \simeq K_1 Q(d_{free}/2\sigma), \tag{69}$$

where $$K_1 = \sum_{\epsilon \in E_{d_{free}}} P_1 .$$

As K_1 is independent of the noise variance σ^2, and eq. (69) is independent of the starting time k_1, P(E) can be considered as the probability of an error event per unit time.

The probability of symbol error is computed by weighting each error event by the number of decision errors, that is, the Hamming weights $w_H(\epsilon)$ of the error event. For example, an input error sequence $\epsilon_x(D) = 1 + D$ will have Hamming weights of 2. The probability of symbol error of an error event is

$$\begin{aligned} P(e) &\leq \sum_{\epsilon \in E} w_H(\epsilon) P(\epsilon) \\ &\leq \sum_{\epsilon \in E_d} w_H(\epsilon) P_1 \sum_{d \in D} Q(d/2\sigma) \\ &\approx K_2 Q(d_{free}/2\sigma) , \end{aligned} \tag{70}$$

with

$$K_2 = \sum_{\epsilon \in E_{d_{free}}} w_H(\epsilon) P_1 .$$

Forney went on to show that both P(E) and P(e) are lower bounded by

$$K_0 Q(d_{free}/2\sigma), \tag{71}$$

where $K_0 \leq 1$ is the probability that the input sequence X(D) will be such that $\hat{X}(D) = X(D) + D^k \epsilon_x(D)$ is also an allowable input sequence for at least one $\epsilon_x(D) \in E_{d_{free}}$ starting at time k. For example, when $d_{free}^2 = ||F(D)||^2$, $E_{d_{free}}$ contains only $\epsilon_x(D) = \pm 1$ and $K_0 = 1$. K is the coefficient for the lower bound

for both the probability of symbol error and the probability of an error event per unit time. For the upper estimate, Forney gets different constants: K_1 for the probability of error event per unit time and K_2 for probability of symbol error. The quantity K_2/K_1 may be regarded as the average number of symbol errors per error event at high S/N ratio.

4.4 Performance of Viterbi Algorithm in the Presence of ISI

In the absence of ISI, $x_k = 0$ for $k \neq 0$; the signal sequence is of the form $Y(D) = x_0 F(D)$ and the probability of symbol error for m-ary input is approximately equal to

$$P(e) \simeq K_3 Q(||F(D)||/2\sigma), \tag{72}$$

where

$$K_3 = \sum_{\in \varepsilon} E_{d_{free}} w_H(\in) P_1$$

$$= \sum_{\in \varepsilon} E_{d_{free}} (m-1)/m = 2(m-1)/m,$$

since $w_H(\in) = 1$ and $P_1 = (m-1)/m$.

Defining the effective S/N ratio as

$$SNR_{eff} \triangleq \sigma_x^2 d_{free}^2/\sigma^2 \tag{73}$$

with $\sigma_x^2 = (m^2 - 1)/12$ [Appendix A.1]

and $\sigma^2 = N_0/2$,

We note that the probability of symbol error of an m-level PAM system with

ISI differs from that of the same system without ISI by the ratio K_2/K_3, when Viterbi Algorithm is used. In decibels such a difference is small and goes to zero under most practically attainable SNR_{eff}. For correlative codes in which $d^2_{free} = ||F(D)||^2$ under no ISI condition, $SNR = SNR_{eff}$ and the degradation is negligible.

4.5 Performance of MLSE Against DFE

We shall now see that the performance of the Viterbi Algorithm for a given channel or, equivalently, for a PRS code depends only on the quantity d^2_{free}, the minimum Euclidean weight of any possible error event.

Following Messerschmitt [7], we can demonstrate a close relationship between MLSE and DFE by using d^2_{free}, and show that the performance of the former is always better. Consider, for simplicity, binary signalling and error events due to all possible input error sequences $\epsilon_x(D) = e_{x_0} + e_{x_1}D + \ldots\ldots + e_{x_n}D^n$; for all n. Define

$$d^2_{free} \triangleq \inf_{e_{x_0}\neq 0} || \sum_{k=0}^{n} e_{x_k} h_k||^2 \quad \text{with} \quad e_{x_k} \in \{1, -1, 0\} \tag{74}$$

Thus, d^2_{free} is the minimum Euclidean weight for all input error sequences including those of infinite length. We can rewrite this as

$$d^2_{free} = \inf||h_0 + \sum_{k=0}^{n} e_{x_k} h_k||^2 . \tag{75}$$

Note that $\sum_{k=1}^{n} e_{x_k} h_k \in M(h_k, k > 1)$, and the minimisation in eq. (75) is

to find the element of $M(h_k, k \geq 1)$ with restricted coefficients $\{1, -1, 0\}$ which is closest to h_o. The closest element without the restriction in coefficients is the projection of h_o on $M(h_k, k \geq 1)$; i.e., $P(h_o, M(h_k, k \geq 1))$. Thus d^2_{free} is determined by how closely the projection can be approximated by an element with restricted coefficients. From eq. (32), we have

$$h_o = e_o^+ + P(h_o, M(h_k, k \geq 1)),$$

hence

$$d^2_{free} = \inf \| e_o^+ + P(h_o, M(h_k, k \geq 1)) + \sum_{k=1}^{n} e_{x_k} h_k \|^2$$

$$= \|e_o^+\|^2 + \inf \|P(h_o, M(h_k, k \geq 1)) + \sum_{k=1}^{n} e_{x_k} h_k \|^2 , \quad (76)$$

as e_o^+ is orthogonal to $M(h_k, k > 0)$ and $[P(h_o, M(h_k, k \geq 1)) + \sum_{k=1}^{n} e_{x_k} h_k]$ is in $M(h_k, k \geq 0)$. The result then follows from Pythagarus theorem.

$d^2_{free} > \|e_o^+\|^2$ implies that the performance of MLSE in terms of both symbol error probability or SNR_{eff} always exceeds that of DFE. We conclude that the actual amount of difference in performance is governed by the coarseness of the best approximation by an element with restricted coefficients to the projection: the poorer the approximation, the better the performance of the MLSE.

4.6 Double Dynamic Programming Formulation

We are concerned with finding the d^2_{free} of a correlative-level code. A PRS code is characterized by the tap gains $\{f_i\}$ or $F(D)$, the

impulse response. A codeword is any allowable output sequence of the above code.

Flow-graph techniques developed by Viterbi [9] give an organized way of considering all the possible input error sequences in computing the resultant Euclidean weights. This method provides a means of finding the number of error events producing the same Euclidean weights $d^2(\in)$, and gives asymptotically tight upper and lower bounds on probability of symbol error. But this method is difficult to computerize and time consuming.

We apply dynamic programming to find the d^2_{free} and a related quantity, the decision depth of PRS codes, and we neglect the total number of error events that give rise to the same d^2_{free}. These PRS codes can be regarded as real-number convolutional codes as opposed to binary convolutional codes. In the former, the arithmetic operations are in the real-number field with redundancy introduced amplitude-wise, while the latter has operations in the Galois' field and redundancy introduced time-wise. One big difference is the group property possessed by binary convolutional codes [9]: we can choose any specific codeword and compute the set of metrics $\{d_i\}$ from this given codeword to all others; this set of metrics is the same no matter what is the selected codeword, consequently, it is possible to find the d_{free} merely by minimizing over this set alone. However, lacking this group property for correlative-level codes, we must look at the set of metrics for each codeword.

In order to find the free distance in a real number convolutional code, first, we have to compare a given allowable output sequence with all the other allowable output sequences and select the pair that has the

minimum Euclidean distance. This procedure has to be repeated for each and every allowable output sequence. Furthermore, we need to find the minimum of all these minimum distances to get the free distance d_{free}.

To put the free distance concept in to precise mathematical form:

$$d_{free}^2 \triangleq \min_i \{h_i^2 : h_i^2 = \min_j \{||Y_i(D) - Y_j(D)||^2, \text{ for each } j \neq i, \ 1 \leq j \leq m^N\}, \text{ for each } 1 \leq i \leq m^N, K \leq N \leq \infty\}, \tag{77}$$

where $$||Y_i(D) - Y_j(D)||^2 = \sum_{k=1}^{N} (y_{i_k} - y_{j_k})^2.$$

A brute-force where method involves repeating the VA search m^N times followed by $(m^N - 1)$ binary comparisons. To avoid this, we apply dynamic programming again to the "dimension" of all possible codewords; we term this joint dynamic program as "double dynamic programming".

Define the optimal value function $F_n(u, v)$ as the minimum squared Euclidean distance from time t = 0 to t = n between any pair of codewords or signal sequences, such that one is in state u and the other in state v, subject to the boundary condition $F_0(a, a) = 0$ for any $a \in S$

$$F_n(u, v) \triangleq \min_i \{h_i^2 : h_i^2 = \min \{||Y_i(D) - Y_j(D)||^2 \text{ for each } 1 \leq j \leq m^n, \text{ such that } s_{j_n} = v\}, \text{ for each } 1 \leq i \leq m^n, \text{ such that } s_{i_n} = u\} \tag{78}$$

Using backward dynamic programming, F_{n+1} is calculated from F_n by the recurrence formula:

$$F_{n+1}(u, v) = \min\{ F_n(p, q) + \Delta E(p, u), (q, v)): \text{ for each } p, q \in S$$

such that

$$u = \delta(p, x_{i_n}), \; v = \delta(q, x_{j_n}), \; x_{i_n}, x_{j_n} \in X\} \tag{79}$$

where

$$\Delta E((p, u), (q,v)) = (\lambda(p, x_{i_n}) - \lambda(q, x_{j_n}))^2$$

for appropriate $x_{i_n}, x_{j_n} \in X$.

For m-ary input, this involves a minimum of $(m^2 - m)$ and a maximum of m^2 additions followed by $(m^2 - 1)$ binary comparisons, assuming $\Delta E((.,.),(.,.))$ are all computed and stored in memory for table-lookup.

The above eq. (79) using $\delta(.,.)$ is not explicit. See Eq. (57). Using $T(.,.)$ function as defined in eq. (60), we have

$$F_{n+1}(u, v) = \min\{ F_n(T(u, r), T(v, s)) + \Delta E((T(u, r), u),T(v,s), v)): \text{ for every } r, s \in P\}, \tag{80}$$

which is readily programmable.

Call the M x M square matrix, which consists of all $F_n(u, v)$, $0 \leq u, v \leq m^L - 1$, $M = m^L$, the distance matrix. This matrix can be shown to be symmetrical: if all trellises start at time t = 0, then at time t = 1, $F_1(p, q) = F_1(q, p)$. Consequently, $F_n(p, q) = F_n(q, p)$,

by the application of induction and definition of $F_n(.,.)$ for all n. This symmetry allows us to calculate only the weights on or above the diagonal of the distance matrix, a saving in 50(M - 1)/M% of the total operations in computing the whole matrix. The diagonal of the matrix consists of those sequences that diverge at t = 0 and merge at t = n; from the trellis structure, m paths diverge from and merge to a given state at any time $t \geq K$.

When two signal sequences diverge and merge over a time interval of n, the squared Euclidean distance between them is $F_n(u, u)$, $u \in S$. Ordinarily, we would expect $F_n(u, u)$ to vary with different merging states, but computations show that it stays the same for all u at a given depth in the trellis.

For those sequences which haven't yet merged at depth n, we can compare among them and select the minimum one. Defining

$$\text{Min } F_n = \min \{F_n(u, v) : \text{for all } u, v \in S\} \tag{81}$$

and

$$d_n^2 = \min \{F_n(u, u): \text{for all } u \in S\}, \tag{82}$$

we see that

$$d_{\text{free}}^2 = \min\{ d_n^2 : \text{for all } K \leq n \leq \infty\}. \tag{83}$$

As $\Delta E((.,.),(.,.))$ is a positive definite quantity, it follows that Min F_n is a non-decreasing function with depth n along the trellis. Another observation is that $d_n^2 \geq$ Min F_n as some branches carrying non-minimum Euclidean weights have to be followed in order to make two state sequences merge at a later time.

Define

$$D_N^2 \triangleq \min \{d_n^2 : \text{for all } K \leq n \leq N\} \tag{84}$$

The condition for terminating the double dynamic programming depends on when we can equate d_{free}^2 to D_N^2. This in turn depends on at what depth we are sure that d_n^2 has passed the minimum; after that, d_n^2 can only increase or at least stay constant. Thus a sufficient condition for termination is that Min $F_N \geq D_N^2$. The first level at which this occurs is defined as the decision depth. Mathematically, it is shown as

$$\{ \text{lst } N : \text{Min } F_N \geq D_N^2 \} \tag{85}$$

It is a measure the decoding depth required to get the true minimum squared Euclidean distance d_{free}^2 of a given code. It is also important in determining the real performance of a given code. For example, there may exist an unmerged pair of signal sequences of length N with Min $F_N < D_N^2$. This Min F_N will dominate the probability of symbol error P(e) and will consequently replace the d_{free}^2 as the determining influence on P(e).

4.7 Double Dynamic Programming

The algorithm using structured programming is as follows:

$$X = \{0, 1, 2, \ldots, m - 1\},$$

$$S = \{0, 1, 2, \ldots, m^L - 1\},$$

$$P = \{0, 1, 2, \ldots, m - 1\}.$$

Begin

For each $s_i \in S$, $x_i \in X$ do

$$\lambda(s_i, x_i) \leftarrow x_i f_0 + \sum_{k=1}^{L} f_i x_{i-L} ;$$

For each $u \in S$, $p \in P$ do

$$T(u, p) \leftarrow m(u - \lfloor u/m^{L-1} \rfloor \cdot m^{L-1}) + p;$$

/COMMENT/ Store the incremental weights in memory for table-lookup

For each $u \in S$, $v \in S$, $p_1 \in P$, $p_2 \in P$ do

$$\Delta E(T(u, p_1), u), (T(v, p_2), v)) \leftarrow$$

$$(\lambda(T(u, p_1), \lfloor u/m \rfloor) - \lambda(T(v, p_2), \lfloor v/m \rfloor))^2 ;$$

Choose any $a \in S$:

/COMMENT/ Setting the boundary condition : $F_0(a, a) = 0$;

For each $u, v \in S$ do

if $(u = a$ and $v = a)$ then $F_0(u, v) \leftarrow 0$ else $F_0(u, v) \leftarrow \infty$;

/COMMENT/ Computing $F_n(u, v)$ by the recursion formula;

$n \leftarrow 0$;

Repeat

$n \leftarrow n + 1$;

For each $u, v \in S$ do

For each $p_1, p_2 \in P$ do

if $F_n(T(u, p_1), T(v, p_2)) = \infty$

or $(T(u, p_1) = T(v, p_2)$ and $u = v)$

then $F_{n+1}(u, v) \leftarrow \infty$

else $F_{n+1}(u, v) \leftarrow \min\{ F_n(T(u, p_1), T(v, p_2)) +$

$\Delta E(T(u, p_1), u), (T(v, p_2), v))$;

/COMMENT/ Sequences start merging at $t \geq K$; comparisons are made from then on to get the free distance;

if $n \geq K$ then

Begin

Min $F_n \leftarrow \min \{F_n(u, v)$: for all u, v in S$\}$;

$d_n^2 \leftarrow F_n(u, u)$, for any u in S ;

$D_N^2 \leftarrow \min \{d_i^2 : K \leq i \leq n\}$;

End

until Min $F_n \geq D_n^2$;

$d_{free}^2 \leftarrow D_n^2$;

/COMMENT/ The decision depth is n;

END

4.8 Complexity of Double Dynamic Programming

The complexity of Double Dynamic Programming is estimated as follows:

1) In terms of memory, it requires m^{2L} storage for the optimal value function $F_n(.,.)$ at each stage. To list the pair sequences corresponding to the $F_n(u, v)$ requires another m^{2L} locations. Another $m^L.m^L.m^2 = m^{2K}$, $K = L + 1$, locations are needed to store the incremental Euclidean weights $\Delta E((.,.),(.,.))$.

2) In terms of computation: in each symbol period, m^{2K} additions subtractions, squaring and finally $m^{2L}(m^2 - 1)$ comparisons are required.

The amount of storage is proportional to the square of the number of states and the amount of computation is proportional to the square of the number of transitions in each symbol period. The complexity involved in double dynamic programming is the square of that of the VA. Assuming equal termination of depths, the brute-force method would increase the complexity by m^n-fold for a trellis of depth n.

CHAPTER 5

DEGRADATION CONTOURS OF PARTIAL RESPONSE SIGNALLING SYSTEMS

We showed in the last chapter that the double dynamic program is an optimal instrument in computing the free distances of real number convolutional codes. This chapter focusses on applying this programming concept in finding the free distances of PRS codes in a grid pattern fashion. We then analyse theoretically the possible error events causing degradation and obtain a wealth of information concerning the contour's shapes of both the degradation and non-degradation regions. These theoretical results are then compared with the computational ones and the resulting implications are discussed.

5.1 Normalized Free Distance

In order to compare the free distances of all codes, a base upon which all Euclidean distances are normalized is needed. For equally probable polar inputs, the mean input value is zero. For m-ary input, the input variance is (Appendix A.1)

$$\sigma_x^2 = (m^2 - 1)/12. \tag{86}$$

After passing through a PRS filter with energy (Appendix B.2).

$$R(0) = ||F(D)||^2 = \sum_{i=0}^{L} f_i^2 , \tag{87}$$

the output variance becomes (Appendix A.2)

$$\sigma_y^2 = \sigma_x^2 R(0). \tag{88}$$

We may change d_{free}^2 at will, simply by scaling the tap-gains $\{f_i\}$ of the PRS filter or by scaling the energy of the impulse response of the pulse shaping filter. In order to normalize out these effects, we divide d_{free}^2 by the output variance of the filter, Thus,

$$d_{norm}^2 = d_{free}^2 / (\sigma_x^2 R(0)) \tag{89}$$

For m-ary PAM with equal separation between levels of $\underline{a}$ volts, $d_{free}^2 = \underline{a}^2$. Since $R(0) = f_o^2 = \underline{a}^2$, d_{norm}^2 for PAM is $1/\sigma_x^2$. Define the degradation of a PRS system against ideal PAM as

$$\begin{aligned} \text{Degradation} &= 10\log_{10}(d_{norm}^2 \text{ of PRS}/d_{norm}^2 \text{ of PAM}) \\ &= 10\log_{10}(d_{free}^2/R(0)). \end{aligned} \tag{90}$$

The quantity $d_{free}^2/R(0)$ is the most fundamental quantity in measuring degradation caused by ISI. If $d_{free}^2 = R(0)$, then all the energy in the signal is utilized for detection purpose. For $d_{free}^2 < R(0)$, essentially some of the available signal energy can not be used in the detection, resulting in performance loss. A code F(D) showing $\underline{b}$ dB in degradation will require $\underline{b}$ dB higher input signal energy to attain the same

probability of symbol error or SNR ratio as that of an isolated pulse. The correlation between levels in a PRS system, while it may have good spectral aspects, does waste the energy in detection.

The d^2_{norm} for binary PAM is 4, and for 4-level PAM is 5/4. In our computation to follow, we find that for PRS systems with $K \leq 2$, d^2_{norm} is the same as the corresponding ideal PAM systems. But for $K \geq 3$, some codes will give d^2_{norm} less than that of the corresponding PAM systems, that is, $d^2_{norm} < 4$ for binary level inputs and $d^2_{norm} < 5/4$ for 4-level inputs.

5.2 Contour Maps

We are interested in the following performance measures for PRS codes:

1) Degradation,
2) 99% power bandwidth,
3) Decision Depth,
4) Total channel cost

A good way to display and present the above measures is to draw their contour maps with different tap-gains $\{f_i\}$ as the axis. We compute each measure in a rectangular grid-pattern and interpolate to form contours with constant function value. These contours are invaluable in determining trade-offs of the above quantities.

The grid pattern we use is as follows:

1) f_1 is the y-axis in steps of 0.2 over range of (-6, 6)
2) f_2 is the x-axis in steps of 0.5 over range of (-4.5, 4.5)
3) f_3 has values of 3.1, 1.6, 0.8, 0.0, -0.8, -1.6.

For K = 1, we have ordinary PAM.

For K = 2, the contours exist in one dimension along the f_1 axis. In fact, we can get these contours from contours of higher K values by cutting along the f_1 axis.

For K = 3, the contours are two-dimensional along axis f_2, f_1.

For K = 4, the contours are in three dimensions and can be shown in two dimensions by taking cuts along f_3.

We shall plot free distances in this chapter, and return to performance measures 2), 3) and 4) in chapter 6, 7, and 8 respectively.

For impulse response $F(D) = \sum_{i=0}^{L} f_i D^i$, the relative performance of $F(D)/f_0$ is the same as F(D), thus we can normalise our tap gains such that $f_0 = 1$. In this case, $F(D) = 1 + \sum_{i=1}^{L} f_i D^i$. Another point to note is that in the context of a coding system, the term constraint length K = L + 1 is used; while in the context of a band limited channel with the same characteristic as the coding system, the term channel length is usually used instead.

5.3 Analysis of Input Error Sequences

In this section we will consider the degree of input error sequence $\epsilon_x(D)$ which causes degradation as defined in eq. (90) for PRS systems with constraint length $K \leq 3$, where K = L + 1 with L = number of delay units.

We want to show that for K = 2, for which $F(D) = 1 \pm f_1 D$, the input error sequence that leads to d_{free} can only be $\epsilon_x(D) = \pm 1$. Assume that the input error sequence of degree one can lead to d_{free}; i.e.,

$\epsilon_x(D) = 1 + D$.

The Euclidean weight $d^2(\epsilon) = ||\epsilon_y(D)||^2$, where

$\mathcal{E}_y(D) = (1 + D)(1 + f_1D)$. By long multiplication, we have

$$\mathcal{E}_y(D) = 1 + f_1D + \frac{D + f_1D^2}{1 + (1 + f_1)D + f_1D^2}$$

Thus, $d^2(\mathcal{E}) = 1 + (1 + f_1)^2 + f_1^2$. But the total energy R(0) for $F(D) = 1 + f_1D$ is only $1 + f_1^2$; by comparison, we have $d^2(\mathcal{E}) > R(0)$, which is impossible. Thus, $\mathcal{E}_x(D) = 1 + D$ will not lead to d_{free} and causes degradation. The same argument is applicable for the case $\mathcal{E}_x(D) = 1 - D$, as $(1 - f_1)^2$ is a positive definite quantity. We have succeeded in showing that the input error sequence that leads to d_{free} for K = 2 can be of degree zero only. Therefore $d^2(\mathcal{E}) = R(0) = 1 + f_1^2$, and thus no degradation in SNR ratio for PRS systems with K = 2 will occur if an MLSE is used.

By using the above "proof by contradiction" principles, we now show that for K = 3 with $f(D) = 1 + f_1D + f_2D^2$, the maximum degree of $\mathcal{E}_x(D)$ which can cause degradation is one. For $\mathcal{E}_x(D) = 1$,

$$d^2(\mathcal{E}) = ||\mathcal{E}_y(D)||^2 = ||1 + f_1D + f_2D^2||^2.$$

For $\mathcal{E}_x(D) = 1 - D$ then,

$$\mathcal{E}_y(D) = 1 + f_1D + f_2D^2 - \frac{D - f_1D^2 - f_2D^3}{1 + (f_1 - 1)D + (f_2 - f_1)D^2 - f_2D^3}.$$

Accordingly, $d^2(\mathcal{E}) = 1 + (f_1 - 1)^2 + (f_2 - f_1)^2 + f_2^2$. (91).

This error sequence will cause degradation if $d^2(\mathcal{E}) < R(0)$. By inspection, we note that the only possibility where the above inequality can occur is when f_1 and f_2 are of the same sign.

Consider an error sequence of degree two that will lead hypothetically to d_{free} and cause degradation. Assume for simplicity that

$$\mathcal{E}_x(D) = 1 - D + D^2$$

and so

$$\mathcal{E}_y(D) = \frac{1 + f_1 D + f_2 D^2 - D - f_1 D^2 - f_2 D^3 + D^2 + f_1 D^3 + f_2 D^4}{1 + (f_1 - 1)D + (f_2 - f_1 + 1)D^2 + (f_1 - f_2)D^3 + f_2 D^4} .$$

The above input error sequence will lead to d_{free}, if its Euclidean weight is less than that of the input error sequence $\mathcal{E}_x(D) = 1 - D$. By comparison with eq. (91), we found that the Euclidean weight for an input error sequence of degree two is larger than that of degree one, because $(f_2 - f_1 + 1)^2$ is a positive definite quantity. Thus it is impossible for the input error sequence $\mathcal{E}_x(D) = 1 - D + D^2$ to lead to d_{free} and cause degradation. Further increase in the degree of the error sequence will not result in degradation because this will only increase the

Euclidean weight further due to the increase in the number of positive definite terms.

By the same reasoning, $\epsilon_x(D) = 1 + D$ will also lead to d_{free} while $\epsilon_x(D) = 1 + D + D^2$ will not. Again, the above statement is true only if the tap-gains f_1 and f_2 are of opposite signs. This is because when f_1 and f_2 are of the same sign, the Euclidean weight will always be greater than that of R(0) and so no degradation can occur.

The only input error sequence left to consider is of the form $1 - D^2$. It will give the signal error sequence:

$$\epsilon_y(D) = (1 - D^2)(1 + f_1 D + f_2 D^2)$$

$$= 1 + f_1 D + f_2 D^2 - D^2 - f_1 D^3 - f_2 D^4$$

$$\overline{1 + f_1 D + (f_2 - 1)D^2 - f_1 D^3 - f_2 D^4}$$

It is immediately obvious that this signal error sequence will produce Euclidean distance greater than that of R(0), and thus cannot lead to degradation at all. The same conclusion can be drawn for $\epsilon_x(D) = 1 + D^2$. We conclude that the only input error sequence that causes worse degradation is $\epsilon_x(D) = 1 \pm D$ for constraint length K = 3 and the state path merge leading to d_{free} occurs at depth K + 1.

5.4 Demarcation Contours Separating the Non-Degradation and Degradation Regions for Constraint Length 3

For PRS systems with constraint length K = 3, we have proved that degradation will occur if and only if

$$||(1 - D)(1 + f_1 D + f_2 D^2)||^2 < R(0);$$

i.e.,

$$1 + (f_1 - 1)^2 + (f_2 - f_1)^2 + f_2^2 < 1 + f_1^2 + f_2^2;$$

i.e.,

$$f_1^2 - 2f_1 f_2 + f_2^2 - 2f_1 + 1 < 0 \qquad (92)$$

The above inequality signifies a region within a parabola tilted by 45^0 counterclockwaise, and marks the region where degradation is inevitable even with an MSLE receiver. For $_x(D) = 1 + D$, the resulting inequality becomes

$$f_1^2 + 2f_1 f_2 + f_2^2 + 2f_1 + 1 < 0. \qquad (93)$$

Eq. (9-) and eq. (93) are mirror reflection of each other along the f_2 axis. The 0 dB curve in the 1st quadrant of figure 5.1 corresponds to eq. (92), while the 0 dB curve in the 4th quadrant corresponds to eq. (93).

5.5 Contours with Given Degradation for Constraint Length 3

Recall that $d^2_{norm} = d^2_{free}/(\sigma_x^2 R(0))$. The region within the parabola where degradation occurs consists of contours of constant degradation as defined in eq. (90). The d^2_{norm} is given by

$$d^2_{norm} = (1 + (f_1 - 1)^2 + (f_2 - f_1)^2 + f_2^2)/(R(0)/4),$$

as $\sigma_x^2 = 1/4$.

By letting $d^2_{norm} = x$, the equation for constant d^2_{norm} becomes

$$(8 - x)f_1^2 - 8f_1f_2 + (8 - x)f_2^2 - 8f_1 + (8 - x) = 0. \qquad (94)$$

Let us analyse the above equation as a general equation of second degree of form:

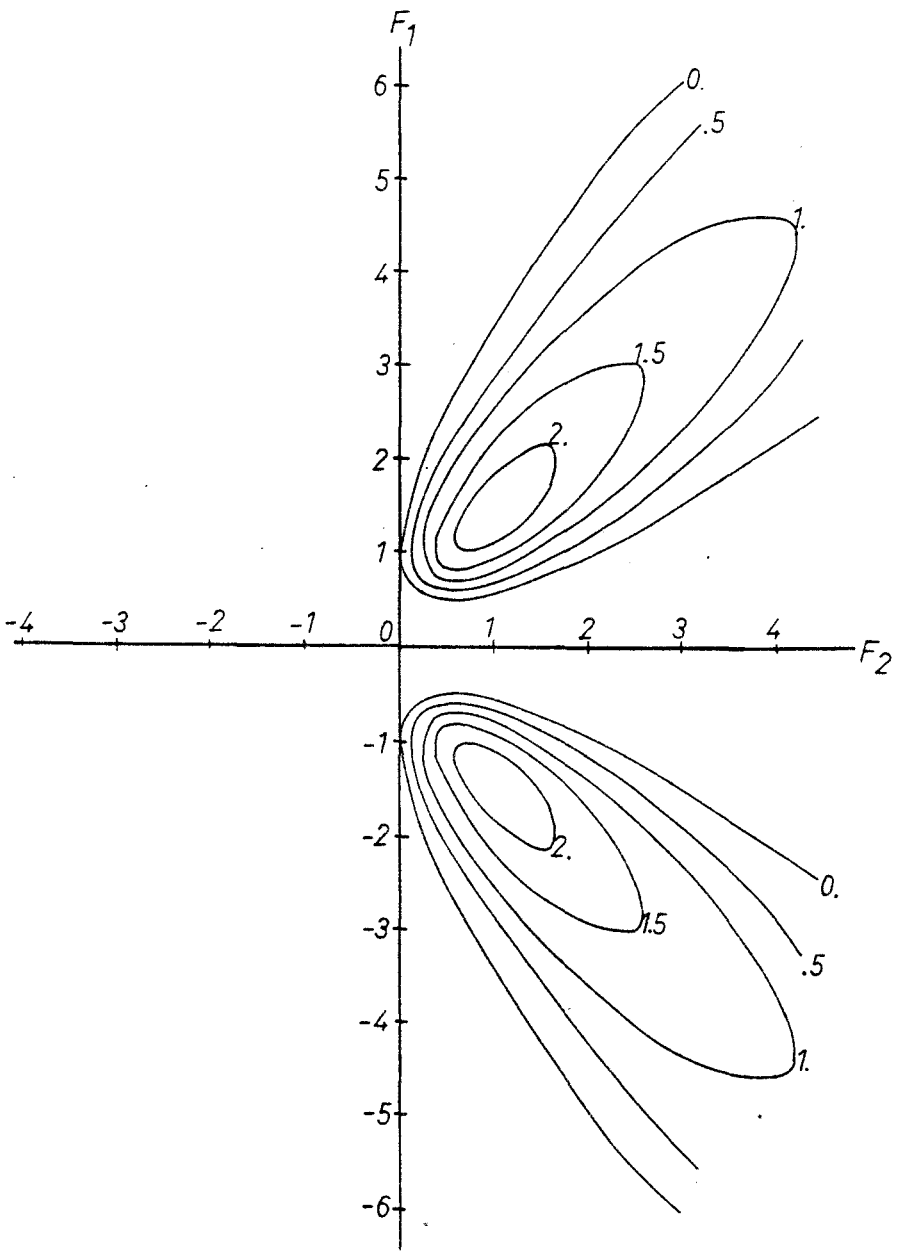


Fig. 5.1 Contours of constant degradation

$F_0 = 1$, $F_3 = 0.0$

$$Af_1^2 + Bf_1f_2 + Cf_2^2 + Df_1 + Ef_2 + F = 0.$$

By comparison of eq. (94) with eq. (95) we have

$$A = 8 - x ,$$

$$B = -8,$$

$$C = 8 - x,$$

$$D = -8,$$

$$E = 0, \qquad \text{and}$$

$$F = 8 - x.$$

The discriminant $B^2 - 4AC$ will give

1) a parabola if the discriminant is equal to zero, or

2) an ellipse if the discriminant is less than zero.

From eq. (94) the discriminant is equal to

$$(-8)^2 - 4(8 - x)^2.$$

Thus the curve is a parabola if $-(x^2 - 16x + 48) = 0$. This condition is equivalent to $(x^2 - 16x + 48) = 0$, which after factorization is equivalent to $(x - 12)(x - 4) = 0$, which is equivalent to $x = 4$, as $x = 12$ is outside the range of d^2_{norm}. This agrees with the fact that for $d^2_{norm} = 4$, there is no degradation and the curve is a parabola. Substituting $x = 4$ into eq. (94) will give the equation of the parabola on which there is no degradation. Refer to all the 0 dB curves of the degradation contours of this chapter.

For $B^2 - 4AC < 0$, this is equivalent to $(x^2 - 16x + 48) > 0$, which after factorization is equivalent to $(x - 12)(x - 4) > 0$, which then is equivalent to $x < 4$. Hence, a contour with constant degradation as defined in eq. (90)

corresponds to a certain contour with a constant d^2_{norm} less than 4. This contour is an ellipse with the general form of eq. (94) within the parabola with eq. (92) or eq. (93). Refer to Figure 5.1 - 5.7.

5.6 Further Analysis of Input Error Sequences for PRS Systems with Longer Constraint Lengths

We will now move on to consider the case for constraint length K = 4. From our computation, we find that input error sequences of order zero, one and two all can lead to d_{free}. R. R. Anderson and G. J. Foschini [22] have shown that input error sequence of order three can also occur. From the definition of an error event, we note that $\in_x(D)$ of degree n leading to d_{free} will occur between paths that merge at depth K + n in the state trellis; i.e., $d_{free} = d_{K+n}$ in the notation of section 4.6, eq. (82).

1) It is obvious that d_{free} will be equal to d_K if and only if d_K is less than both d_{K+1} and d_{K+2}. Accordingly, $d^2_{free} = R(0)$ with $\in_x(D) = \pm 1$. No degradation will occur if an MLSE receiver is used.

2) In the same way, d_{free} will be equal to d_{K+1} if and only if d_{K+1} is less than both d_K and d_{K+2}. With $\in_x(D) = 1 - D$, d^2_{K+1} can be computed as

$$d^2_{K+1} = ||(1 - D)(1 + f_k D + f_2 D^2 + f_3 D^3)||^2$$

$$= 1 + (1 - f_1)^2 + (f_1 - f_2)^2 + (f_2 - f_3)^2 + f_3^2 . \qquad (95)$$

This input error sequence of $1 - D$ can lead d^2_{K+1} to be equal to d^2_{free}

and causes degradation if the tap gains are of the same sign. When the tap gains are of opposite signs, the corresponding input error sequence that will cause degradation is $\epsilon_x(D) = 1 + D$.

3) d_{free} will be equal to d_{K+2} if and only if d_{K+2} is less than both d_K and d_{K+1}. For the input error sequence $\epsilon_x(D) = 1 - D + D^2$, we have

$$d_{K+2}^2 = ||(1 - D + D^2)(1 + f_1D + f_2D^2 + f_3D^3)||^2$$

$$= 1 + (f_1 - 1)^2 + (1 + f_2 - f_1)^2 + (f_1 + f_3 - f_2)^2$$

$$(f_2 - f_3)^2 + f_3^2. \qquad (96)$$

This input error sequence can lead to degradation if the tap gains are of the same sign. For tap gains of opposite signs, $\epsilon_x(D) = 1 + D - D^2$ will be the necessary input error sequence.

5.7 Equations of Contours with Given Degradation for Constraint 4

In this section we will derive the equations of contours with given degradations for PRS systems with constraint length K = 4.

1) When $d_{free} = d_{K+1}$, and letting $d_{norm}^2 = x$, we have from eq. (89)

$$x = d_{K+1}^2/(R(0)/4).$$

This is equivalent to the equation

$$4d_{K+1}^2 = R(0)x,$$

and substituting eq. (95) for d^2_{K+1} gives the following

$$(8 - x)f_1^2 - 8f_1(1 - f_2) + (8 - x)(1 + f_2^2 + f_3^2) - 8f_2f_3 = 0. \tag{97}$$

This is the equation of the locus of points which satisfies $d^2_{norm} = x$, with d^2_{free} occurring from a merge of depth K + 1. By substituting x = 4 in to eq. (97), we get the equation for

$$d_K^2 = d_{K+1}^2 .$$

This equation is

$$f_1^2 - 2f_1(1 + f_2) + (f_2 - f_3)^2 + 1 = 0. \tag{98}.$$

Essentially, this is the parabolic equation demarcating the region of no degradation from the region due to the input error sequence $\epsilon_x(D) = 1 - D$. From Fig. 5.5. to Fig. 5.7, we see that when the tap-gains f_3's are positive, all the 0 dB curves corresponding to eq. (98) are situated in the 1st quadrant. From Fig. 5.2 to Fig. 5.4, in which all f_3's are negative, we see that the 0 dB curves are mirror reflections along the f_2 axis of the 0 dB curves with positive f_3's and thus are situated in the 4th quadrant. The equations of these mirror reflections are

$$f_1^2 + 2f_1(1 + f_2) + (f_2 + f_3)^2 + 1 = 0,$$

with the appropriate negative f_3's. With regard to the contours

with degradation, the above discussions still hold. The equations of the degradation contours with f_3's negative are given by:

$$(8 - x)f_1^2 + 8f_1(1 + f_2) + (8 - x)(1 + f_2^2 + f_3^2) + 8f_2f_3 = 0.$$

Refer to Fig. 5.2 - Fig. 5.4.

2) Similarly, by substituting eq. (96) into the equation

$$4d_{K+2}^2 = R(0)x,$$

we get the equation of the locus of points satisfying $d_{free}^2 = d_{K+2}^2$, with x being the d_{norm}^2. The equation becomes

$$(12 - x)f_1^2 - 2f_1(8 + 8f_2 - 4f_3) + (12 - x)(1 + f_2^2 + f_3^2) + 8f_2(1 - 2f_3) = 0 . \quad (99)$$

Substituting x = 4 into eq. (99) gives the equation of $d_K^2 = d_{K+2}^2$:

$$2f_1^2 - 2f_1(2 + 2f_2 - f_3) + (1 + f_2)^2 + f_3^2 + (f_2 - f_3)^2 - 2f_2f_3 + 1 = 0. \quad (100)$$

3) Finally, we get the equation of $d_{K+1}^2 = d_{K+2}^2$ by equating eq. (95) and eq. (96):

$$f_1^2 - 2f_1(1 + f_2 - f_3) + (1 + f_1)^2 - 2f_2f_3 + f_3^2 = 0. \quad (101).$$

This equation provides the sufficient condition so that all the degradation curves due to state merges at different depths can be

shown to situate in different regions of the $(f_2 - f_1)$ plane. This equation turns out to be the locus of points of two parallel straight lines across the $(f_2 - f_1)$ plane. Between the two parallel lines, d^2_{K+1} d^2_{K+2}. These parallel lines are shown as dotted lines in the figures. Within the parabolic curve of eq. (100), degradation occurs if d^2_{free} is less than d^2_K. But before we can be sure that $d^2_{free} = d^2_{K+2}$, we must show that d^2_{K+2} is the least among d^2_{K+n} for $0 \leq n \leq 2$ and $K = 4$. If the degradation region within the locus of points of eq. (100), where $d^2_{K+2} = d^2_K$, intersects with the region between the two parallel lines of eq. (101), then in the region of intersection, $d^2_{free} = d^2_{K+2}$. The loci of points of eq. (99) and eq. (100), as shown in Fig. 5.5 - Fig. 5.7 for positive f_3's, are across the 2nd, 3rd and 4th quadrants. The locus of points of eq. (100) for 0 dB curves is almost within the regions of eq. (101) except for a small section which shows a jump, or point of inflection. Generally, a jump resulting in contour discontinuity occurs when a contour with a given d^2_{norm} passes from a region where $d^2_{free} = d^2_{K+2}$ into another region where $d^2_{free} = d^2_{K+1}$, thus changing the equation of the contour from eq. (99) to eq. (97). For constraint length $K = 4$, only the 0 dB curve exhibits this jump when its equation changes from eq. (100) to eq. (98). For the case where all f_3's are negative, their corresponding loci of points on which $d^2_{free} = d^2_{K+2}$ and $d^2_K = d^2_{K+2}$ are the mirror reflections along the f_2 axis of eq. (99) and eq. (100) respectively. They are shown in Fig. 5.2 - Fig. 5.4.

Summarising eqs. (98), (100) and (101) subdivide the $(f_2 - f_1)$ plane into regions where different error events resulting in degradations can occur. Eq. (97) gives the locus of points on which a given degradation occurs with $d^2_{free} = d^2_{k+1}$. These regions are referred to as region 1. Eq. (99) gives the locus of points on which a given degradation occurs with $d^2_{free} = d^2_{k+2}$. These regions are referred to as region 2. Eq. (101) provides the sufficient condition to ensure the validity of the above observations on different degradation regions. Within the locus of points satisfying eq. (101), $d^2_{k+2} < d^2_{k+1}$. The intersection of this region with region 2 is the region within which degradations will occur from state merges of depth K + 2. Outside this region on the $(f_2 - f_1)$ plane, $d^2_{K+2} > d^2_{K+1}$. Thus the intersection of the region where $d^2_{K+2} > d^2_{K+1}$ with region 1 gives the region within which degradations occur from state merges of depth K + 1. For the remaining areas on the $(f_2 - f_1)$ plane, d^2_{free} occurs at a state merge of depth K and thus no degradation results. Finally, eqs. (98), (100) and (101) intersect at a single point, known as the point of inflection, as shown in Fig. 5.2 - Fig. 5.7.

5.8 Duality of PRS Systems

In the last section, we noticed that the degradation curves for negative f_3's are mirror reflections along the f_2 axis of the corresponding ones with positive f_3's. This is a manifestation of the duality of the codes. The duality of PRS systems 1 - D and 1 + D was noted by Kobayashi [10]. From our simulations and analysis above, we discovered that this duality of codes can be further extended, at least for the cases where the constraint length K is equal to 3 and 4. This duality

is due to the fact that an error sequence or its complement will result in the same d^2_{free}, if the sign of the tap gains are adjusted accordingly.

For K = 3 $\quad 1 - f_1D + f_2D^2$ is dual with $1 + f_1D + f_2D^2$.

For K = 4 $\quad 1 + f_1D \pm f_2D^2 + f_3D^3$ is dual with

$1 - f_1D \pm f_2D^2 - f_3D^3$ and that

$1 - f_1D \pm f_2D^2 + f_3D^3$ is dual with $1 + f_1D \pm f_2D^2 - f_3D^3$.

This means that the contours of degradation for positive f_3's may be reflected about the f_2 axis to obtain those degradation curves with negative f_3's. This manifestation of the duality of codes is valid for only degradation and decision depth contours because only the above two parameters are error events dependent. The choice of PRS systems with positive or negative f_3's may then depend solely on the total channel costs which we will discuss in Chapter 8.

5.9 General Comments on the Degradation Contours

In summary, within the locus of a parabolic contour where no degradation occurs are degradation contours of elliptical shapes. As a degradation contour crosses a region with a different merging depth for d^2_{free}, a jump or point of inflection results when the equation of the contour changes.

Magee and Proakis [15] gave an estimate on the minimum d^2_{norm} due to different error events for channels with constraints length up to 10. This minimum d^2_{norm} corresponds to the deepest depression in our degradation contours. They exploited the quadratic form of d^2_{norm}, and used the fact that the minimum of d^2_{norm} under unity energy constraint for

an isolated impulse response is the minimum eigenvalue of the matrix. of the quadratic form. They considered some input error sequence $\epsilon_x(D)$ that may give rise to the minimum Euclidean weight. They postulated that the input error sequences of form $1 \pm D$ causes the minimum d^2_{norm} occurring from merges of different depths due to different input error sequences, but were unable to prove this extremal property of $\epsilon_x(D) = 1 \pm D$. It was later shown by Anderson and Foschini [23] that the above statement is indeed true but only up to constraint length $K = 6$. Thus the minimum d^2_{free} occurs from a merge of depth $K + 1$ for $K \leq 6$. This can be seen in the Fig. 5.2 to Fig. 5.4 where f_3's are negative. Notice that the worst degradation in terms of dB occurs in the 4th quadrant, where d_{free} occurs from a merge of depth $K + 1$. For example, a 4 dB degradation occurs in this region for $f_3 = -0.8$. While within the region where d^2_{free} occurs from a merge of depth $K + 2$, only around 0.5 dB degradation is observed. At $f_3 = 0$, we have $K = 3$. Here the region in which d^2_{free} equals d^2_{K+2} ceases to exist and d^2_{free} only occurs from merges at depth K or $K + 1$. This is shown in Fig. 5.1. As f_3's become positive, degradation contours will be the reflections along the f_2 axis of their negative counterparts.

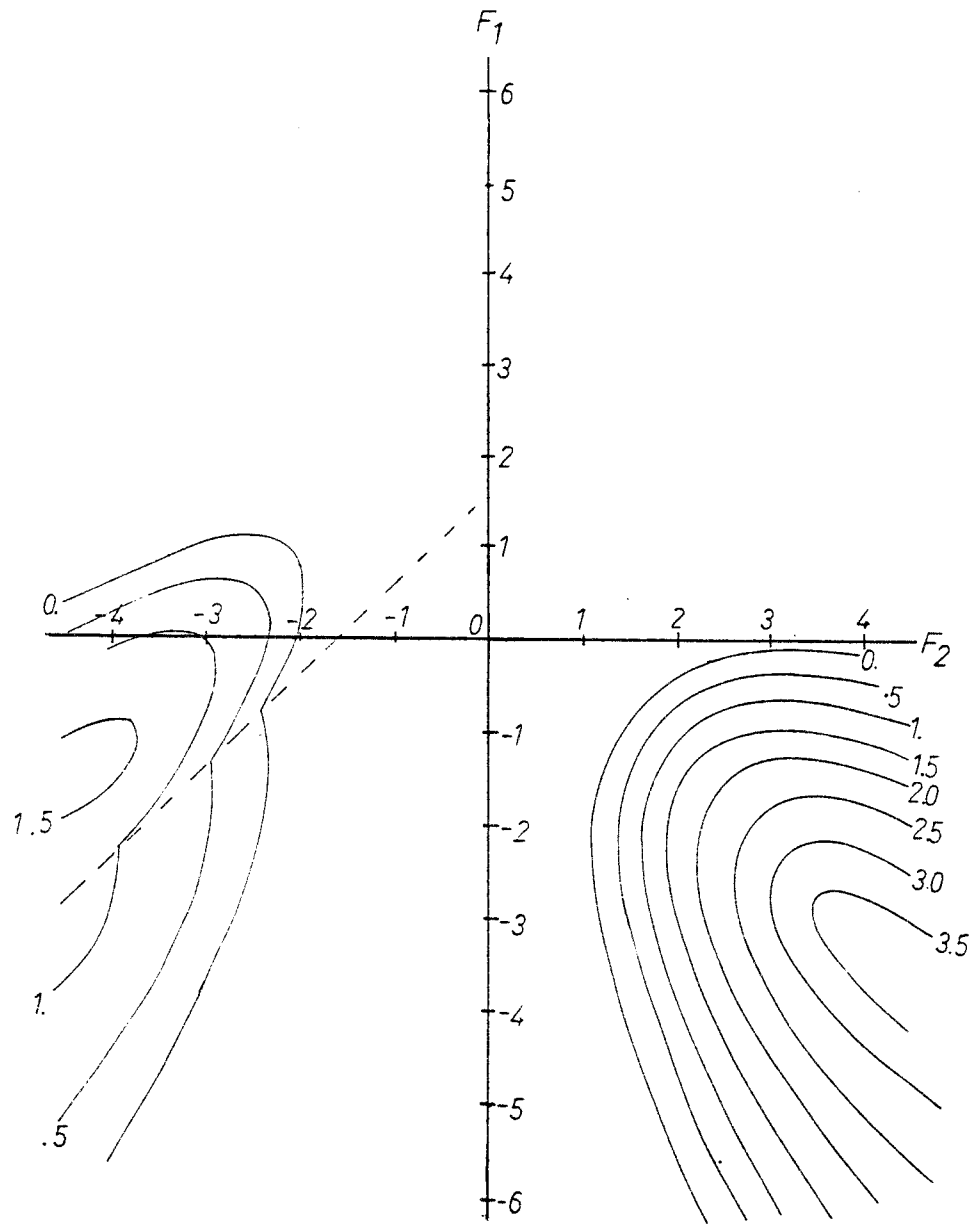


Fig. 5.2 Contours of constant degradation

$F_0 = 1$, $F_3 = -3.1$.

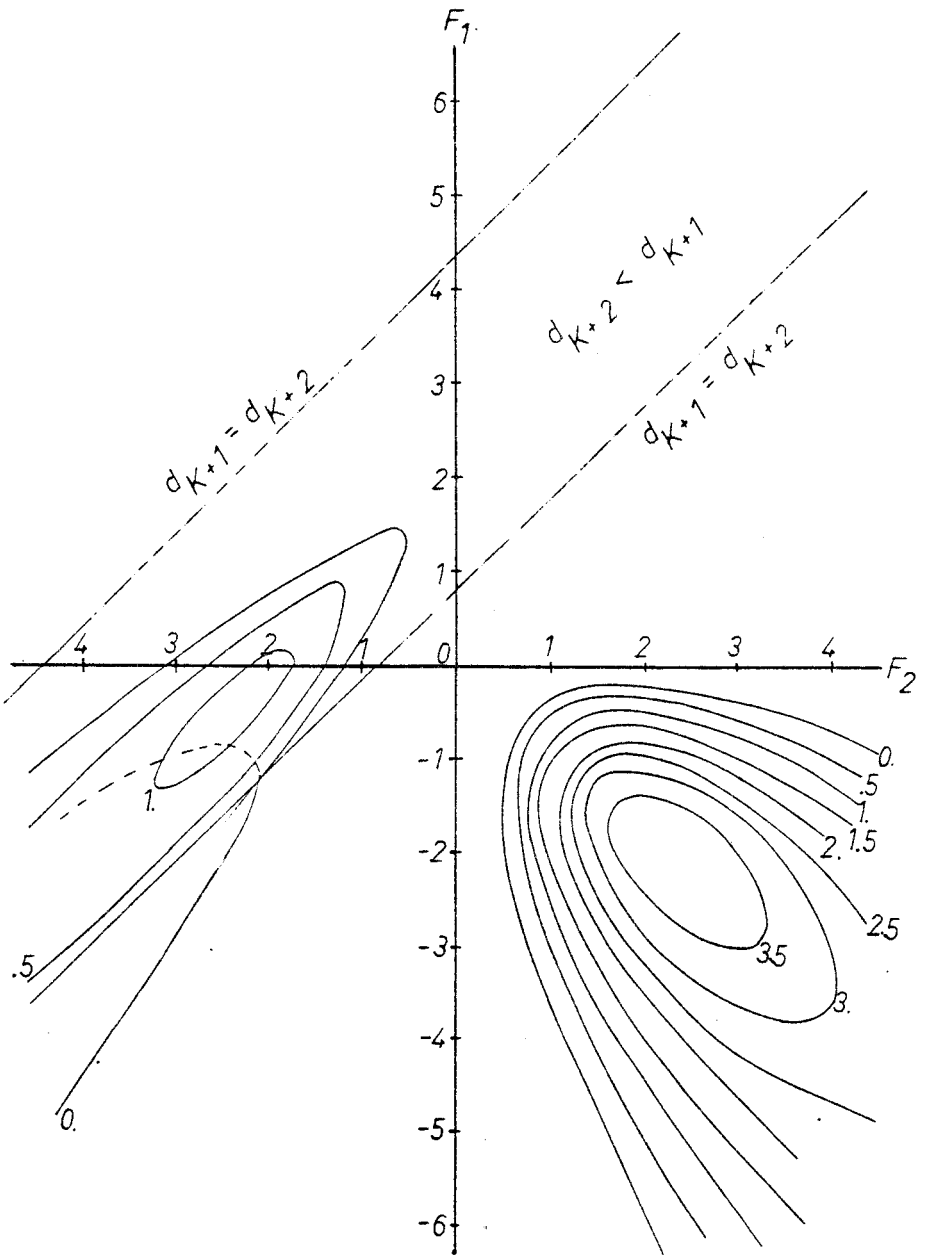


Fig. 5.3 Contours of constant degradation
$F_0 = 1$, $F_3 = -1.6$.

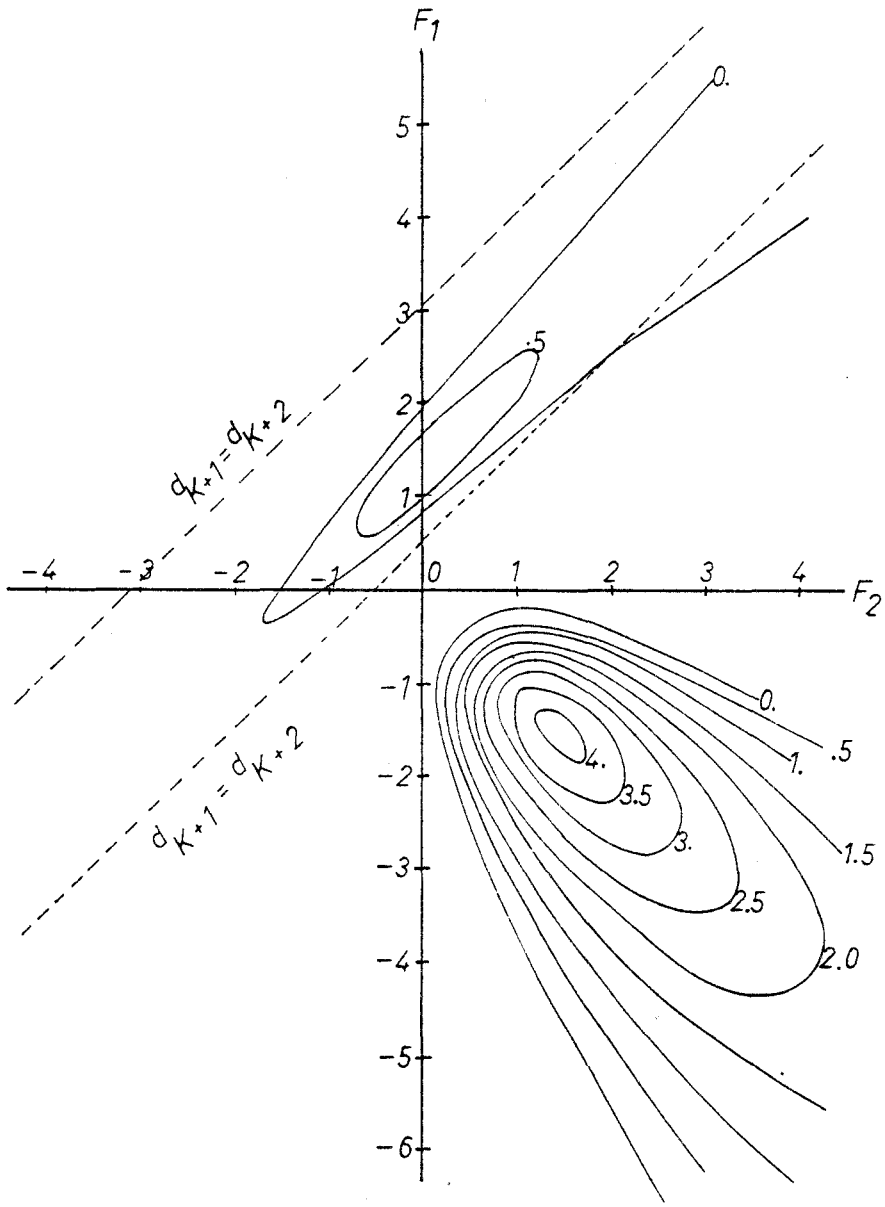


Fig. 5.4 Contours of constant degradation

$F_0 = 1$, $F_3 = -0.8$.

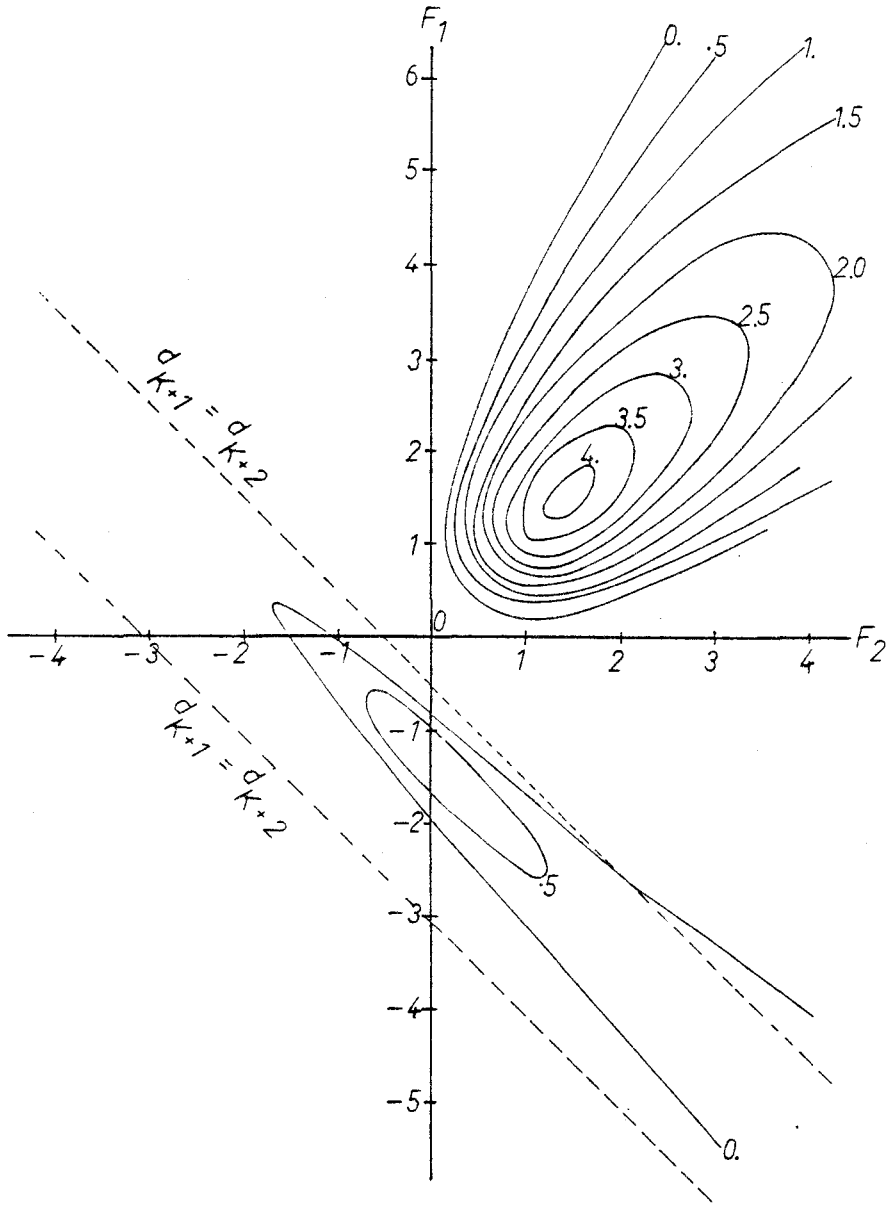


Fig. 5.5 Contours of constant degradation

$F_0 = 1$, $F_3 = 0.8$.

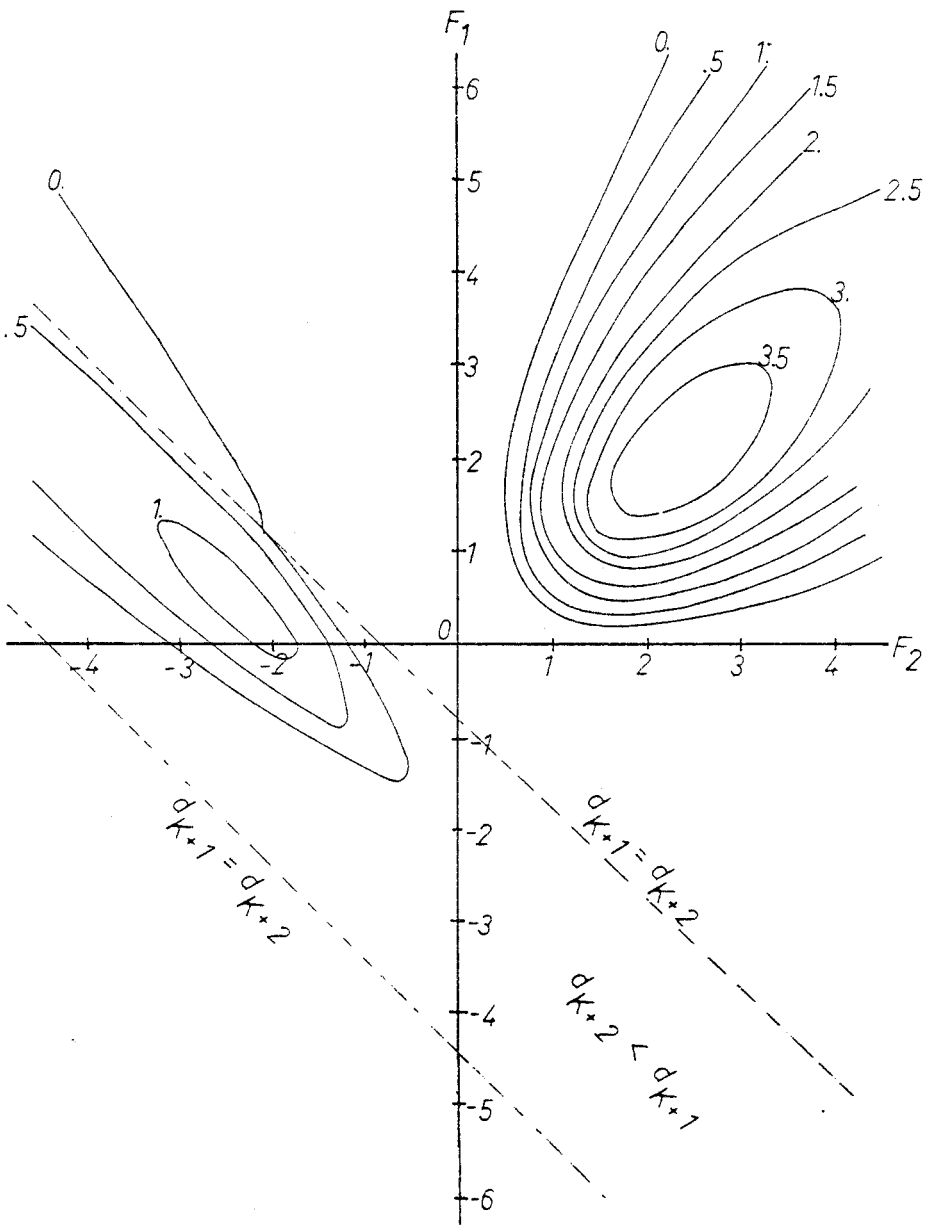


Fig. 5.6 Contours of constant degradation

$F_0 = 1$, $F_3 = 1.6$.

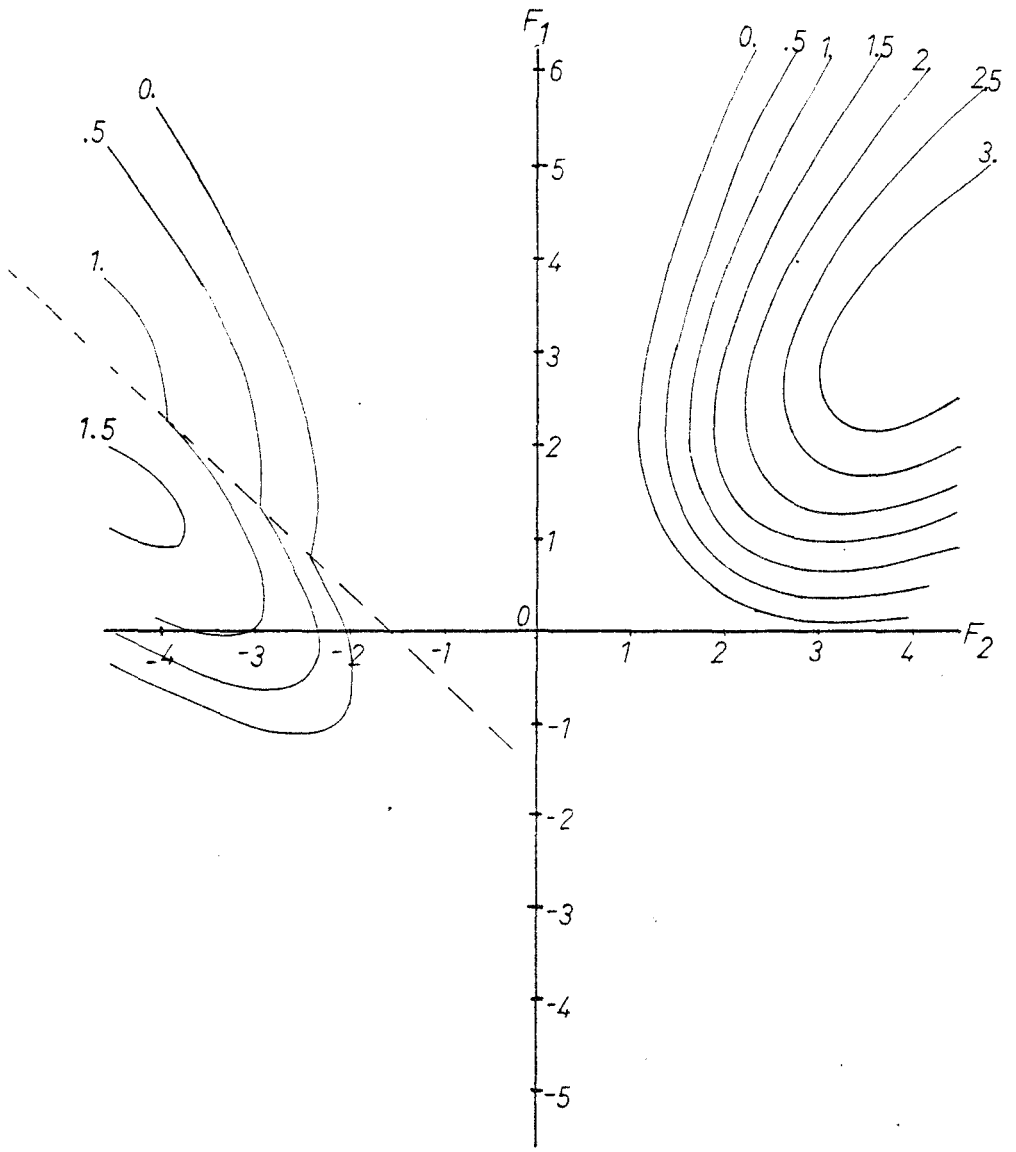


Fig. 5.7 Contours of constant degradation

$F_0 = 1$, $F_3 = 3.1$.

CHAPTER 6

CATASTROPHIC ERROR PROPAGATION

The previous chapter presents the degradation contours of PRS systems with constraint lengths up to four. Another concept which is intimately related to free distances is the decision depth. In fact the definition of decision depth utilizes the concept of free distance. In this chapter, we start by developing the concept of decision depth, and then by using the results of the double dynamic program on PRS systems, we analyse how catastrophic error propagation comes about.

6.1 Decision Depth and Catastrophic Error Propagation

In trellis decoding, decision depth is the least depth in a trellis at which all pairs of paths, either merged or not, have Euclidean distance between them greater than the free distance. It has two attributes: first, it gives the maximum length of a path that must be stored in memory when the VA is used. Receiver complexity increases linearly with this length . Second, it provides a measure of the depth in the trellis where the dynamic program has found the free distance. If a decoder decides on a symbol at a depth shorter than that of its decision depth, its performance in terms of the probability of symbol errors will be poorer than that predicted by the free distance, that is, the effective free distance will be decreased.

Recall that the definition of decision depth in eq. (85) is {lst N: Min $F_N > D_N^2$}. For certain codes, the decision depth N approaches infinity, and we say catastrophic error propagation (CE) occurs. For these codes, the performances can never achieve that predicted by free distance no matter how far down the trellis the algorithm pursues. A closer look reveals that CE results if the Euclidean distance between any two non-merging signal sequences does not increase with time. Min F_n is non-decreasing with n because $\Delta E((.,.)(.,.))$ is a positive definite quantity. If $\Delta E((.,.)(.,))$ is small or tends to be zero, then Min F_n increases only slightly or remains the same with increasing depth n. Thus the decision depth gets larger or approaches infinity.

Let us consider a state sequence pattern that will produce zero incremental Euclidean weight. Considering at depth n and referring to section 3.6 and 4.6 for definitions, we let

for all $N > n$, $D_N^2 = d_n^2$ and Min $F_n < d_n^2$.

Assume

$$\delta(p, x_{1_n}) = q, \quad \delta(q, x_{2_n}) = p, \qquad p, q \in S, \tag{104}$$

where the subscript 1_n refers to sequence 1 at time n. Now suppose

$$x_{2_n} = x_{1_{(n+1)}} \quad \text{and} \quad x_{1_n} = x_{2_{(n+1)}}, \tag{105}.$$

then the output levels become

$$\lambda(q, x_{1_{(n+1)}}) = \lambda(q, x_{2_n}),\ \lambda(p, x_{2_{(n+1)}}) = \lambda(p, x_{1_n}) .$$

Now if

$$\lambda(q, x_{2_n}) = \lambda(p, x_{1_n}) \tag{106}$$

then

$$\lambda(q, x_{1_{(n+1)}}) = \lambda(p, x_{2_{(n+1)}}).$$

If the conditions in eq. (104), (105) and (106) stay true for all $n \geq K - 1$ then $F_n(p, q) = F_{n+1}(p, q)$. As the Euclidean distance between these two non-merging sequences does not increase with depth, this $F_n(p, q)$ will be selected as the Min $F_{n'}$ for any $n' > n$. As Min $F_{n'}$ does not increase with depth, it is always less than D_N^2 for all N. Fig. 6.1 shows a typical section of a state trellis in which catastrophic error propagation can occur. The state sequence pair causing catastrophic error propagation is:

....p q p q....

....q p q p....

with a period of T. This is the simplest state sequence pair for CE to occur, other patterns have longer period. Conditions for the occurrence of catastrophic error propagation are:

1) The output caused by transitions of state sequence #1

is equal to that of state sequence #2; i.e., $y(\alpha_{2_k}) = y(\alpha_{2_k})$ $k \geq K - 1$.

2) Both the state sequence pair and the state sequence have to be periodic, although with different periods. Periodic means that a state sequence starting with a given state will return to the start state in a time interval of nT, $n \geq 2$. For example, the sequence ...p q r s p... has a period of 4T. Usually, a state sequence pair is formed when one state sequence is displaced by a multiple of T from an identical one. See. Fig. 6.2

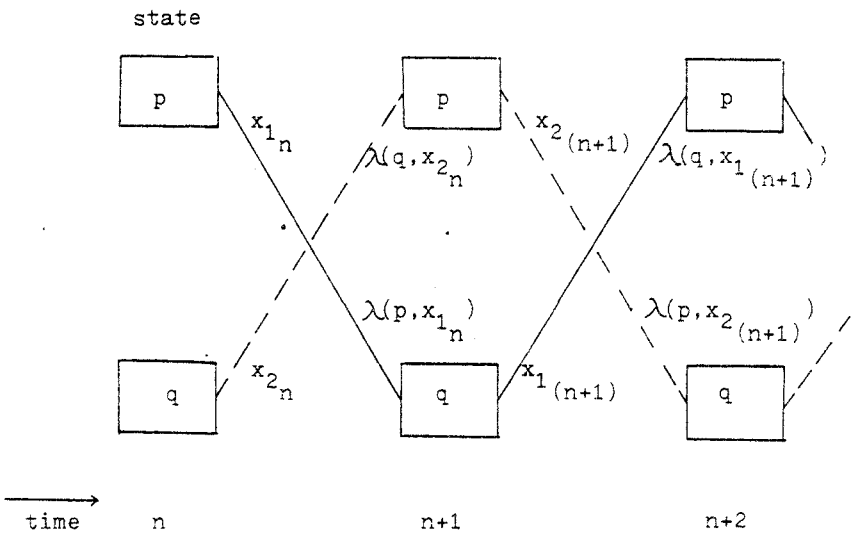
6.2 State Sequence Pairs Causing Catastrophic Error Propagation for Constraint Length K = 3.

As an example, consider K = 3 with $F(D) = 1 + f_1 D + f_2 D^2$. A typical section of the state trellis for K = 3 is shown in Fig. 3.5 State 1 is represented in the shift-register as (01). An input x = 1, that is, a one transition in state 1 will cause the machine to change to state 2 represented as (10). While in state 2, a zero transition will give state 1. Thus eq. (104) is satisfied. In order to satisfy eq. (106), we need

$$\lambda(1,1) = \lambda(2,0), \text{ which is equivalent to}$$
$$y(1,2) = y(2,1), \text{ which is equivalent to}$$
$$f_1 = f_2 + 1 \tag{107}$$

See Fig. 6.11.

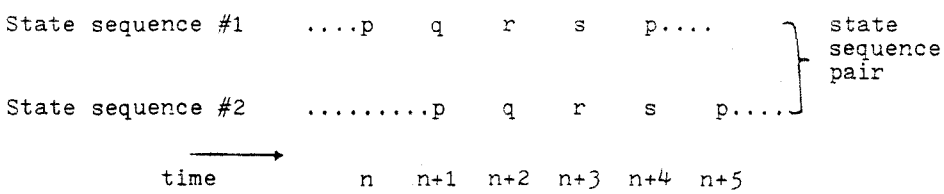
The above linear equation is a locus of points in the $(f_2 - f_1)$ plane, along which catastrophic error propagation can occur. But computations



x_{1_n} – input data bit for sequence #1 at time n

x_{2_n} – input data bit for sequence #2 at time n

Fig. 6.1 Two state paths of a state trellis that may cause catastrophic error propagation if $\lambda(q,x_{2_n}) = \lambda(p,x_{1_n})$ for all $n \geq K-1$..



Period of each sequence = 4T

Period of the sequence pair = 1T = Displacement of #1 from #2

Fig. 6.2 Periodicity of a state sequence and its associated state sequence pair

show that this is true only in the 1st and 3rd quadrants. For 2nd and 4th quadrants, the locus of points with catastrophic error propagation is governed by the equation

$$f_1 = f_2 - 1 \tag{108}$$

instead. This straight line has a negative slope and makes an angle of 135^o with the f_2 axis.

There are other codes with catastrophic error propagation due to different state sequence pairs. Consider the code $1 + D + D^2$ for which CE is caused by the state sequence pair

$$\begin{array}{l} \ldots.3\ \ 1\ \ 2\ \ 3\ldots. \\ \ldots.2\ \ 3\ \ 1\ \ 2\ldots. \end{array} \tag{109}$$

Note that the sequences are just one symbol period T displaced from one another. Each sequence has a period of T. But the sequence pair has a period of T only. The above sequence pair causes CE because $f_2 = f_1 = 1$, thus

$$1 + f_1 = f_1 + f_2 \quad \text{and} \quad 1 + f_2 = f_1 + f_2. \tag{110}$$

Hence, $y(3,1) = y(2,3)$, $y(1,2) = y(3,1)$ and $y(2,3) = y(1,2)$. Refer to Fig. 3.5. Referring to the state sequence pair (109), we note that the Euclidean distance between the sequences over the 3T period is zero. As the sequence pair repeats itself indefinitely with time, the Euclidean distance between the sequences will not increase with depth resulting in CE. This codes is shown as a cross in the 1st quadrant at

coordinates (1,1) in the $(f_2 - f_1)$ plane of Fig. 6.3.

The dual of the above code takes on a state sequence pair of form

$$\begin{array}{l} \ldots 0\ \ 0\ \ 2\ \ 3\ \ 1\ \ 0\ \ 0\ldots. \\ \ldots 3\ \ 1\ \ 0\ \ 0\ \ 0\ \ 2\ \ 3\ldots. \end{array} \qquad (111)$$

with a period of 3T. This dual code is $1 - D + D^2$. By means of the same argument as above, we infer that this sequence pair causes CE because $f_1 + f_2 = 0 = 1 + f_1$ and that $f_2 = 1$. Hence, $y(0,0) = y(3,1)$, $y(0,2) = y(1,0)$ and $y(2,3) = y(0,0)$. Refer to Fig. 3.5 and the sequence pair (111). Note that although that the period of the sequence pair is 3T, the period of each sequence is 6T. This code is shown as a cross in the 4th quadrant at coordinates (1,-1) in Fig. 6.3. Also note that the sequence pair is formed from two identical sequences displaced from each other by 3T.

For the code $1 + D^2$, the state sequence pair that causes catastrophic error propagation is

$$\begin{array}{l} \ldots. 0\ \ 0\ \ 2\ \ 1\ \ 0\ldots. \\ \ldots. 2\ \ 1\ \ 0\ \ 0\ \ 2\ldots.. \end{array} \qquad (112)$$

with a period of 2T. Each sequence has a period of 4T. Using the same inference technique as above, we note that $y(0,0) = y(2,1)$ and $y(0,2) = y(1,0)$ because $f_1 = 0$ and $f_2 = 1$ for this code, and so the

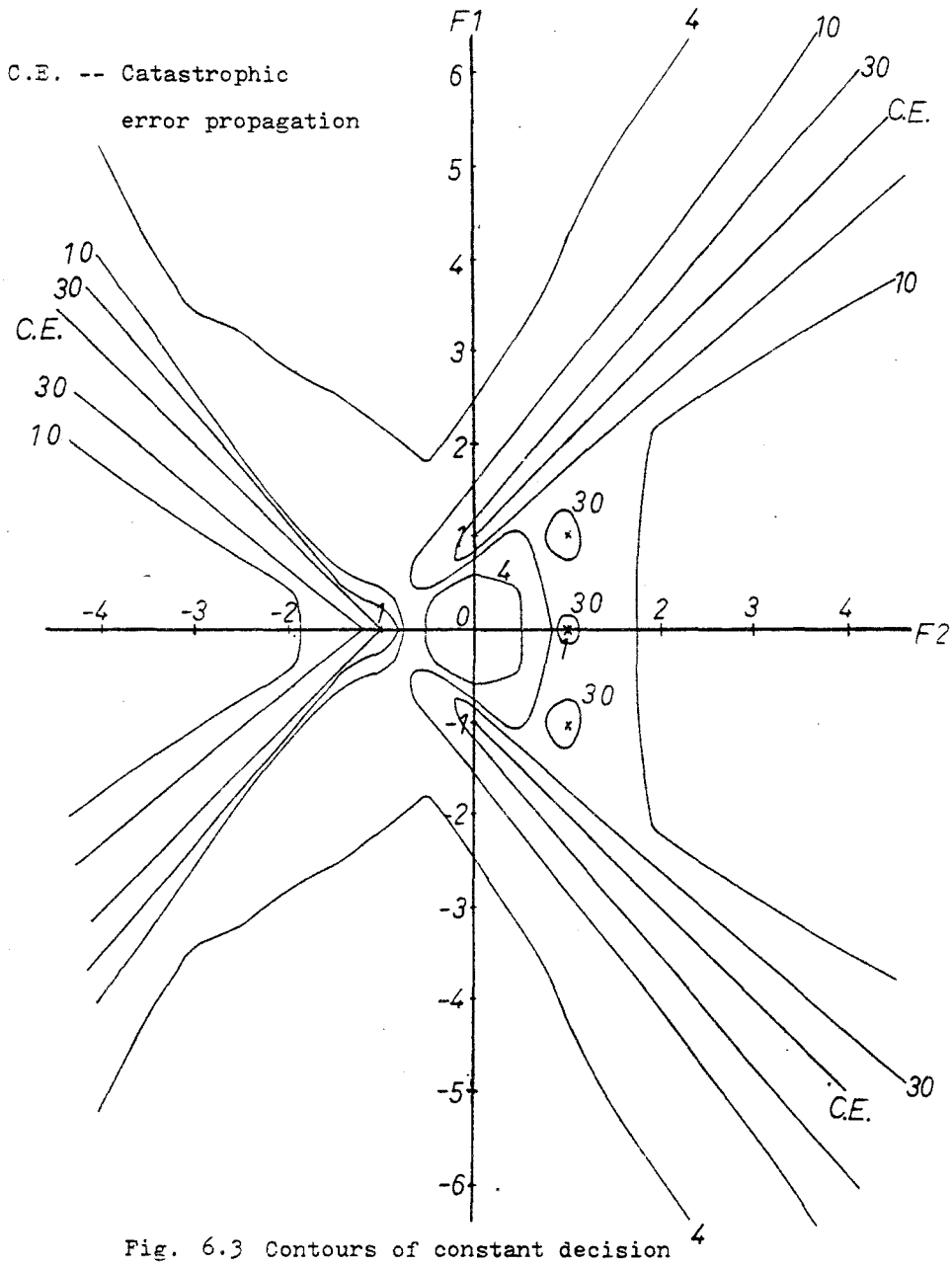


Fig. 6.3 Contours of constant decision depth ; $F_0 = 1$, $F_3 = 0.0$.

sequence pair (112) causes CE. The dual of the above code is $1 - D^2$ and the pair of sequences that causes CE is

$$\begin{array}{l} \ldots.1\ \ 2\ \ 1\ \ 2\ldots\ldots \\ \ldots.3\ \ 3\ \ 3\ \ 3\ldots\ldots \end{array} \qquad (113)$$

This is the first sequence pair which consists of a pair of sequences that are not just mere displacements of each other. Referring to Fig. 6.2 again, we see $y(1, 2) = 1 + f_2$ and $y(3, 3) = 1 + f_1 + f_2$. As $f_1 = 0$, therefore

$$1 + f_2 = 0 = 1 + f_1 + f_2$$

Thus $y(1, 2) = y(3, 3) = y(2, 1)$. Also as $y(0, 0) = y(3, 3) = 0$, therefore the sequence pair that causes catastrophic error propagation can also be

$$\begin{array}{l} \ldots..1\ \ 2\ \ 1\ \ 2\ldots\ldots \\ \ldots..0\ \ 0\ \ 0\ \ 0\ldots\ldots \end{array}$$

The above two codes, which have catastrophic error propagation, are illustrated in Fig. 6.3 as crosses at coordinates

(1, 0) and (-1, 0) respectively.

6.3 State Sequence Pairs Causing Catastrophic Error Propagation for Constraint Length K = 4.

A typical section of the state trellis for K = 4 and binary inputs is shown in Fig. 6.4 Also shown are the outputs for different transitions. For this trellis, the simplest state sequence pair causing catastrophic error propagation is

$$\begin{aligned} &.....2 \quad 5 \quad 2 \quad 5.... \\ &.....5 \quad 2 \quad 5 \quad 2.... \end{aligned} \tag{114}$$

conditioned on $y(2,5) = y(5,2)$. Eq. (104) is satisfied, as

$$\delta(2,1) = \delta(5,0).$$

$y(2,5) = y(5,2)$ is equivalent to

$$1 + f_2 = f_1 + f_3,$$

which is equivalent to

$$f_1 = f_2 + (1 - f_3). \tag{115}$$

In the $(f_2 - f_1)$ plane, this is the equation of the line on which catastrophic error propagation can occur. It makes an angle of 45^{o} with the

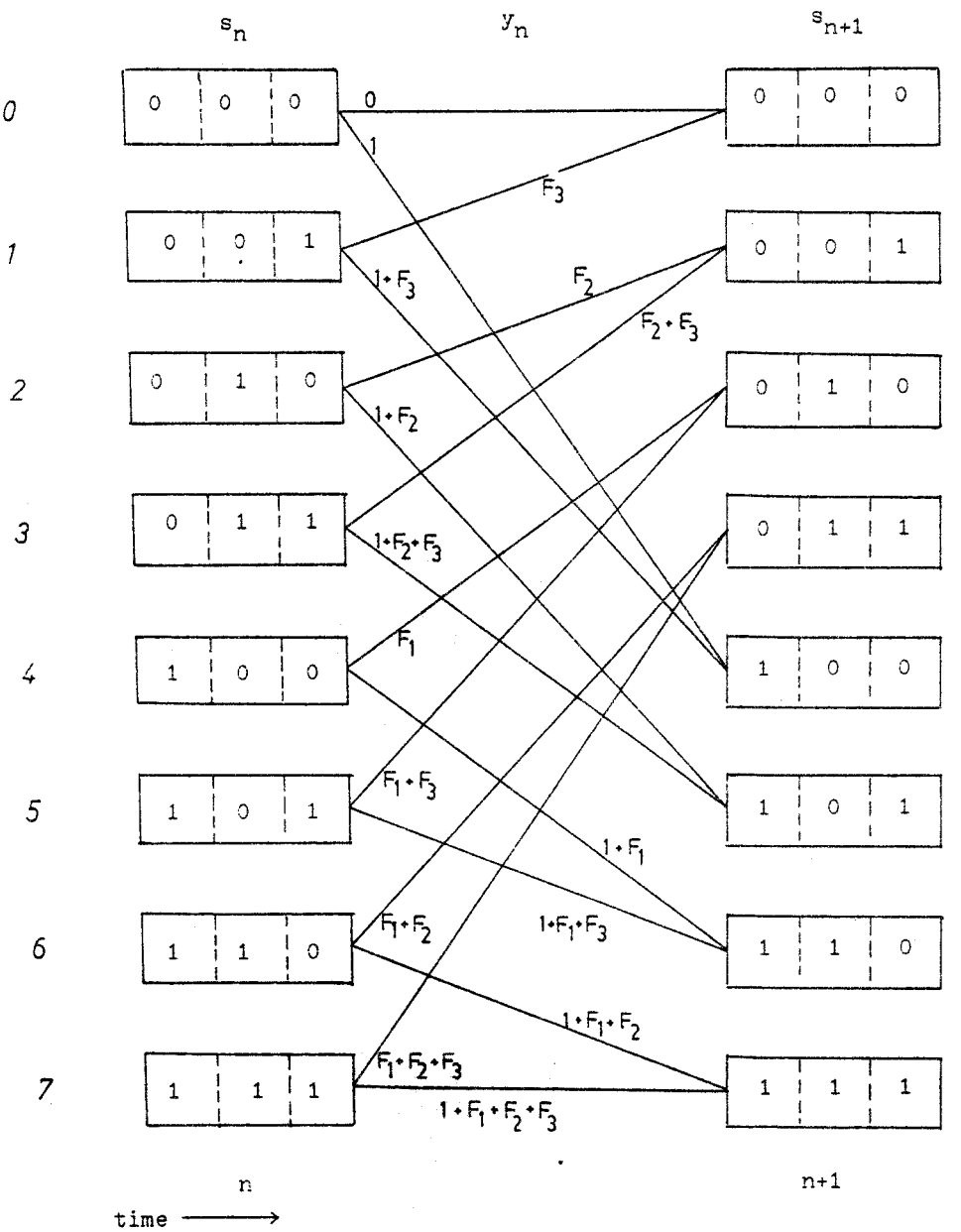


Fig. 6.4 A typical section of a state trellis of a three delays shift-register showing the output y_n in the transition from the state s_n to state s_{n+1}

f_2 axis. Computation results show that not all codes along the line have catastrophic error propagation. Near the origin of the $(f_2 - f_1)$ plane, there is no catastrophic error propagation. Therefore the line is split into two parts. For positive f_3's, one part is situated in the 1st quadrant and the other in the 3rd quadrant. As the duality of codes applies to decision depth, thus for negative f_3's, eq. (115) becomes

$$f_1 = -f_2 - (1 + f_3) \tag{116}$$

Consequently, the CE loci for negative f_3's are mirror reflections about the f_2 axis of their positive counterparts. The same observation applies to decision depth contours. Fig. 6.5 - Fig. 6.7 show the decision depth contours for f_3's = -3.1 -1.6 and -0.8 respectively. Fig. 6.8 - Fig. 6.10 show the decision depth contours for f_3's = 0.8, 1.6 and 3.1 respectively. Comparison of Fig. 6.5 with its corresponding positive counterpart in Fig. 6.10 will show the mirror reflection relationship.

Another dominant state sequence pair that causes catastrophic error propagation is

$$\begin{aligned} &....1\ \ 4\ \ 6\ \ 3\ \ 1.... \\ &....6\ \ 3\ \ 1\ \ 4\ \ 6..... \end{aligned} \tag{117}$$

with a period of T. The sequences, each with a period of 4T, are identical, but are displaced by 2T from each other. To find the locus of points for this pair of state sequences, observe that $y(1,4) = y(6,3)$ is equivalent to

$$1 + f_3 = (f_1 + f_2).$$

Computation results show that the locus of points for this sequence pair (117) passes through the 3rd quadrant where both f_1 and f_2 are negative; so in order to accomodate the right hand side of the above equation for positive f_3's, negation of the right hand side is required. Hence the equation becomes that of $y(1,4) = -y(6,3)$ and is equal to

$$1 + f_3 = -(f_1 + f_2) \tag{118}$$

Similarly, by referring to Fig. 6.4 and using the above reasoning, the equation for $y(4,6) = -y(3,1)$ is

$$1 + f_1 = -(f_2 + f_3) \tag{119}$$

Eqs. (118) and (119) are not independent but are equivalent. Putting them into a standard form, the locus of points along which catastrophic error propagation occurs for positive f_3's is

$$f_1 = -f_2 - (1 + f_3) \tag{120}$$

For negative f_3's, the equation becomes

$$f_1 = f_2 + (1 - f_3) \tag{121}$$

Computation results show that all codes along the lines of Eqs. (120) and (121) give rise to catastrophic error propagation. Thus it can be considered that eq. (120) and (121) are the necessary conditions for the occurence of catastrophic error propagation. The CE contours do not split

into two parts as in the case for the sequence pair (114). For positive f_3's, eq. (120) cuts across the 2nd, 3rd and 4th quadrants diagonally at 135^0 with the f_2 axis. For negative f_3's, eq. (121) cuts across the 1st, 2nd and 3rd quadrants diagonally at 45^0 with the f_2 axis being the mirror reflections along the f_2 axis of their positive counterparts. See Fig. 6.5 - Fig. 6.10.

Another interesting state sequence pair causing CE is

$$\begin{aligned} &.....4\ \ 6\ \ 3\ \ 1\ \ 4..... \\ &.....3\ \ 1\ \ 4\ \ 6\ \ 3.....,\end{aligned} \tag{122}$$

which can be considered as the state sequence pair (117) displaced by a time T. The necessary conditions for CE to occur are $y(4,6) = y(3,1)$ and $y(6,3) = y(1,4)$. They are equivalent to the following two equations:

$$1 + f_3 = (f_1 + f_2), \qquad 1 + f_1 = (f_2 + f_3).$$

Solving these two simultaneous equations gives the solution

$$f_2 = 1 \quad \text{and} \quad f_1 = f_3.$$

Thus the codes with catastrophic error propagation caused by the sequence pair (122) are given by:

$$1 \pm f_3 D + D^2 \pm f_3 D^3.$$

It is obvious from the solution that for positive f_3's, the sequence pair (122) will cause catastrophic error propagation in the 1st quadrant but

in the 4th quadrant for negative f_3's.

We found that the code $1 + 2.1D - 2.1D^2 + 3.1D^3$ and its dual have catastrophic error propagation. The state sequence pair which causes CE is found by noting that

$$f_2 + f_3 = 1, \quad f_1 + f_2 = 0 \text{ and } f_3 = 1 + f_1.$$

Then by inspection of Fig. 6.4, we get

$$\begin{aligned} y(3,1) &= f_2 + f_3 = 1 = y(0,4) \\ y(1,0) &= f_3 = 1 + f_1 = y(4,6) \\ y(0,0) &= 0 = f_1 + f_2 = y(6,3). \end{aligned}$$

We deduce that the state sequence pair is

$$\begin{aligned} &....3\ 1\ 0\ 0\ 4\ 6\ 3..... \\ &....0\ 4\ 6\ 3\ 1\ 0\ 0..... \end{aligned} \tag{123}$$

with a period of 3T. Each sequence has a period of 6T. This is shown in the 2nd quadrant of Fig. 6.10 as a cross with a decision depth contour of 30 surrounding it. Its dual is shown in Fig. 6.5

6.4 Comments on the Equations for Catastrophic Error Propagation

Let us consider the linear equations for different f_3's along which catastrophic error propagation occurs.

1) $f_3 = \pm 0.8$

For $f_3 = 0.8$, the equation for CE due to the sequence pair (114) is

is

$$f_1 = f_2 + 0.2 \tag{124}$$

according to eq. (115). While for $f_3 = -0.8$, the equation becomes

$$f_1 = -f_2 - 0.2 \tag{125}$$

according to eq. (116).

The equation for CE due to the state sequence pair (117) is

$$f_1 = -f_2 - 1.8, \tag{126}$$

for positive f_3. While for $f_3 = -0.8$ according to eq. (121), the equation becomes

$$f_1 = f_2 + 1.8. \tag{127}$$

Note that eq. (124) and eq. (125) are mirror reflections of each other along the f_2 axis, a manifestation of the duality of PRS systems. The same relationship applies to eq. (126) and eq. (127). The isolated code with CE due to the state sequence pair (122) is $1 + 0.8D + D^2 + 0.8D^3$ for positive f_3, with its dual $1 - 0.8D^2 + D^2 - 0.8D^3$ for negative f_3. All of the above can be seen in Fig. 6.7 and Fig. 6.8.

2) $f_3 = \pm 1.6$

In the same way as above, the equations of the loci of points with CE due to the sequence pair (114) for positive and negative f_3's are respectively

$$f_1 = f_2 - 0.6 \quad \text{and} \quad f_1 = -f_2 + 0.6, \tag{128}$$

while the equations for CE due to the sequence pair (117) for positive and negative f_3's are respectively

$$f_1 = -f_2 - 2.6 \quad \text{and} \quad f_1 = f_2 + 2.6. \tag{129}$$

The code with CE due to the sequence pair (122) is $1 + 1.6D + D^2 + 1.6D^3$ for positive f_3, with its dual $1 - 1.6D + D^2 - 1.6D^3$ for negative f_3. See. Fig. 6.6 and Fig. 6.9.

3) $f_3 = \pm 3.1$

The equations for CE due to the sequence pair (114) for positive and negative f_3's are

$$f_1 = f_2 - 2.1 \quad \text{and} \quad f_1 = -f_2 + 2.1 \tag{130}$$

respectively. The equations for CE due to the sequence pair (117) for both positive and negative f_3's are

$$f_1 = -f_2 - 4.1 \quad \text{and} \quad f_1 = f_2 + 4.1 \tag{131}$$

respectively.

Similarly, the codes with CE due to the sequence pair (122) are $1 \pm 3.1D + D^2 \pm 3.1D^3$. Also as noted in section 6.3, the code $1 + 2.1D - 2.1D^2 + 2.1D^3$ causes catastrophic error propagation due to the sequence pair (123).

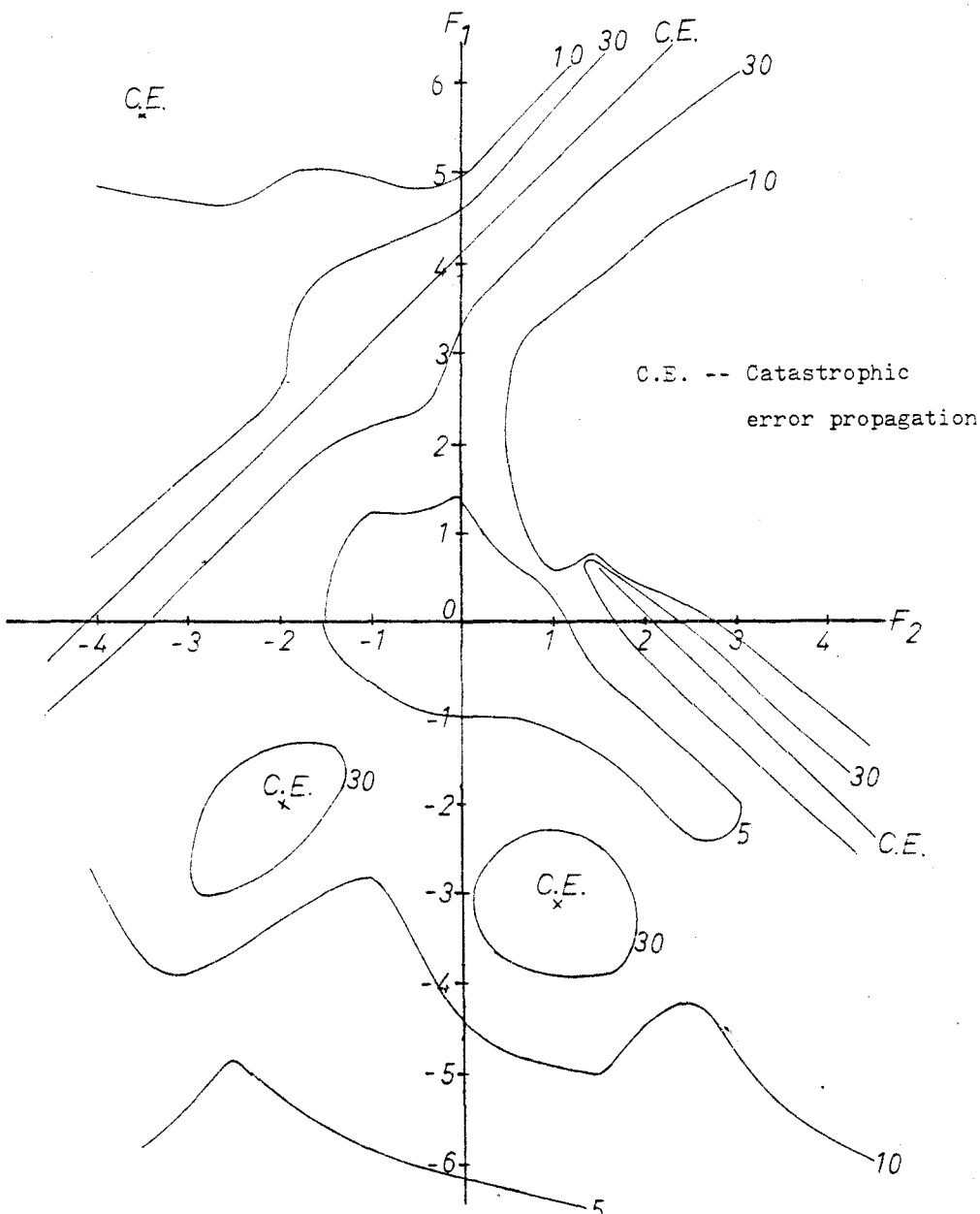


Fig. 6.5 Contours of constant decision depth

$F_0 = 1$, $F_3 = -3.1$.

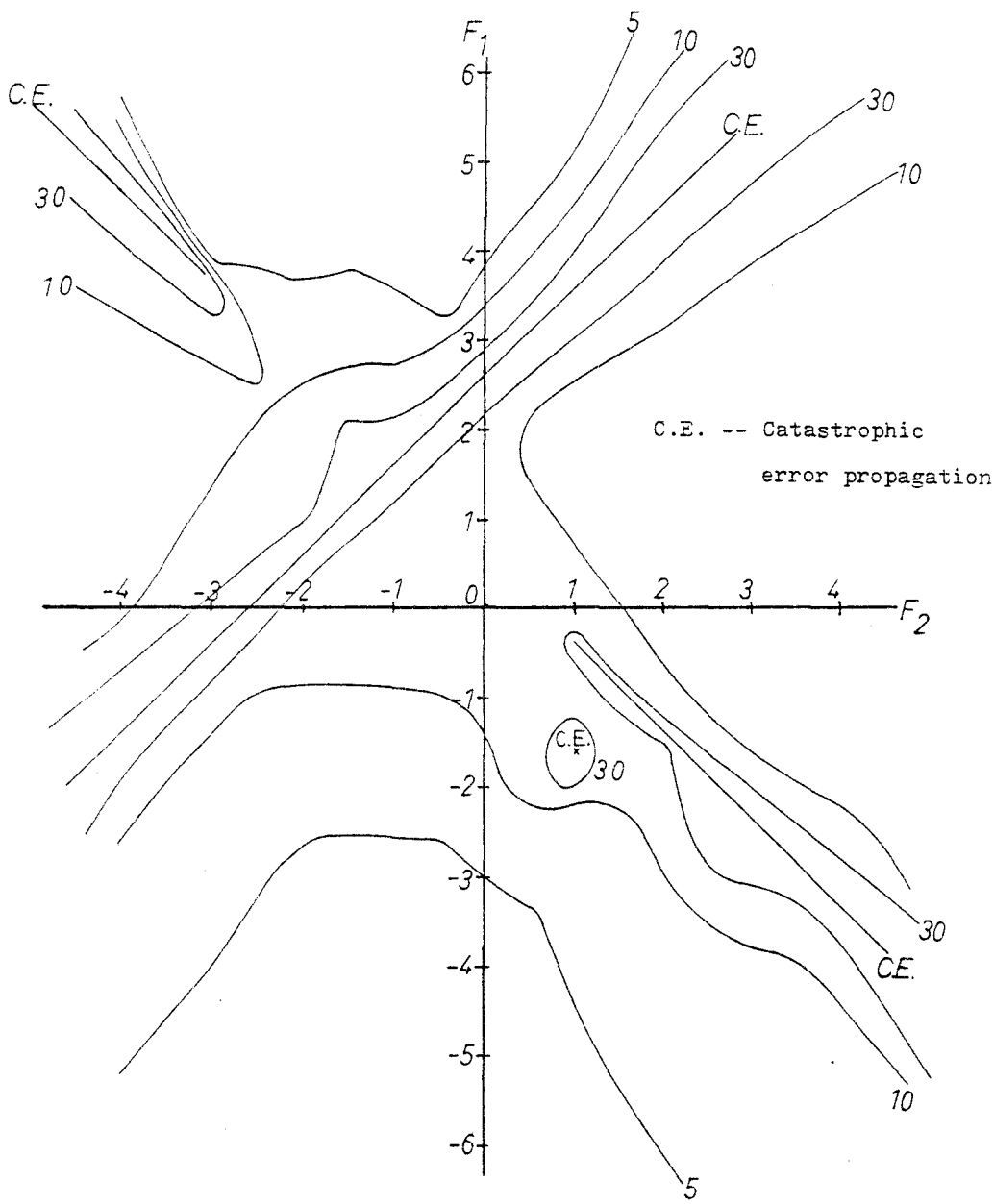


Fig. 6.6 Contours of constant decision depth

$F_0 = 1$, $F_3 = -1.6$.

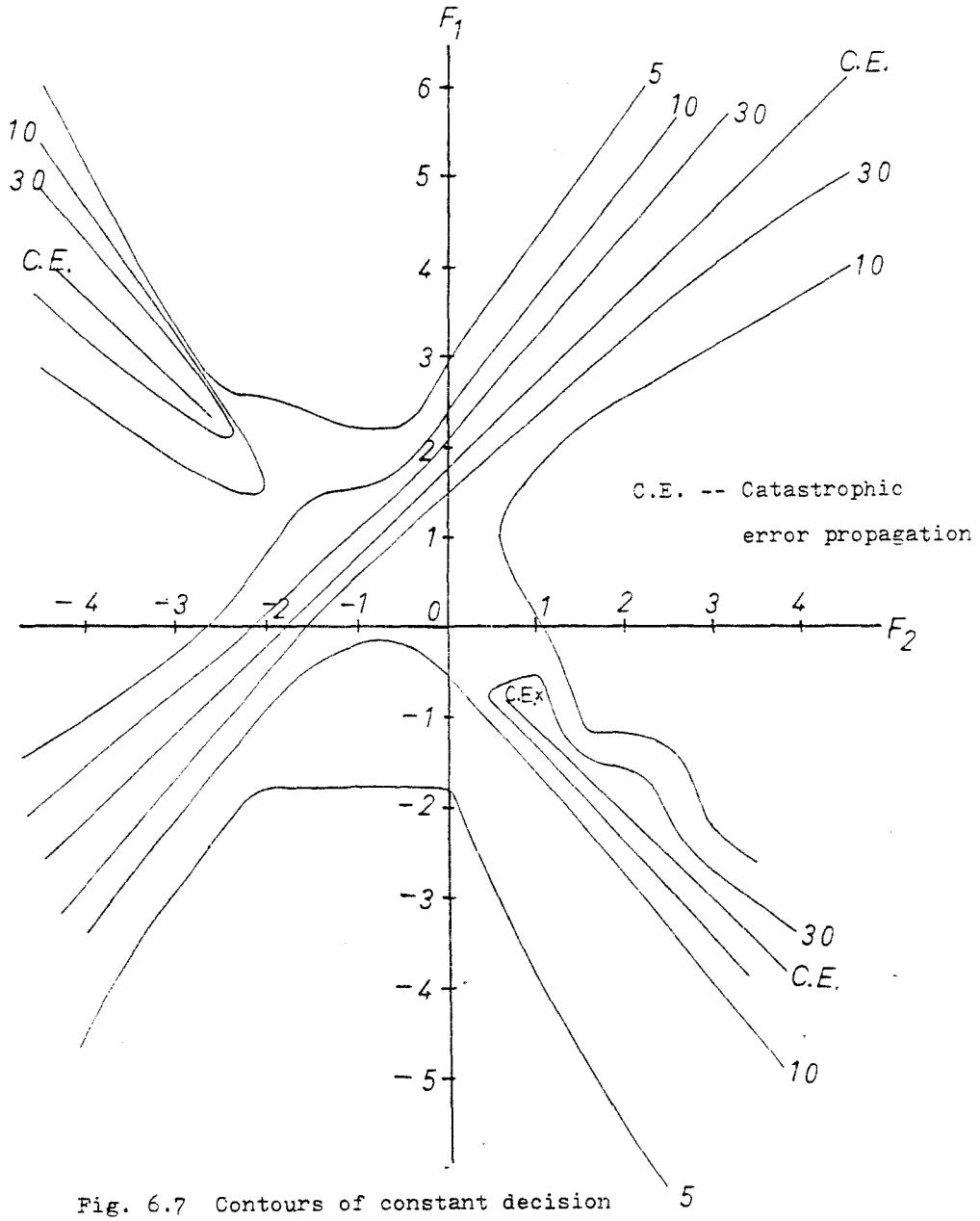


Fig. 6.7 Contours of constant decision depth; $F_0 = 1$, $F_3 = -0.8$.

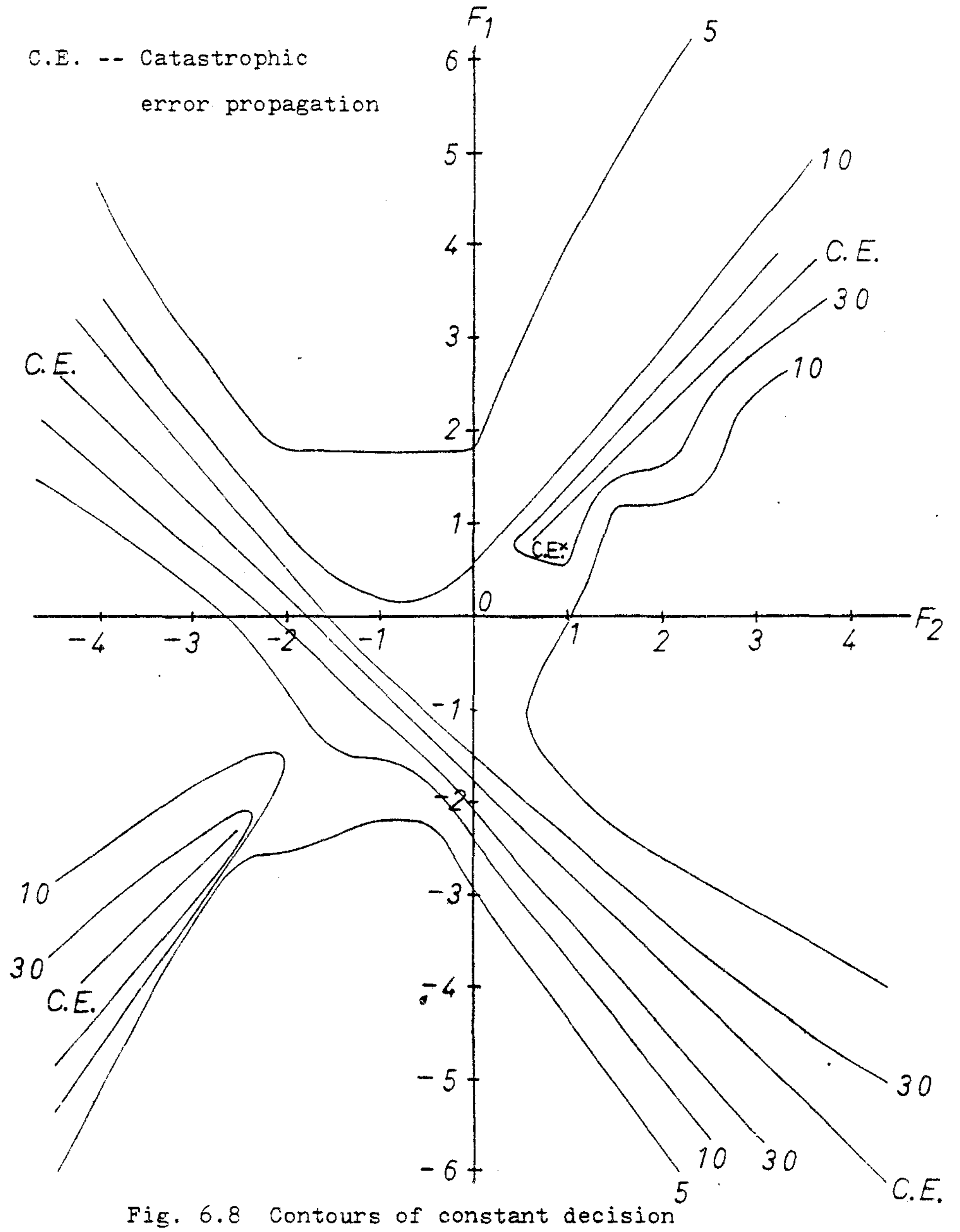


Fig. 6.8 Contours of constant decision depth; $F_0 = 1$, $F_3 = 0.8$.

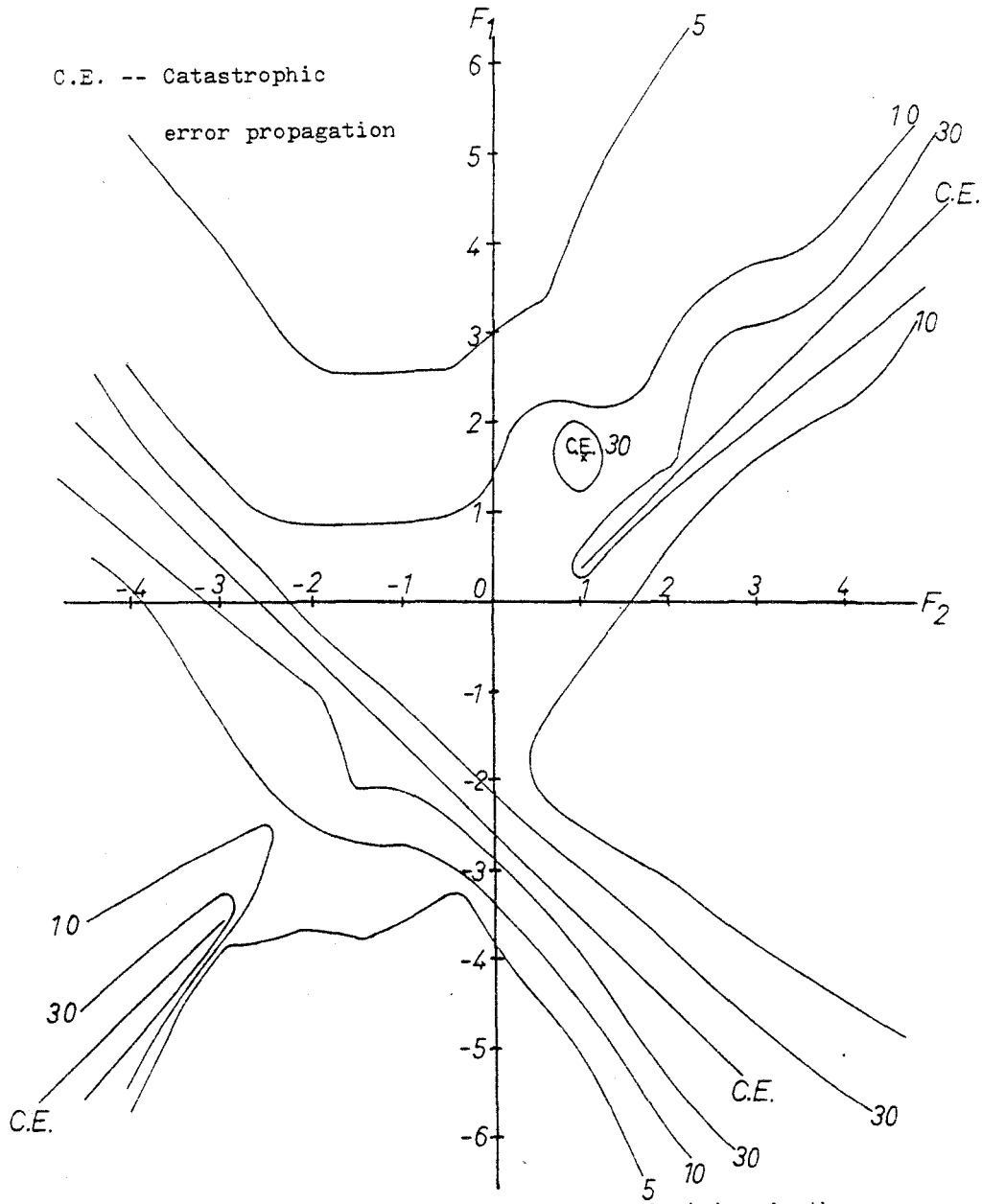


Fig. 6.9 Contours of constant decision depth

$F_0 = 1$, $F_3 = 1.6$.

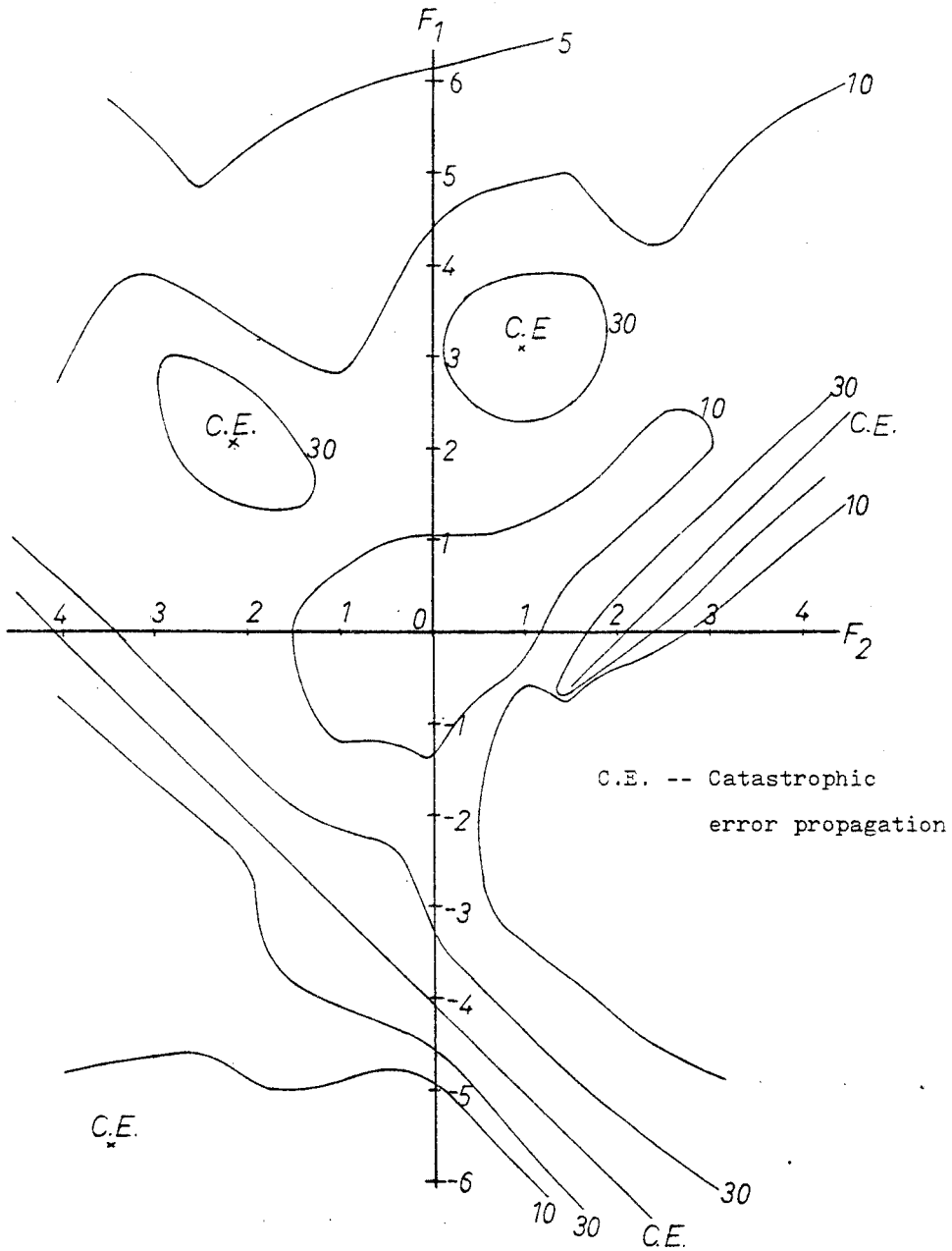
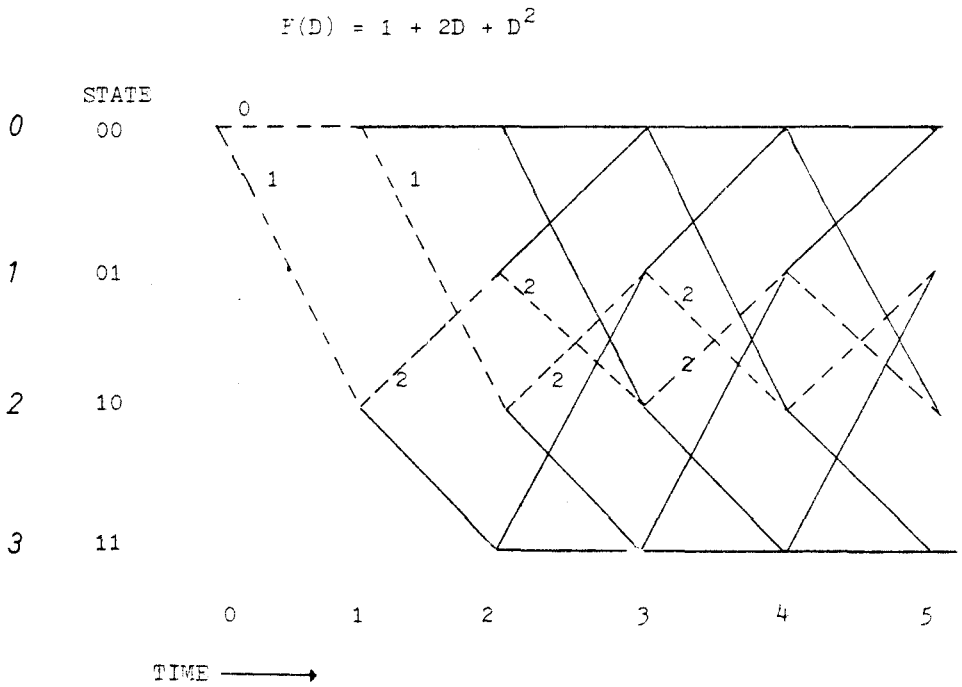


Fig. 6.10 Contours of constant decision depth

$F_0 = 1$, $F_3 = 3.1$



----- The pair of state sequences causing catastrophic error propagation. The output level for each transition up to t = 2 is shown. The output level from t = 2 onwards stays the same for each transition.

Fig. 6.11 The state trellis of a two delays PRS system showing the state sequence pair causing catastrophic error propagation.

CHAPTER 7

99% ENERGY BANDWIDTH

This Chapter reports on the 99% power bandwidth of PRS codes with channel length $K \leq 4$ using binary signalling. Although the codes have a well defined single-sided bandwidth of π/T radians, where T is the sampling interval, it is interesting and useful to determine just how much bandwidth is conserved merely by sacrificing 1% of the total energy at the high frequency end of the spectrum.

7.1 99% Energy Bandwidth of PRS systems

99% energy bandwidth is the frequency band within which 99% of the energy lies. The frequency response of a digital transversal filter is the discrete Fourier transform of the code F(D) and is given by substituting the delay operator D with the quantity $\exp(-j\omega iT)$;i.e.,

$$\begin{aligned} H(\omega) &= F(D)\,|_{D = \exp(-j\omega iT)} \\ &= \sum_{i=0}^{L} f_i \exp(-j\omega iT) \\ &= 1 + \sum_{i=1}^{L} f_i \exp(-j\omega iT), \end{aligned} \tag{134}$$

with $f_0 = 1$, $j^2 = -1$, $\omega = 2\pi f$ where f is used to denote frequency, i is an integer, and L is the number of delays of the filter.

The total energy confined from $-\alpha$ to α Hertz is equal to the integration of the energy density over the frequency range. One point to

note is that although $H(\omega)$ is periodic, that is, $H(\omega) = H(\omega - 2\pi k/T)$ for integer k, $H(\omega)G(\omega)$ is non-periodic if $G(\omega)$ is an ideal low-pass filter of 2-sided bandwidth $2\pi/T$ radians. Thus the term energy density is more appropriate for this non-periodic frequency response. The energy density is given by the product of $H(\omega)G(\omega)$ and its complex conjucate $H^*(\omega)G^*(\omega)$. But $G^*(\omega)$ is equal to one within the bandwidth of $2\pi/T$, thus the energy density is

$$H(\omega)H^*(\omega) = (1 + \sum_{i=1}^{L} f_i \exp(-j\omega iT))(1 + \sum_{i=1}^{L} f_i \exp(+j\omega iT)), \quad |\omega| < 2\pi/T.$$

The energy within the frequency band $|\alpha|$ is

$$B = 1/\pi \{ \int_0^{2\pi\alpha} |H(\omega)|^2 \, d\omega \}. \tag{135}$$

Refer to Fig. 7.1

It is shown in appendix B.2 that the total energy of a PRS code of form F(D) is

$$R(0) = ||F(D)||^2 = 1 + \sum_{i=1}^{L} f_i^2. \tag{136}$$

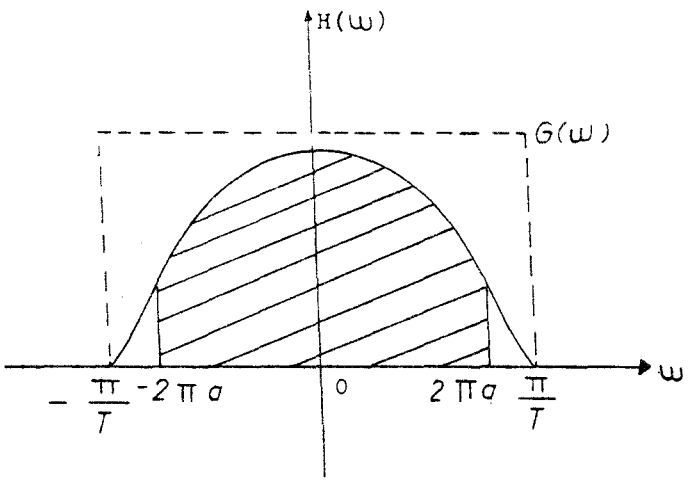
Thus the equation governing the 99% energy bandwidth is

$$B/R(0) = 0.99. \tag{137}$$

For K = 3, the equation for B becomes [Appendix B.1].

$$B = 1/\pi \int_0^{2\pi\alpha} (1 + f_1^2 + f_2^2 + 2f_1\cos\omega + 2f_1 f_2 \cos\omega + 2f_2\cos 2\omega) d\omega$$

$$= 2\alpha(1 + f_1^2 + f_2^2) + \frac{2f_1}{\pi} \sin 2\pi\alpha + \frac{2f_1 f_2}{\pi} \sin 2\pi\alpha + \frac{2f_2}{\pi} \sin 4\pi\alpha. \tag{138}$$



$$\frac{\frac{1}{\pi}\int_{0}^{2\pi\sigma} |H(\omega)\cdot G(\omega)|^2 d\omega}{1+\sum_{k=1}^{L} F_k^2} = 0.99$$

σ --- 99% energy bandwidth for PRS systems in Hertz

$H(\omega)$ -- discrete Fourier transform of a PRS system

$G(\omega)$ -- unity gain low pass filter of bandwidth $2\pi/T$

Fig. 7.1 Formulation of the 99% energy bandwidth of a PRS system

Substituting B in eq. (137), we have

$$(1 + f_1^2 + f_2^2)(2\alpha - 0.99) + \frac{2f_1}{\pi} \sin 2\pi\alpha + \frac{2f_1 f_2}{\pi} \sin 2\pi\alpha + \frac{2f_2}{\pi} \sin 4\pi\alpha = 0. \tag{139}$$

Finding the minimum bandwidth involves minimization over all the tap gains of α, within the range $0 < \alpha < 1/2T$ and satisfying the above equation. This minimization, using variational calculus techniques, involves a highly non-linear operation, and the periodic nature of the sine function adds to the difficulty. We instead compute 99% energy bandwidth of each codes using eq. (137), and draw contours of all codes with a given bandwidth. A contour line of 80%, for example, means that 99% of the energy of all codes on the line is confined within 80% of the minimum Nyquist bandwidth of π/T.

7.2 99% Energy Bandwidth Contours

The bandwidth contours as f_3 ranges from -1.6 to +3.1 are shown in Fig. 7.2 to Fig. 7.7. Some facts observed from these plots are now summarized.

1) For $f_3 = -1.6$, the minimum bandwidth is around 72.7% and resides in the 2nd and 3rd quadrants. The region enclosed by the 90% contours stretches across the 1st, 2nd and 3rd quadrants.

2) For $f_3 = -0.8$, the minimum is around 74%, and its position has moved diagonally upwards occupying the 1st and 2nd quadrants.

3) For $f_3 = 0.0$, which corresponds to K = 3, the minimum bandwidth is around 63.6%. Both the 70% and 80% contours have moved into the 1st quadrant.

4) For $f_3 = 0.8$, the minimum bandwidth decreases to 52.7% This represents the minimum bandwidth for all f_3 considered in this work and thus approximates the minimum for K = 4. The region enclosed by the 90% contour has split into two separate portions, one in the 1st quadrant and the other in the 3rd quadrant.

5) For $f_3 = 1.6$, the minimum bandwidth is around 53.5%. The 80% or less contours in the 1st quadrant have moved diagonally further away from the origin.

6) For $f_3 = 3.1$, the minimum bandwidth is 59.2%. The migration of the narrow bandwidth regions away from the origin is such that the 60% contour is almost off the plot in the (f_2, f_1) plane.

7) For K = 2, $f_3 = f_2 = 0.0$, the minimum bandwidth is around 81.65%. For the binary PAM case, $f_3 = f_2 = f_1 = 0.0$, the 99% bandwidth is 99%.

The migration of the region enclosed by the 90% contour away from the origin as f_3 increases implies that as f_3 increases, f_1 and f_2 have to increase proportionally to give the same 99% energy bandwidth. Also notice that all narrow bandwidth contours reside in the 1st quadrant. This can be explained by observing the z-transform of codes having all positive taps: their zeroes are all on the left-half z-plane so that more energy is concentrated in the low frequency portion of the spectra. For K = 4, the narrowest bandwidth region is in the 1st quadrant with the second best region of around 90% in the 3rd quadrant.

The minimum 99% energy bandwidth for K = 2 is 81.65% and it occurs around code 1 + D. Its z-transform has a zero at z = -1. For K = 3, the

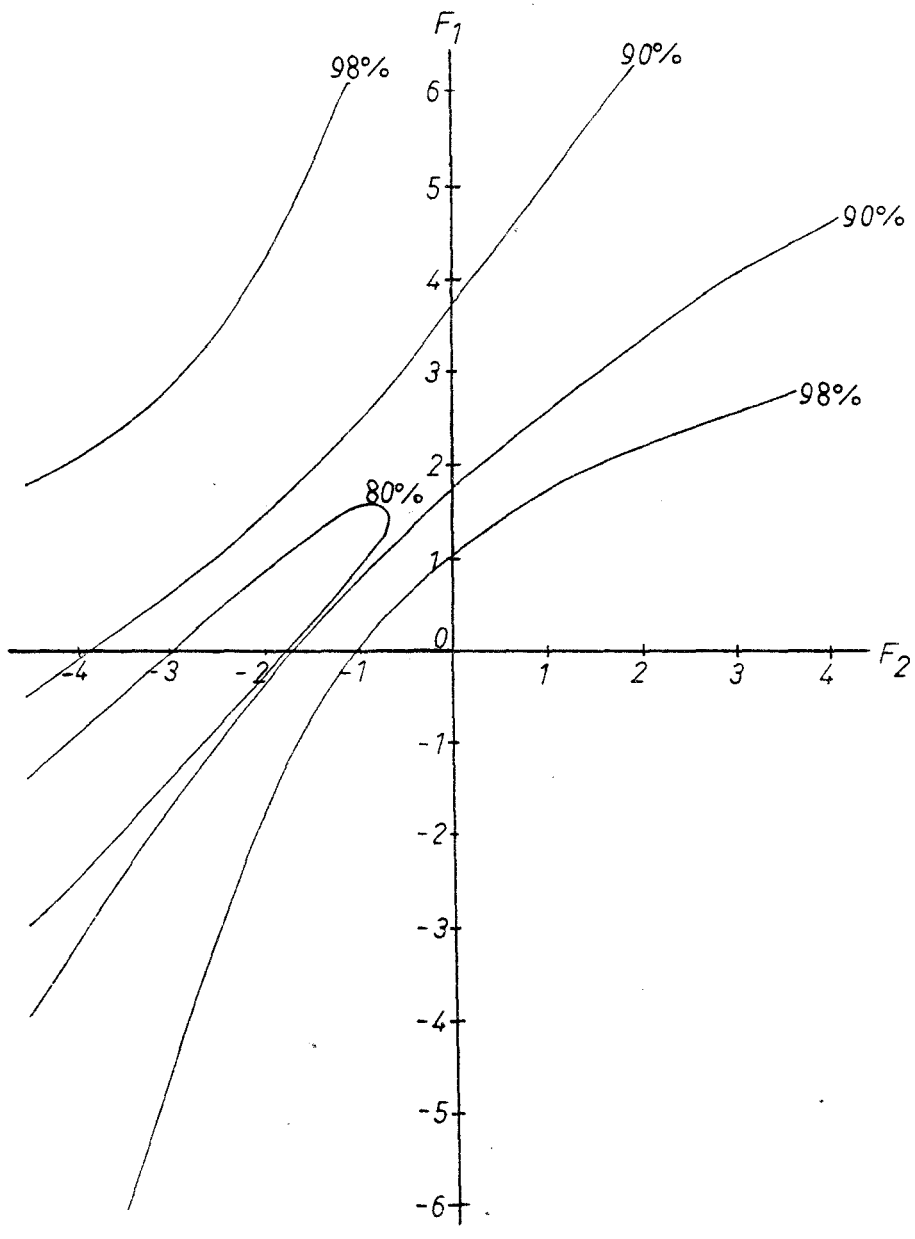


Fig. 7.2 Contours of constant 99% energy bandwidth.

$F_0 = 1$, $F_3 = -1.6$.

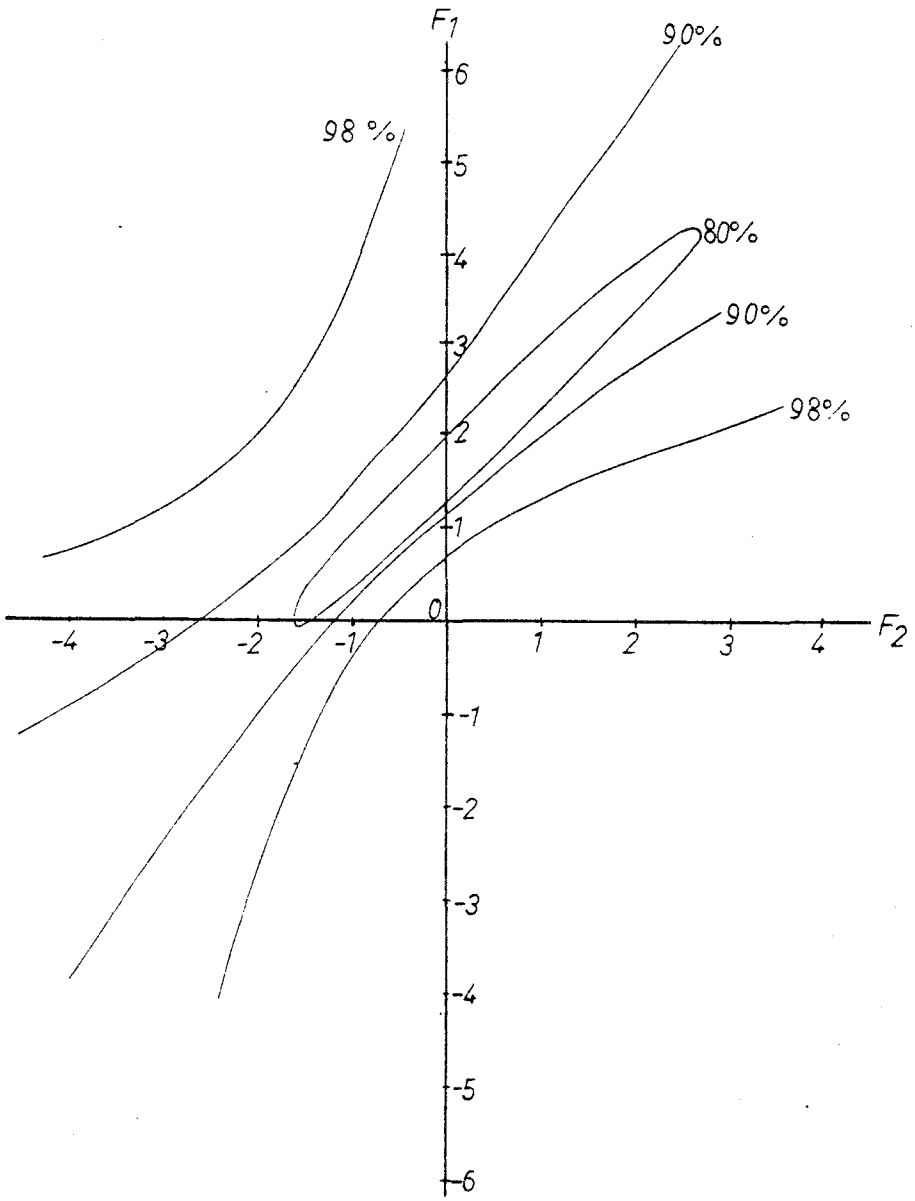


Fig. 7.3 Contours of constant 99% energy bandwidth

$F_0 = 1$, $F_3 = -0.8$.

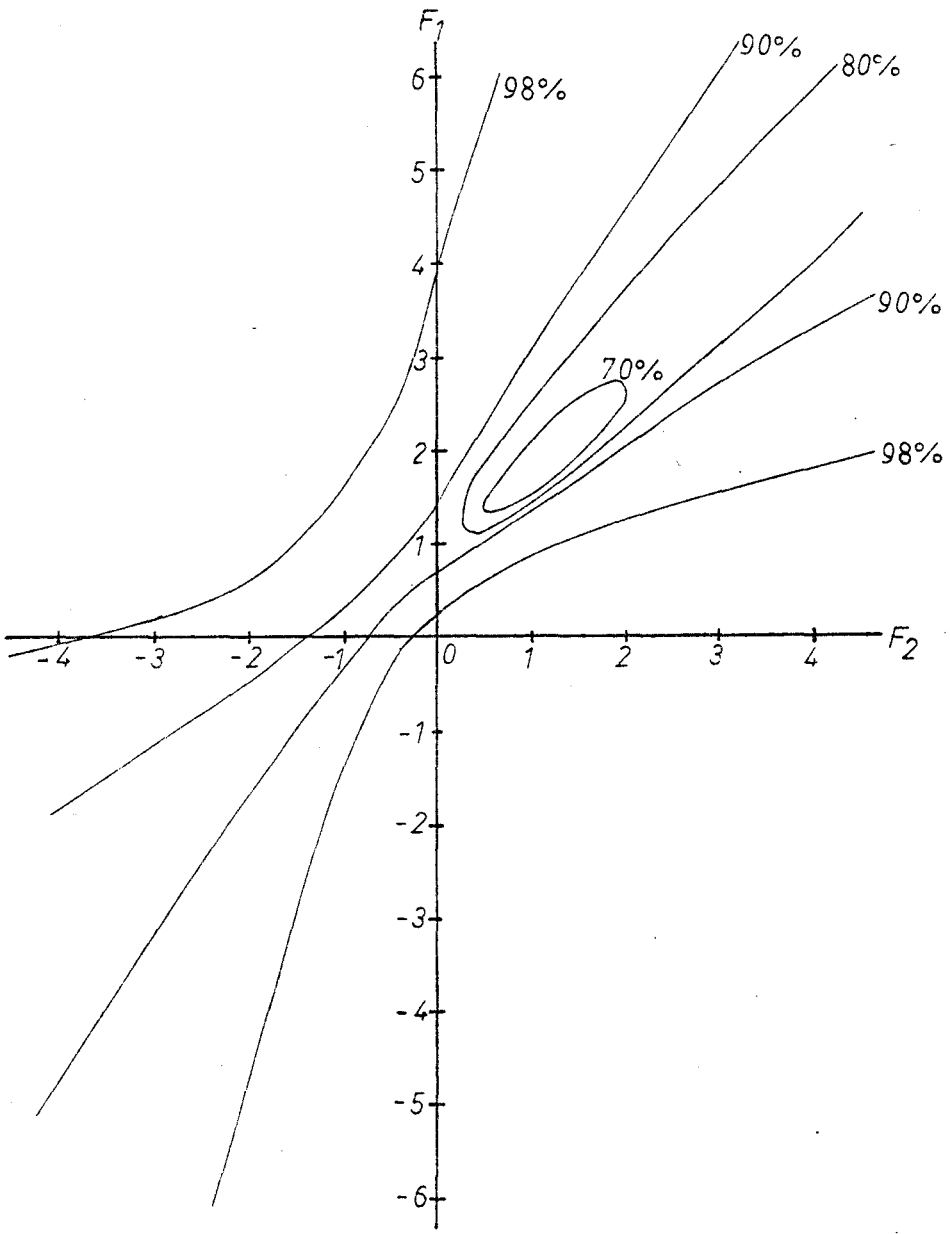


Fig. 7.4 Contours of constant 99% energy bandwidth

$F_0 = 1$, $F_3 = 0.0$.

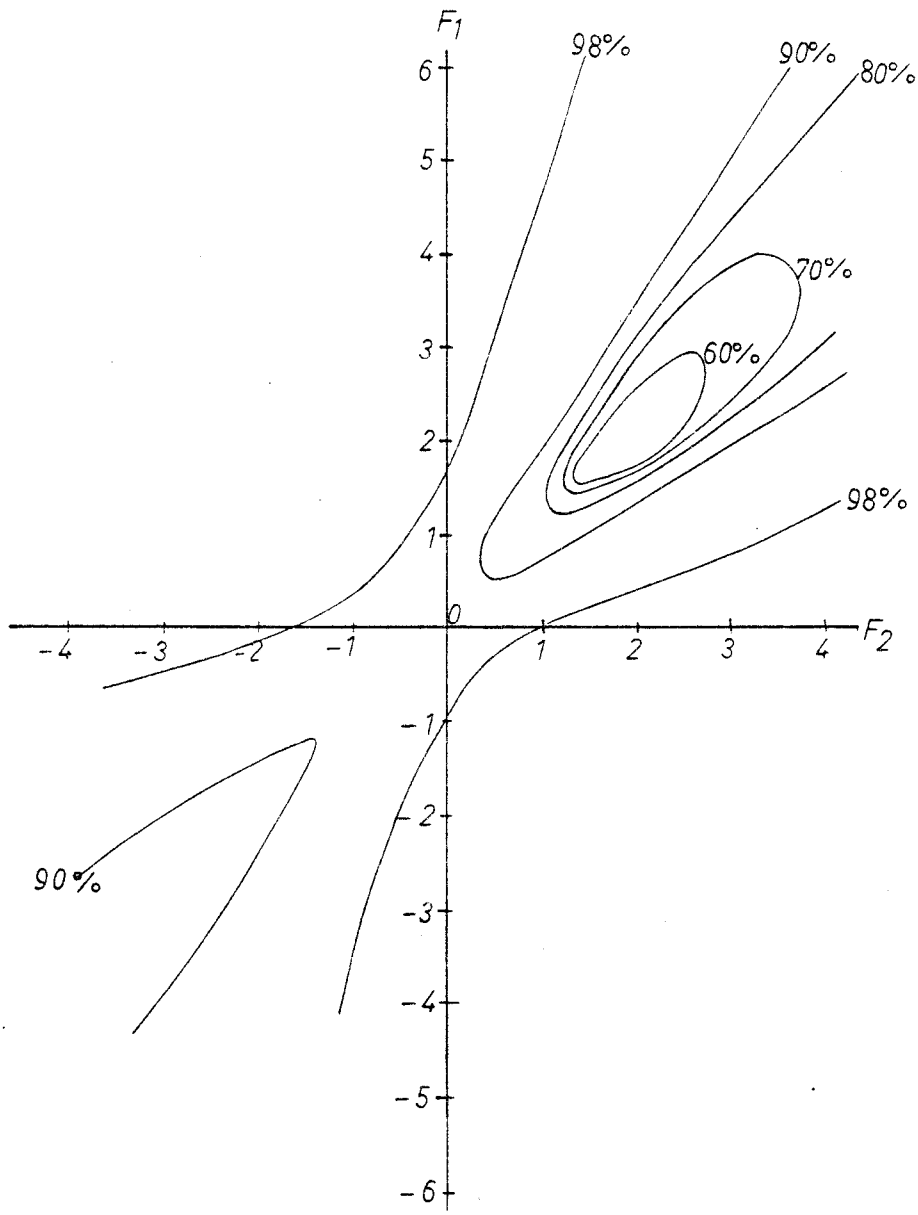


Fig. 7.5 Contours of constant 99% energy bandwidth

$F_0 = 1$, $F_3 = 0.8$.

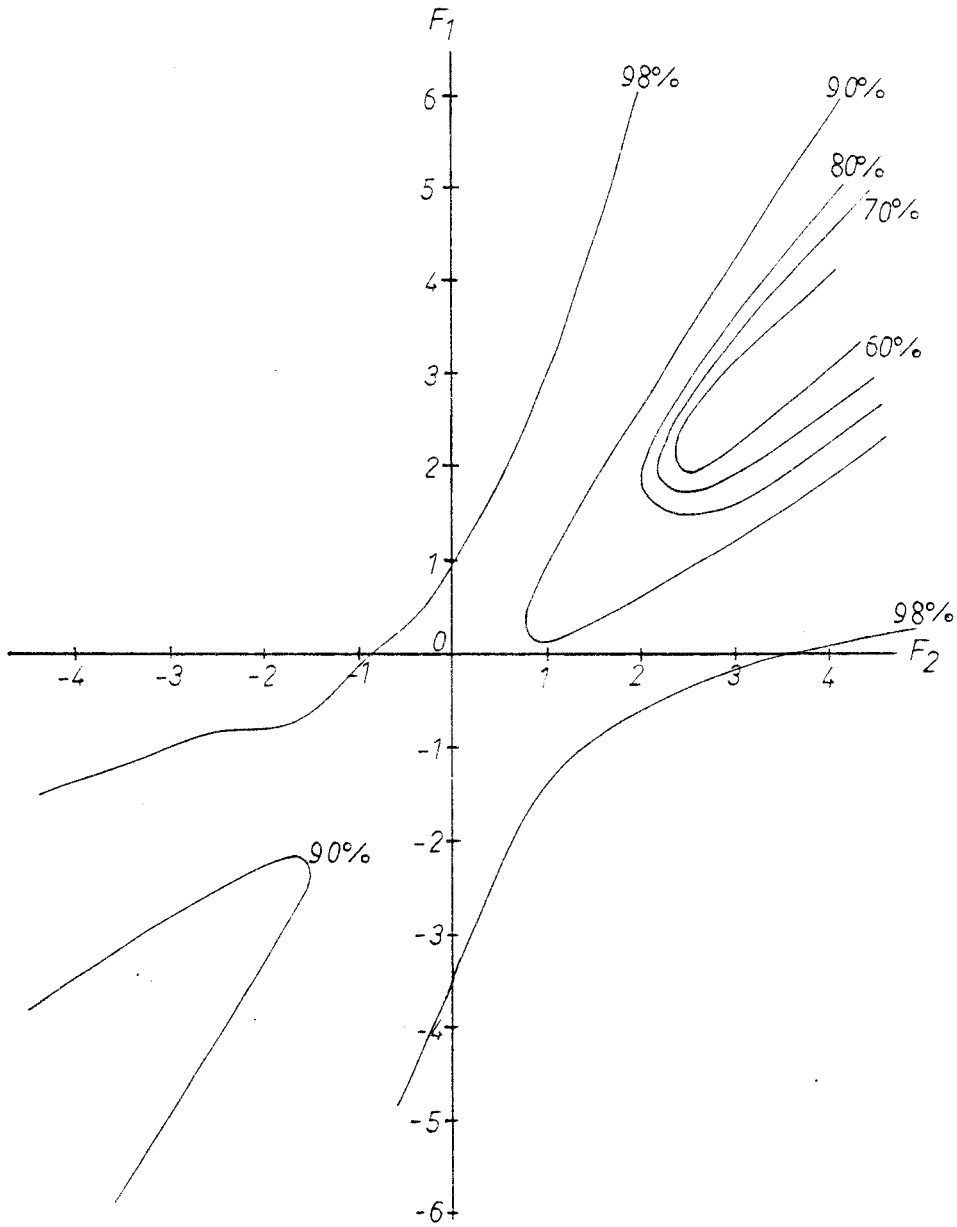


Fig. 7.6 Contours of constant 99% energy bandwidth

$F_0 = 1$, $F_3 = 1.6$.

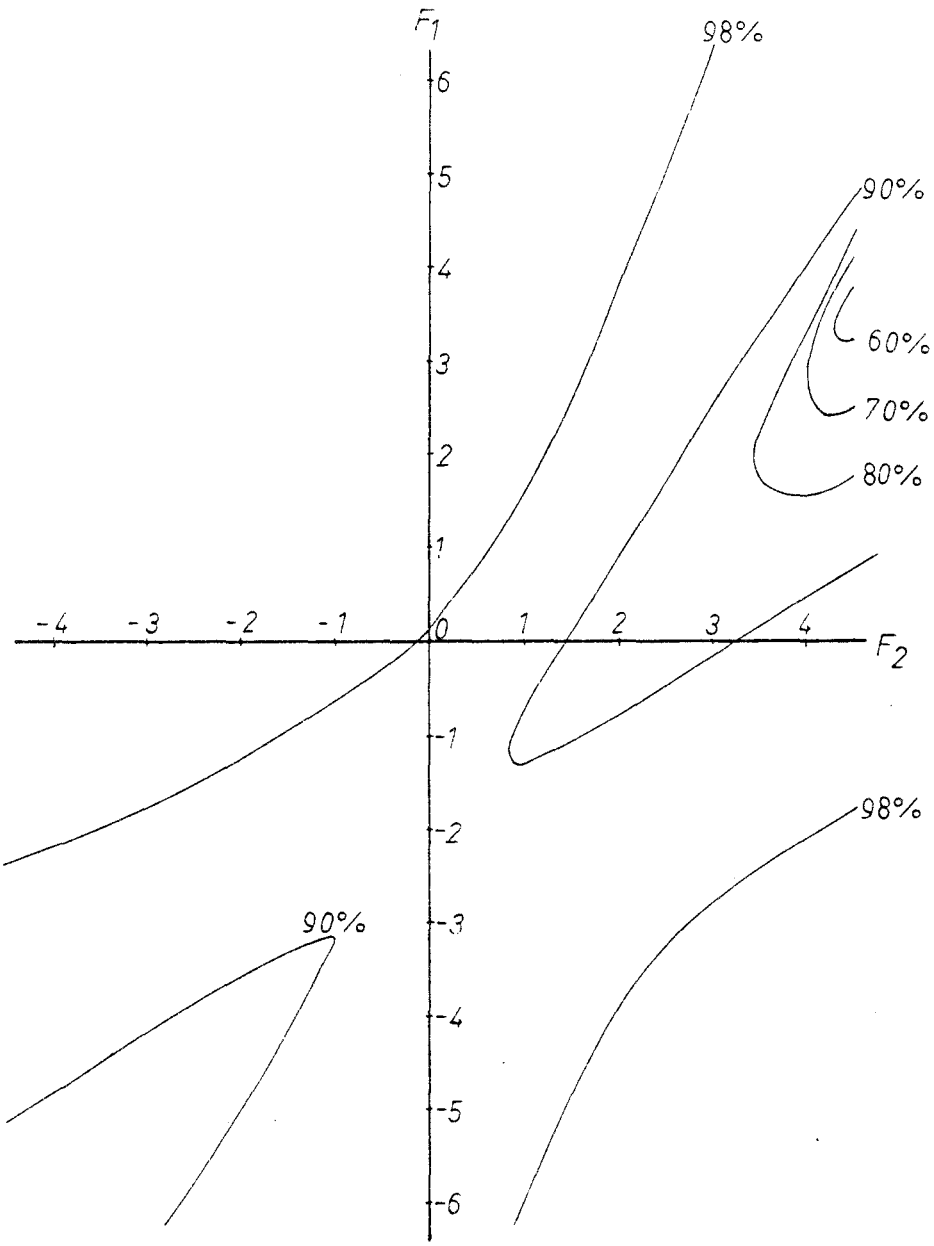


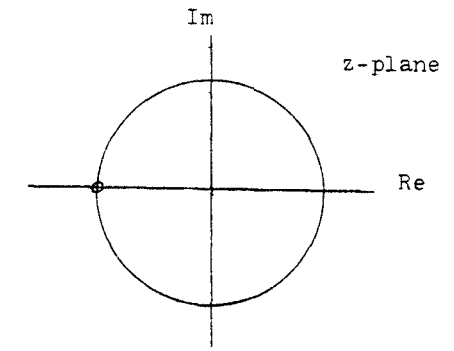
Fig. 7.7 Contours of constant 99% energy bandwidth

$F_0 = 1$, $F_3 = 3.1$.

minimum bandwidth is around 63.6% and occurs at around code $1 + 1.6D + D^2$. Its z-transform is $(z^2 + 1.6z + 1)/z^2$. This has zeroes at $-0.8 \pm j0.6$; i.e., all zeroes are on the left-hand z-plane.

For K = 4, the minimum bandwidth is around 52.7% and it occurs around code $1 + 2D + 2D^2 + 0.8D^3$ for the different f_3's considered in this work. For convenience sake, let the minimum occur at code $1 + 2D + 2D^2 + D^3$. The zeroes are thus -1, $-0.5 \pm \frac{j1.732}{2}$, all on the left-half z-plane.

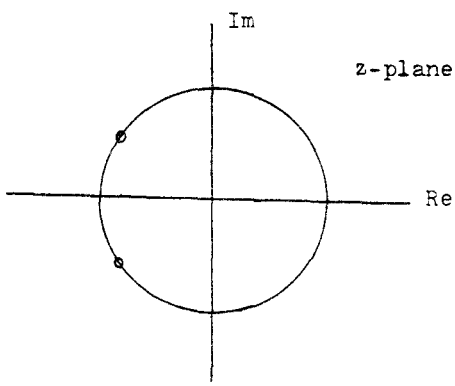
Considering the case where all tap-gains are positive, we note that as constraint length K increases, the z-transform of F(D) has more zeroes in the left-hand z-plane. This implies that the amplitude of the high frequency response is pulled down by more zeroes resulting in higher energy in the low frequency portion of the spectrum. Thus we expect the minimum bandwidth to further decrease as the number of taps increases. Fig. 7.8 shows the zeroes of the z-transforms of the PRS systems which have the narrowest 99% energy bandwidths for K = 2, 3 and 4.



$$F(D) = 1 + D$$

$$Z[F(D)] = \frac{z + 1}{z}$$

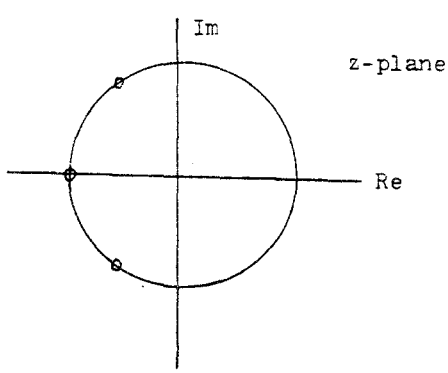
zero = -1



$$F(D) = 1 + 1.6D + D^2$$

$$Z[F(D)] = \frac{z^2 + 1.6z + 1}{z^2}$$

zeroes = $-0.8 \pm j0.6$



$$F(D) = 1 + 2D + 2D^2 + D^3$$

$$Z[F(D)] = \frac{z^3 + 2z^2 + 2z + 1}{z^3}$$

zeroes = $-1, -0.5 \pm \frac{j1.732}{2}$

Fig. 7.8 Locations of zeroes in the z-plane for PRS systems with minimum bandwidth of constraint length 2, 3, 4.

CHAPTER 8

A COST FUNCTION AND ITS EVALUATION

By comparing both the degradation and bandwidth contours, we note that as a general rule, each plot has a region of greatest degradation near the region of narrowest bandwidth. Thus, the aim of simultaneously minimizing both degradation and bandwidth leads to a fundamental conflict. In order to trade off the above two quantities, we define a channel penalty function. A rationale for the function is given. The rest of the chapter reports on its evaluation.

8.1 Shannon Capacity Equation

Consider the Shannon channel capacity equation,

$$C = W\log_2(1 + P/N_0W) \tag{140}$$

where W is the effective one-sided bandwidth of the channel in Hertz, P is the average transmitted power, and N_0 is the one-sided power spectral density of white Gaussian thermal noise. This equation implies that is is possible to transmit digital information over a channel of bandwidth W at a rate $R < C$ with arbitrarily small probability of error by using a sufficiently sophisticated coding system, but it is impossible to send information at a rate $R > C$ without a finite error rate [16]. Several points to note are: first, the equation only applies to white Gaussian

noise under average signal power limitation;second, in order to approximate the limiting rate of transmission, the signals must to likened in statistical properties to white noise. This is due to the fact that the received signals will have a maximum entropy if they also form a white noise ensemble, since noise has the greatest entropy for a given power [17]. Third, from the channel capacity equation, we see the unfavourable exchange in the signal power in order to reduce the bandwidth used for an ideal system: an increase of the signal power by 2^n fold will only reduce the required bandwidth by a ratio of n. For example, in a PCM system the power to bandwidth trade-off follows the logarithmic relationship of an ideal system, but requires about eight times the power theoretically needed to realize a given channel capacity for a given bandwidth. Practical systems using various digital and digitized-analog modulation techniques display the power-bandwidth trade-off in a variety of shapes, and the trade-off relationship may not be logarithmic as in the ideal system.

8.2 Rationale for a Cost Function

We can derive the relationship between degradation and bandwidth as follows for PRS systems with constraint length K = 3. The equation of the contour with constant d^2_{norm} is elliptical and is given by eq. (94)

$$(8 - x)f_1^2 - 8f_1f_2 + (8 - x)f_2^2 - 8f_1 + (8 - x) = 0,$$

where $x = d^2_{norm}$.

From this we have $1 + f_1^2 + f_2^2 = 8(f_1 + f_1 f_2)/(8 - x)$.
Substituting the above equation into eq. (139) gives

$$\frac{8(f_1 + f_1 f_2)}{(8 - x)}(2\alpha - 0.99) + \frac{2f_1 \sin 2\pi\alpha}{\pi} + \frac{2f_1 f_2 \sin 2\pi\alpha}{\pi} + \frac{2f_2 \sin 4\pi\alpha}{\pi} = 0 \qquad (140)$$

In eq. (140), α and x are the dependent variables while f_1 and f_2 are the independent variables. The region where both minimum degradation and bandwidth occurs can be found by maximizing x and minimizing α in eq. (140) over tap-gains f_1 and f_2 subject to the constraint of

$$|\alpha| < 1/2T \quad \text{and} \quad x < 4.$$

The same derivation can be applied to different constraint lengths K. We found that the optimization exercise presented above is rather difficult due to the sinusoidal terms, and does not give intuition into the trade-off of power and bandwidth in PRS systems. Rather we choose to approach this optimization problem by defining a cost or penalty function that jointly takes both power and bandwidth into appropriate account for PRS systems.

Generally, we provide resources to get certain returns. Surely, we would like to maximize our return for a given resource. But if the return is not "accumulative", then we want the return to be as close to the needs as possible so that nothing will be wasted. "Accumulative" refers to the property that certain return can be accumulated over time,

for example, a physical entity like energy and metal. If one is not able to fully utilize these entities now, they can be stored and used later on to a certain extent. But there exists certain entities like speed or bandwidth that cannot be stored and used over a period of time. Either one fully utilizes these entities immediately or else the unused capacity will be lost. In an ideal communication system, we are charged for channel capacity directly proportional to the bandwidth occupied and to the logarithm of the power used in signalling for a fixed guaranteed probability of error. Following the line of thoughts expressed above on resources and returns, we would like to choose a PRS system that will provide a capacity close to the required signalling rate. Because any capacity higher than the signalling rate will be wasted as channel capacity is one example of a return that is not accumulative. The penalty for choosing a system with capacity lower than the signalling rate is the frequent occurrence of errors, while the penalty for choosing a system with higher capacity than the required signalling rate is the additional cost imposed for power and bandwidth above that necessary for ths signalling rate.

8.3 Average Transmitted Power of PRS Systems

In order to compare different PRS systems, we have to assure that the energies of all output signals are equal. To do so, we can either fix the gain of the low-pass filter or set the amplitude of the data inputs to a certain value so that the energy of the impulse response of the low-pass filter stays the same. We choose to set the gain of the low-pass filter to $\sqrt{E_s T}$; the energy of each unit pulse response then becomes

$$E = (\sqrt{E_s}T)^2 .1/T$$

$$= E_s .$$

See Fig. 8.1

Now from eq. (21), the impulse response of a PRS system with unity gain low-pass minimum Nyquist bandwidth filter is

$$h(t) = \sum_{i=0}^{L} f_i \text{sinc}(2Wt - i).$$

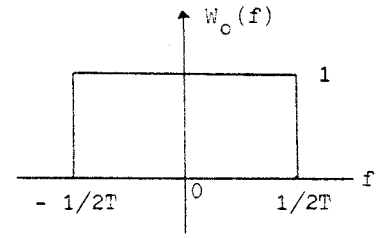
By scaling the gain of the filter from unity to $\sqrt{E_s T}$, the impulse response of a PRS system becomes

$$h(t) = \sum_{i=0}^{L} \sqrt{E_s/T}\ f_i \text{sinc}(t/T - i). \qquad (141)$$

Then the output signal wave from a PRS system with binary inputs x_k can be written as

$$s(t) = \sum_{k} x_k h(t - kT). \qquad (142)$$

Now we want to evaluate the average power P_s of the above signal wave s(t).

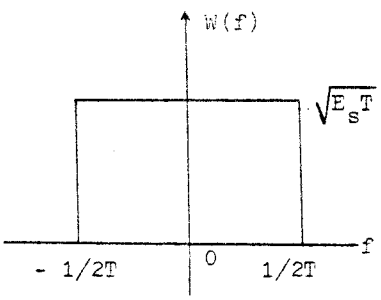


$$w_o(t) = \mathcal{F}\left[W_o(f)\right]$$

$$= 1/T \cdot \mathrm{sinc}(t/T)$$

$$\text{Energy} = 2 \cdot (1/2T) \cdot 1$$

$$= 1/T$$



$$w(t) = \mathcal{F}\left[W(f)\right]$$

$$= \sqrt{E_s/T} \cdot \mathrm{sinc}(t/T)$$

$$\text{Energy} = 2 \cdot (1/2T) \cdot E_s T$$

$$= E_s$$

Fig. 8.1 Energy of the impulse responses of low-pass filters with different amplitude gains

$$P_s = E\left[\lim_{N\to\infty}\frac{1}{(2N+1)T}\int_{-NT}^{NT} {}^{2}(t)dt\right]$$

$$= E\left\{\lim_{N\to\infty}\frac{1}{(2N+1)T}\int_{-NT}^{NT}\left[\sum_{k=-N}^{N} x_k h(t-kT)\right]^2 dt\right\}$$

$$= \lim_{N\to\infty}\frac{1}{(2N+1)T}\sum_{j=-N}^{N}\sum_{k=-N}^{N} E[x_j x_k]\int_{-NT}^{NT} h(t-jT)h(t-kT)dt$$

$$= \lim_{N\to\infty}\frac{E[x_k^2]}{(2N+1)T}\sum_{k=-N}^{N}\int_{-NT}^{NT} h^2(t-kT)dt +$$

$$= \lim_{N\to\infty}\frac{E[x_j x_k]}{(2N+1)T}\sum_{j=-N}^{N}\sum_{k=-N}^{N}\int_{-NT}^{NT} h(t-jT)h(t-kT)dt\Big|_{j\neq k}$$

(143)

The first term of eq. (143) can be written as

$$\frac{E[x_k^2]}{(2N+1)T}\sum_{k=-N}^{N}\int_{-\infty}^{+\infty}\frac{E_s}{T}\left[\sum_{i=0}^{L} f_i\,\mathrm{sinc}(t/T-(i+k))\right]^2 dt$$

$$= \frac{E[x_k^2]}{(2N+1)T}\cdot(2N+1)\cdot\frac{E_s}{T}\cdot\sum_{i=0}^{L} f_i^2 T$$

$$= E[x_k^2]\,\frac{E_s}{T}\sum_{i=0}^{L} f_i^2. \qquad (143)$$

as $\int_{-\infty}^{+\infty}\mathrm{sinc}(t/T-i)\mathrm{sinc}(t/T-j) = 0 \quad \text{for } i\neq j$

and $\int_{-\infty}^{+\infty}\mathrm{sinc}^2(t/T)dt = T.$

Let us evaluate the integral of the second term of eq. (143).

$$\int_{-\infty}^{+\infty} h(t - jT)h(t - kT)dt \Big|_{j\neq k}$$

$$= \int_{-\infty}^{+\infty} [\sum_{i=0}^{L} \sqrt{\frac{E_s}{T}} f_i \text{sinc}(t/T - (i + j))].$$

$$[\sum_{i=0}^{L} \sqrt{\frac{E_s}{T}} f_i \text{sinc}(t/T - (i + j))]dt \Big|_{j\neq k}$$

$$= \int_{-\infty}^{+\infty} \frac{E_s}{T} [\sum_{i=0}^{L} f_i \text{sinc}(t/T - (i + j))].$$

$$[\sum_{i=0}^{L} f_i \text{sinc}(t/T - (i + k))]dt \Big|_{j\neq k}$$

$$\neq 0$$

due to the orthonormal nature of the sinc function. The second expression can be proved to be zero by observing that $E[x_j x_k] = 0$ for $j \neq k$ as the input digits x_j and x_k are uncorrelated.

For polar inputs, $E[x_k] = 0$ and thus $E[x_k^2] = \sigma_x^2$. The average power becomes

$$P_s = \sigma_x^2 \sum_{i=0}^{L} f_i^2 (E_s/T)$$

$$= \sigma_x^2 R(0)(E_s/T) \qquad \text{from eq. (87)}$$

$$= \sigma_y^2 E_s/T \qquad (144)$$

from eq. (88). This is the average power seen by the channel. As discussed in Chapter 5, the output variance σ_y^2 can be varied by scaling the tap-gains $\{f_i\}$. Therefore we can normalize $\sigma_y^2 = 1$. In this case the average power provided to the channel is

$$P_s = E_s/T \tag{145}$$

Referring again to Section 5.1, we understand that the average power as seen by the receiver is less than that specified above due to the correlation among samples of PRS systems. In order to take the degradation effect of PRS systems into account, a factor $R(0)/d_{free}^2$ of eq.(90) should be included, obtaining

$$P_s = R(0)/d_{free}^2 \cdot E_s/T = 4/d_{norm}^2 \cdot E_s/T \tag{146}$$

The above equation is intuitively satisfying: whenever d_{norm}^2 of a PRS system is less than the d_{norm}^2 of binary polar PAM, which is equal to 4, then the energy per symbol should be scaled up by a factor of $4/d_{norm}^2$. By the scaling up of symbol energy, the average power seen by the receiver for sequence estimation will stay the same at E_s/T. If the d_{norm}^2 of a PRS system is equal to 4, the scaling factor becomes one and automatically no scaling occurs.

Referring to Chapter 7, α denotes the 99% energy bandwidth of PRS systems. So the bandwidth utilized by the PRS systems for this study is $\alpha W = \alpha/2T$. In section 2 of this Chapter, we propose the usage of a penalty function to jointly optimize both power and bandwidth simultaneously. The Shannon capacity equation seems a reasonable candidate, although it only

describes the power and bandwidth trade-offs for ideal sytems, which is not the case for PRS systems. Substituting eq. (146) into eq. (140) and using the 99% energy bandwidth $\alpha/2T$, eq. (140) becomes

$$C = \frac{\alpha}{2T} \log_2\left(1 + \frac{8E_s}{\alpha N_o d^2_{norm}}\right) \tag{147}$$

Furthermore, by setting $E_s/N_o = 10$, we guarantee the probability of bit error for binary signalling is about 10^{-5}. Finally, by normalizing the bandwidth $1/2T$ to 100 for calculation purposes, we obtain the penalty function

$$C = 100\,\alpha \log_2\left(1 + \frac{80}{\alpha d^2_{norm}}\right) \tag{148}$$

8.4 Interpretation of the Cost Function

For a given α and d^2_{norm}, the function C in eq. (148) really gives a measure of the channel capacity of an ideal system. We instead use it as a gauge for penalty or cost. The higher the value of C, the higher the cost or penalty paid. From this viewpoint, we want the value of C to be as low as possible but this also implies that we want the lowest channel capacity which contradicts to our usual intuition that the higher the channel capacity of a system, the better it is.

This paradox can be explained as follows. In our model, the maximum single-sided bandwidth of any PRS system is $\frac{1}{2T}$ Hz, which for

binary signalling, only requires a channel capacity of $\frac{1}{T}$ b/s. Because we have normalized $\frac{1}{2T}$ to be 100 Hz, therefore the channel capacity required for binary signalling is only 200 b/s. Any channel capacity above 200 b/s will be wasted as channel capacity is an entity that is not "accumulative" as discussed in Section 8.2. Thus we want the lowest channel capacity as long as it is above 200 b/s. Hence, we have a coherent objective of minimizing C in eq. (148) no matter how we interpret its meaning: either as a penalty function or as a channel capacity function. Another point to note is that it is the relative cost of a code compared with others that has significance, rather than the absolute cost of the code; the main objective of proposing the cost function is to device a gauge for comparisons among all codes in terms of the joint effects of bandwidth and degradation.

The 99% energy bandwidth α shows up in two positions within the function of eq. (148). In the first position, C is directly proportional to α. As α decreases, the penalty decreases proportionally. For this position of α , we want to choose a PRS code that minimizes α, that is, we want a code with energy concentrated in the low frequency portion of the spectrum. In the second position, α is in the denominator of the $\log_2(.)$ function. In this case, α has the reverse effect on the C function. The effect of the α in the second position counteracts the effect of α in the first position in such a way that the net effect on the C function is reduced, but C still increases with α.

The effect of d^2_{norm} on the cost function is similar to the α in the second position. By re-arranging eq. (148) we have

$$C = 100\,\alpha \log_2 \left(1 + \frac{1}{\alpha\left(\frac{d^2_{norm}}{80}\right)}\right). \qquad (149)$$

Thus both α and $\frac{d^2_{norm}}{80}$ have the same effect: as α or $\frac{d^2_{norm}}{80}$ increases, the C function decreases and vice versa. The difference is that their ranges are different: $0 < \alpha < 1$ but $0 < d^2_{norm} < 4$.

8.5 The Cost Contours

By using the above cost function of eq. (148), we compute the cost in a grid pattern fashion on the (f_2, f_1) plane using the appropriate 99% energy bandwidth α and the normalized free distance d^2_{norm} for each code. Observe that the shapes of the cost contours for a particular f_3 depend on the shapes of their corresponding bandwidth and degradation contours. From the C function of eq. (148), we see that the 99% energy bandwidth α always exerts the dominant influence on the cost for small $\alpha(\alpha < 90\%)$. For α approaching 1, the C function of eq. (148) can be rewritten as

$$C = 100 \log_2 \left(1 + \frac{80}{d^2_{norm}}\right). \qquad (150)$$

Hence, the dominant influence on the cost function now switches to d^2_{norm} instead. The cost contours express the above ideas in the pictorial form.

The cost contours for f_3 = -1.6, -0.8, 0.0, 0.8, 1.6 and 3.1 are shown in Fig. 8.2 to Fig. 8.7. The cost contours in the figures are separated from each other by a value of 25.

1) For $f_3 = -1.6$, the cost contours across the 1st, 2nd and 3rd quadrants have similar shapes as the corresponding bandwidth contours of Fig. 7.2. Over these quadrants, we have a minimum bandwidth of about 73% but only a maximum degradation of 0.5 dB. See Fig. 7.2 and Fig. 5.2. Thus the dominant influence on the cost in this area is the bandwidth α. The minimum cost contour of 375 occurs and coincides with the minimum bandwidth contour of 80%. In the 4th quadrant, the bandwidth is larger than 98% and the maximum degradation is 3.5 dB, thus the dominant influence on the cost switches to the d^2_{norm} instead. Comparison of Fig. 8.2 with Fig. 5.2 in the 4th quadrant shows their similar elliptical structures. We see that as d^2_{norm} decreases, that is, as degradation increases, the cost increases accordingly. This is intuitively convincing: as the free distance of a code gets shorter, more power is needed to distinguish one codeword from another, thus costing more.

2) For $f_3 = -0.8$, the same observations as $f_3 = -1.6$ apply. Note that the minimum cost contour of 375 is halfway between the 1st and 2nd quadrants and thus has moved towards the 1st quadrant compared with the same contour for $f_3 = -1.6$. This shows the dominant effect of the bandwidth α on the cost contours: the cost contours migrant along the same path through the (f_2, f_1) plane as the bandwidth contours for increasing f_3's. Refer to Section 7.2.

3) For $f_3 = 0.0$, both the bandwidth contours and the degradation contours reside in the 1st quadrant. For narrow

bandwidth (α < 90%), has the dominant effect on the cost over the d^2_{norm} although the overall shapes of the cost contours skew towards the region of lower degradation. The minimum cost contour of 375 resides entirely within the minimum bandwidth contour of 70% and just outside the maximum degradation contour of 2.0 dB. See Fig. 8.4, Fig. 7.4, and Fig. 5.1.

4) For f_3 = 0.8, the region with bandwidth contours of 60%, 70% and 80% overlaps the region with degradation contours of 4 dB to 2 dB. Refer to Fig. 8.5, Fig. 7.5 and Fig. 5.5. The minimum cost contours of 350, 375, 400 and 425 follow the same concentric elliptical shapes as the 60%, 70% and 80% bandwidth contours. This is a manifestation of the dominant effect of α over d^2_{norm} on the cost function within this region. Also these cost contours are very close to each other showing that a slight variation of position on the (f_2, f_1) plane may result in a large change in the cost. Also note that these cost contours cut across the degradation contours in such a way the effect of increasing bandwidth α is counteracted by the decrease in degradation or vice versa with the net effect that the cost stays the same. Just outside the 80% bandwidth contour, d^2_{norm} begins to exert more effect on the cost giving a region of high cost of about 450. In the 3rd quadrant, the cost contour of 425 is between the 90% and 98% bandwidth contours and follows about the same shape as the bandwidth contours.

5) For f_3 = 1.6, there are two portions to the cost contours in the 1st quadrant. One portion consists of cost contours with values 375, 400 and 425. The other portion consists of cost contours with values 450 and 475. For the former portion, the cost is dominated by the effect of the 99% energy bandwidth α and thus the cost contours follow the same shapes as the bandwidth contours of 60%, 70% and 80%. For the latter portion, the cost is dominated more by the d^2_{norm} and the contours fit between the 90% and 98% bandwidth contours. In the 3rd quadrant, the cost contours of 400 is within the 90% bandwidth contour while the 425 cost contour is between the 90% and 98% bandwidth contours.

6) For f_3 = 3.1, the cost contours in the 1st quadrant also consist of two portions. One portion consists of cost contours of values 375, 400 and 425 and located within the 60% and 70% bandwidth contours. The costs for this portion is dominated by α. The other portion is located between the 80% and 98% bandwidth contours. It consists of cost contours of value 450, 475 and 500, being dominated more by the d^2_{norm}. There is a cost contour of 425 that spans the 1st, 3rd and 4th quadrant. It is dominated by the effect of bandwidth α.

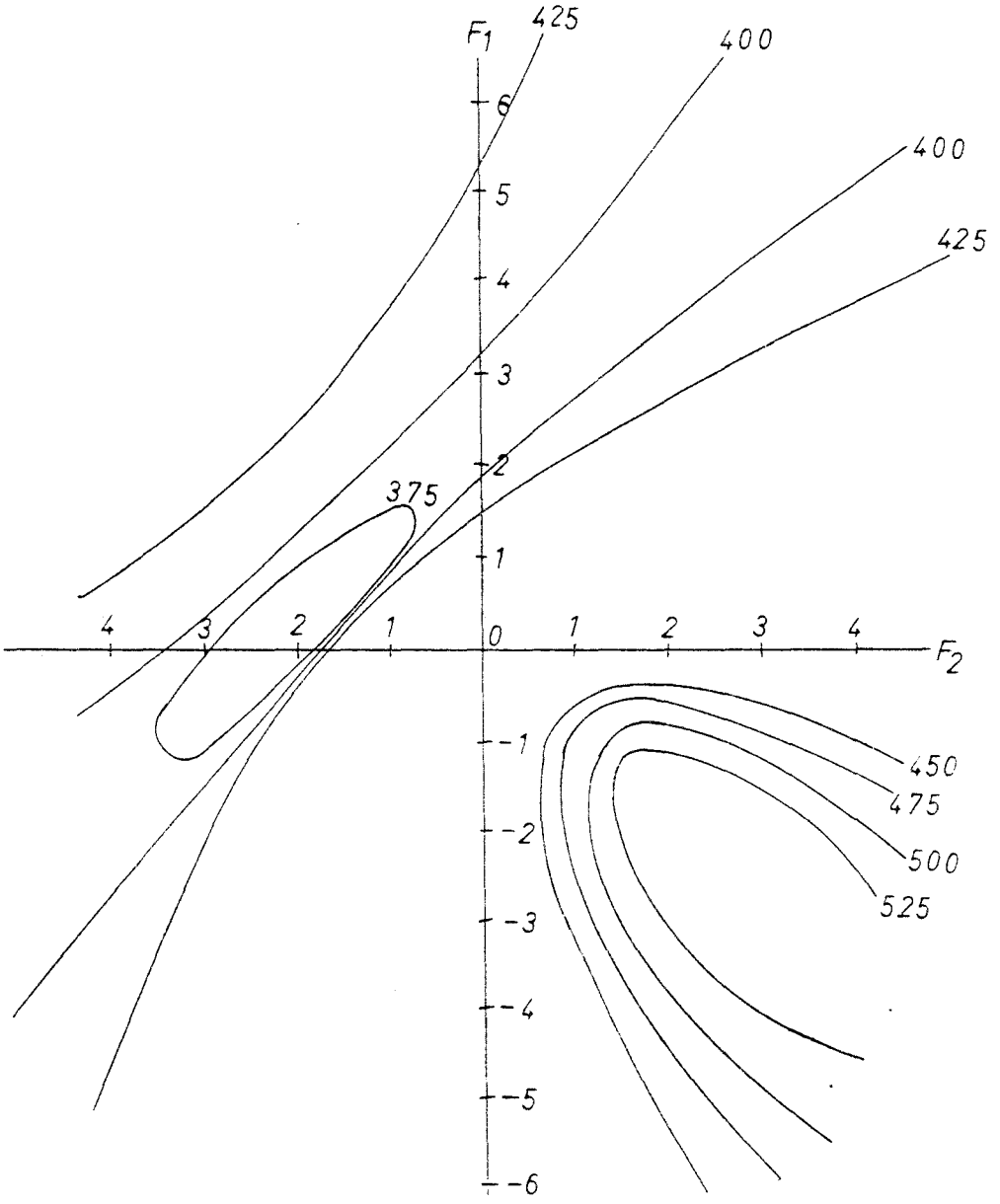


Fig. 8.2 Contours of constant channel cost

$F_0 = 1$, $F_3 = -1.6$.

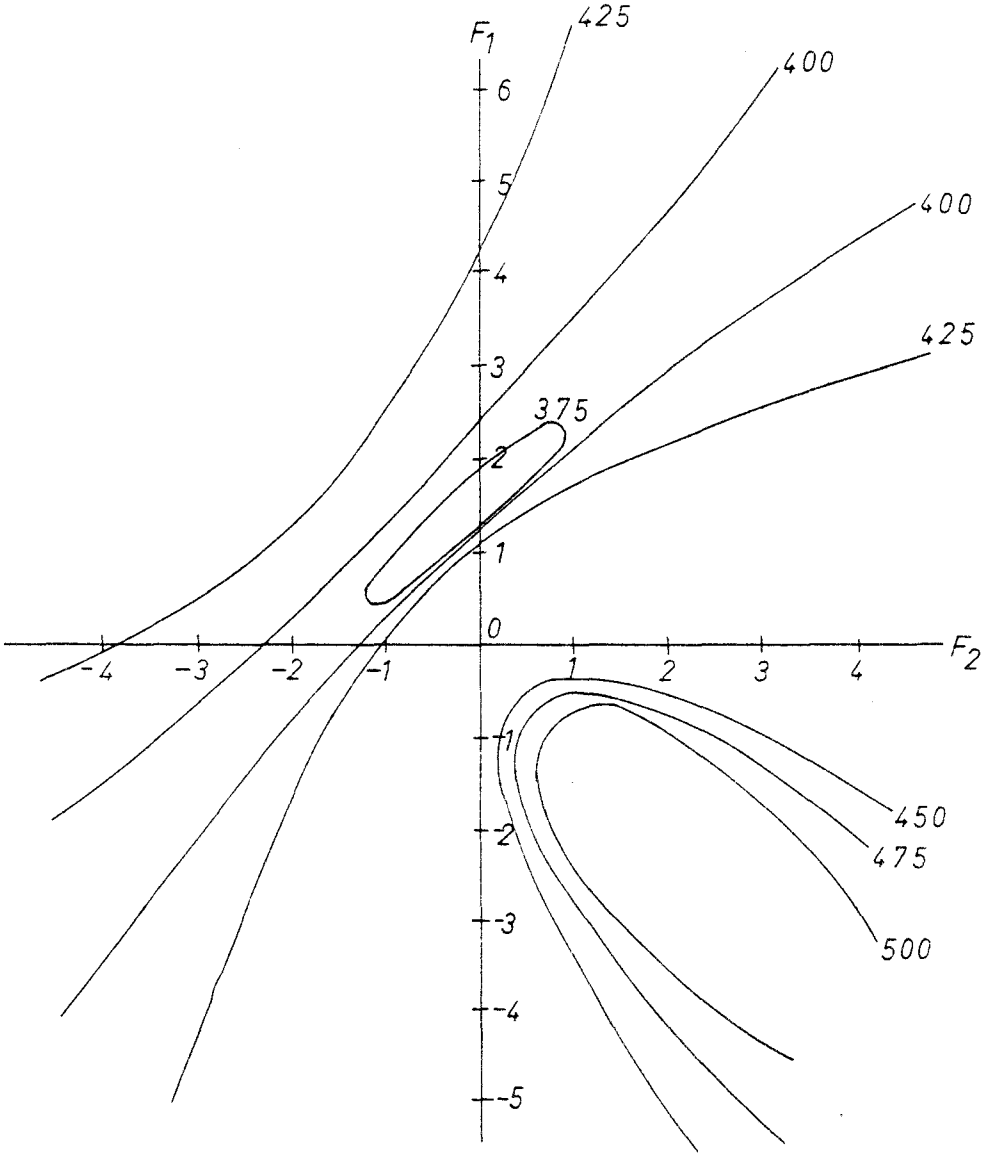


Fig. 8.3 Contours of constant channel cost

$F_0 = 1$, $F_3 = -0.8$.

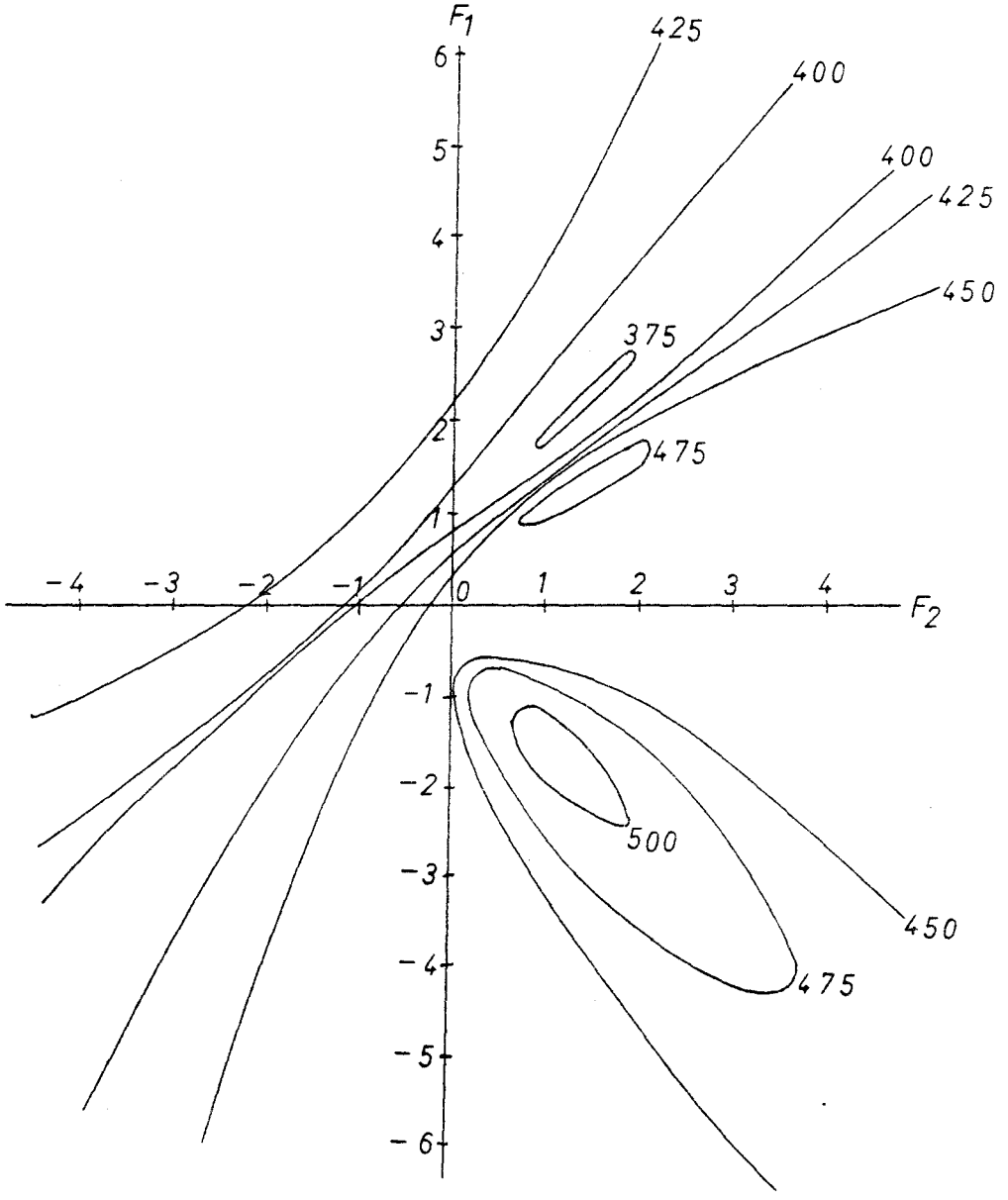


Fig. 8.4 Contours of constant channel cost

$F_0 = 1$, $F_3 = 0.0$.

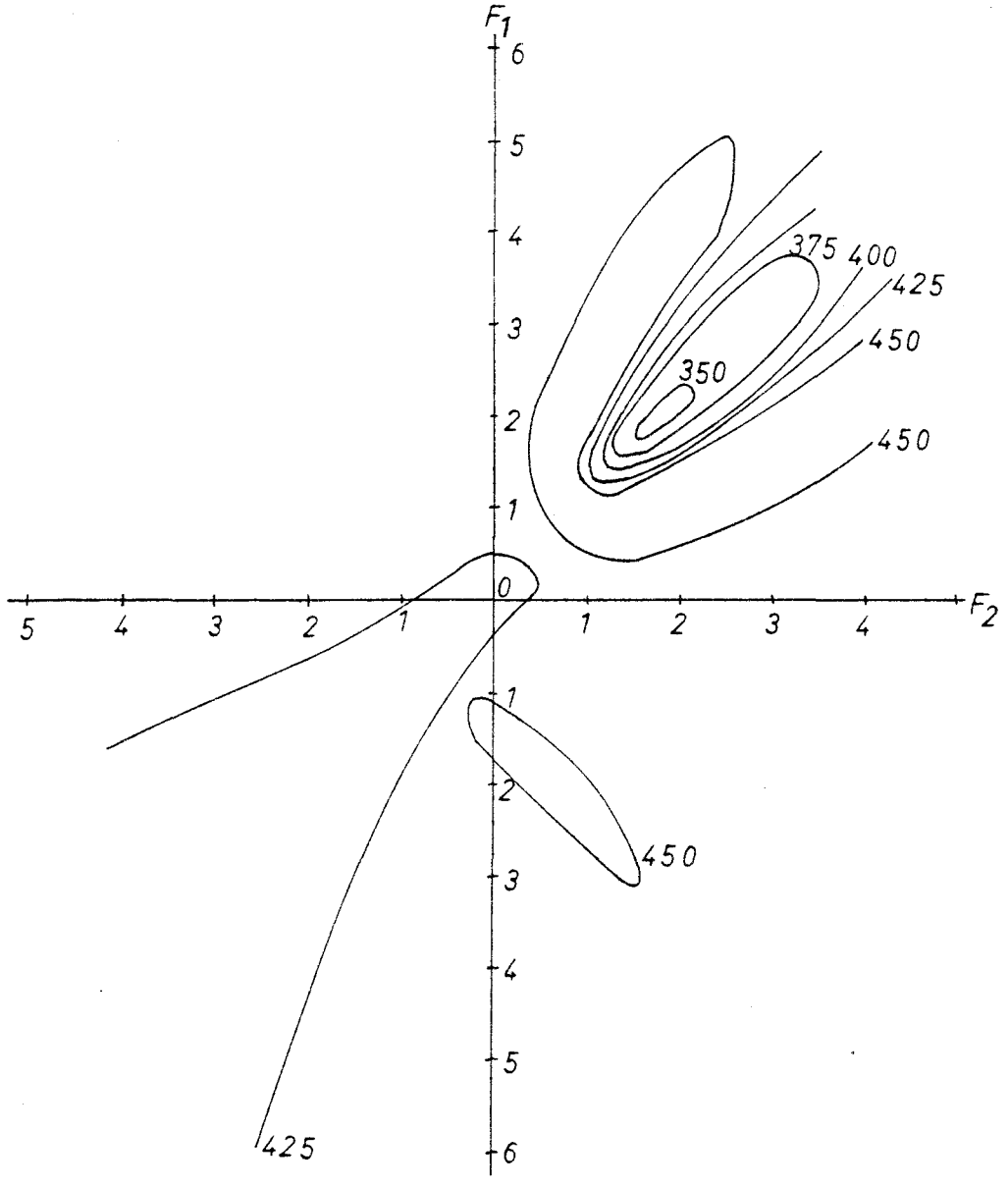


Fig. 8.5 Contours of constant channel cost

$F_0 = 1$, $F_3 = 0.8$.

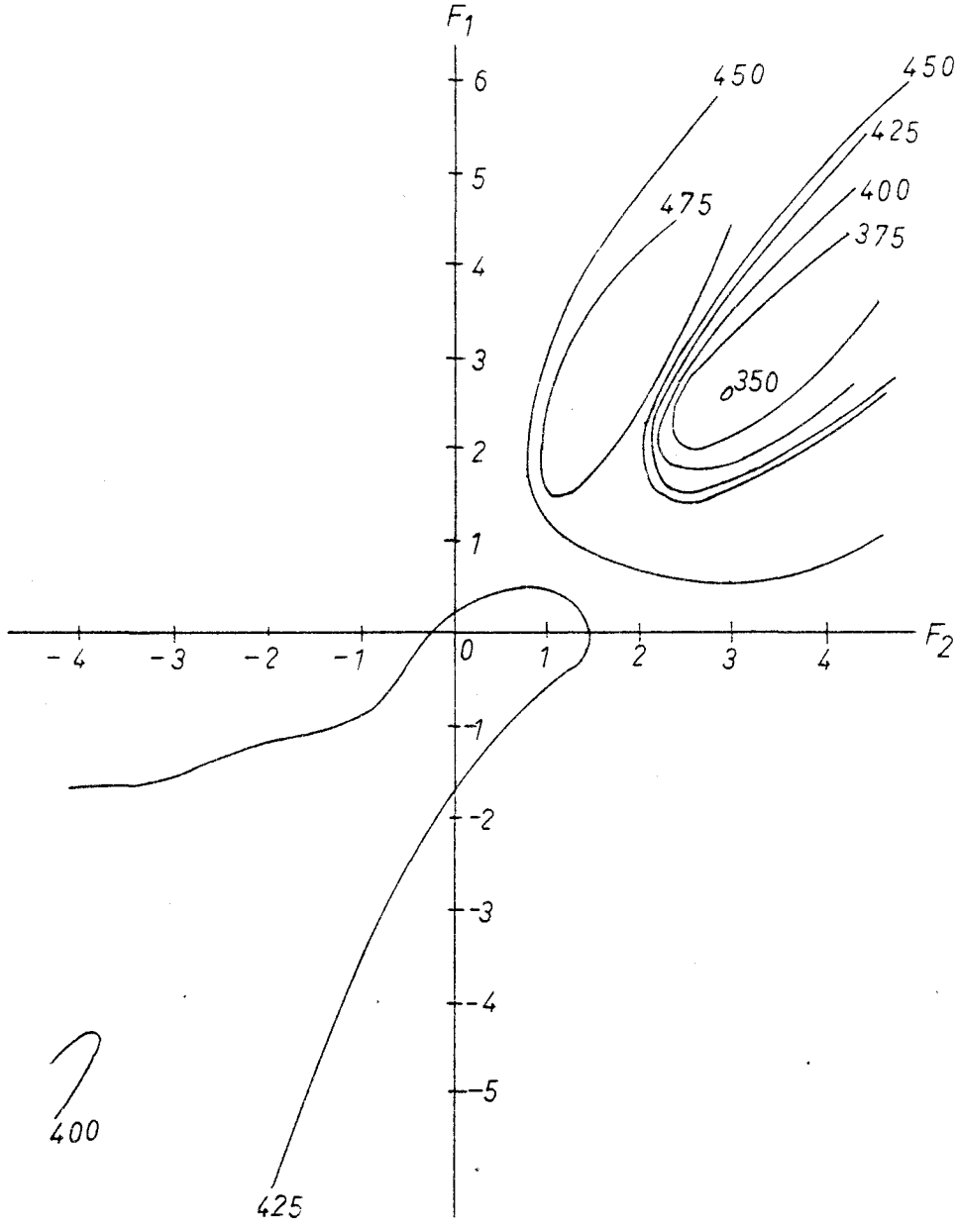


Fig. 8.6 Contours of constant channel cost
$F_0 = 1$, $F_3 = 1.6$.

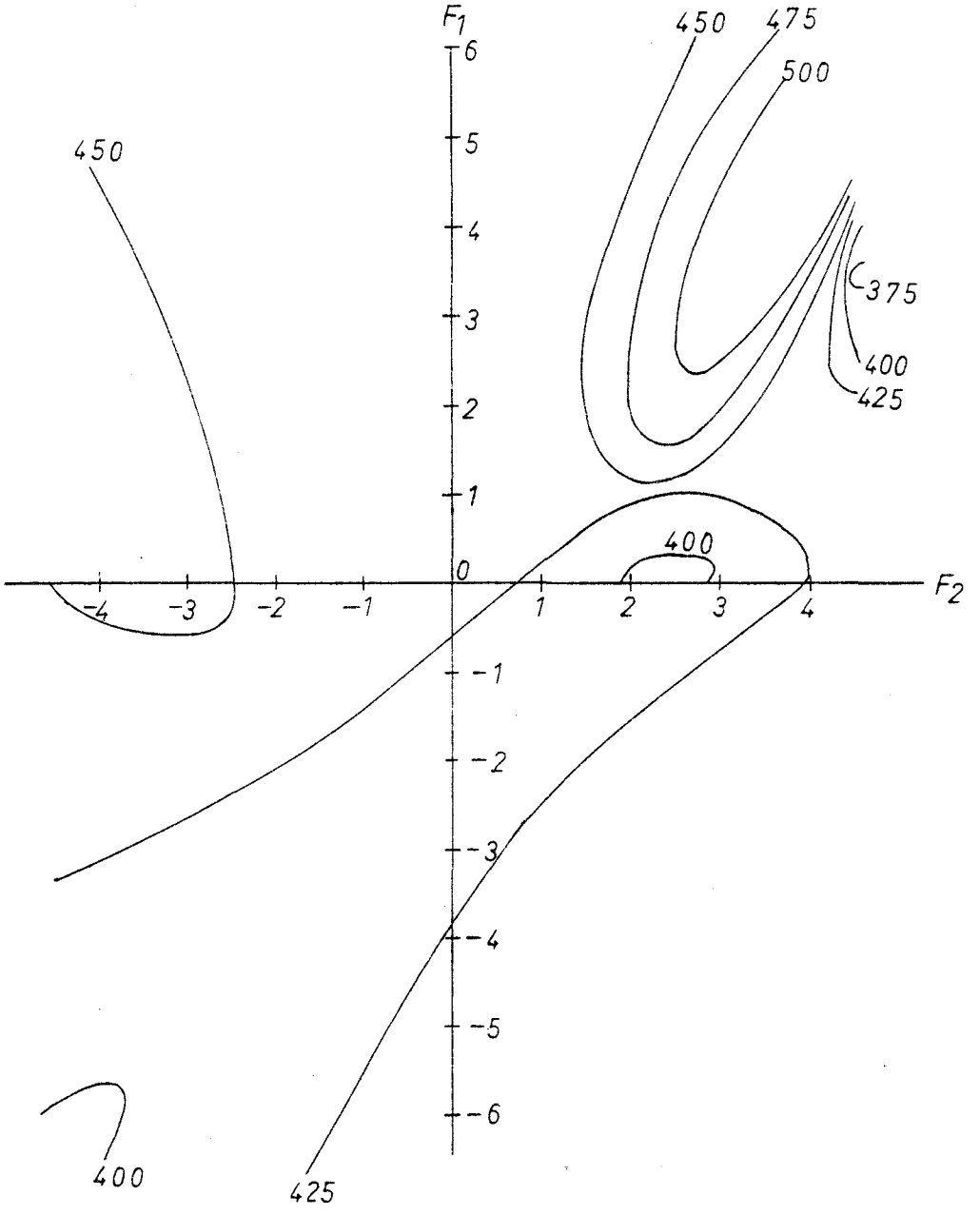


Fig. 8.7 Contours of constant channel cost

$F_0 = 1$, $F_3 = 3.1$.

CHAPTER 9

CONCLUSION

9.1 Summary of Findings and Discoveries

This thesis develops the "double" dynamic programming for the maximum likelihood sequence estimation over channels having finite duration impulse response. The complexity involved in this algorithm is the square of that of the one dimensional Viterbi algorithm.

The four performance measures, namely, degradation, decision, depth, 99% energy bandwidth and channel cost are extensively evaluated for channels with length up to four signalling intervals. We present the results of each parameter for different channels in contour form. A contour shows the locus of channels with a given functional value. Contours with different values are shown in each contour diagram.

Contours with degradation are elliptical and lie within the parabolic contour which demarcates the region of no degradation with the one which has. For channels with length $K = 3$, degradation occurs at a state merge of trellis depth $K + 1$ only. For channels with length $K = 4$, degradation occurs at both depth of $K + 1$ and $K + 2$ with the deeper degradation occurring at depth $K + 1$. It is observed that as a contour crosses from one region to another with a different merging depth for the occurrence of free distance, the equation of the contour changes correspondingly, resulting in a jump or point of inflection.

Two channels are dual when one channel has the same impulse response over a number of signalling intervals as another in terms of magnitude but with the appropriate opposite signs. This duality manifests itself in the degradation contours of the channels, as mirror reflection along the appropriate axis of one channel with another.

Catastrophic error propagation occurs when the decision depth required for maximum likelihood sequence estimation over a channel approaches infinity. It results if the Euclidean distance between any two non-merging signal sequences does not increase with trellis depth in the state trellis of the channel. The decoding algorithm is tricked into an un-ending search for the free distance of the channel. Practically, we truncate the decision depth for manageable computation.

Contours with catastrophic error propagation are mostly governed by linear equations making an angle of 45^{o} and 135^{o} with the f_2-axis of the $(f_2 - f_1)$ plane. Thus they are parallel to the major axis of the elliptical degradation contours. The equations are given by equating the output levels of the pair of periodic state sequences whose Euclidean distance between them does not increase with depth. There exists also some isolated channels with catastrophic error propagation whose loci do not conform to the linear equations. They are caused by some particular sequence pairs. Duality also applies to the decision depth contours as in degradation contours because both measures are caused by error events.

Minimum 99% energy bandwidth contours are found to reside in the first quadrant because the zeroes of the z-transform of all positive impulse response channels are in the left band z-plane, thus pulling down

the amplitude of the high frequency response of the spectra. This results in the concentration of energy in the low frequency portion of the spectra.

It is also found that the minimum 99% energy bandwidth for channels with length three decreases from 63.6% to 52.7% for channels with length four. This is because as the length of a channel increases, the number of zeroes on the left half z-plane also increases, thus pulling down further the amplitude of the high frequency portion! We expect the minimum 99% energy bandwidth to further decrease with increasing channel length.

The proposed channel cost is similar to the Shannon capacity equation and is used to jointly optimize the energy and the bandwidth required for adequate signalling over channels with finite impulse response. According to the equation, the 99% energy bandwidth exerts two counteracting effects on the channel cost, while energy degradation only has noticeable influence on the cost in regions where the 99% energy bandwidth is greater than 90%. Overall, 99% energy bandwidth has dominant influence on the total cost compared with degradation.

From the channel cost contour figures, it can be seen that the contours follow closely the bandwidth contours in those regions where the 99% energy bandwidth is less than 90%. In regions where the 99% energy bandwidth is greater than 90%, degradation takes over to be the dominant influence, causing the channel cost contours to follow the elliptical shapes of the degradation contours.

By referring to the contours of the four measures, one can choose the appropriate channels to minimize the bandwidth required, or the degradation suffered, or the

complexity involved in maximum likelihood sequence estimation. Finally, one can use the channel cost to provide a guide for selecting the channel which minimizes both bandwidth required and degradation suffered.

For channel with lengths up to four, it is found that the regions of narrow 99% energy bandwidth lie on the regions where long decision depths and catastrophic error propagation occur. In fact, for most cases, the linear equations of the catastrophic error propagation contours cut across the regions of minimum 99% energy bandwidth.

The same observation applies to the channel cost contours and the decision depth contours, that is, the regions of lowest channel cost lie on the regions where catastrophic error propagation occur. This is expected since the channel cost is dominated by the 99% energy bandwidth of 90% or less. Thus, in order to have low channel cost, complexity in terms of long decision depth is the price to pay and vice versa. The channel cost can be regarded as a communication cost while complexity shows up in processing cost. Hence, it amounts to the trade-off between communication cost and processing cost.

The following table shows the range of values within which the maximum or minimum of the four performance measures lie when maximum likelihood sequence estimation is used.

Performance Measures	Channel Length(K)			
	1	2	3	4
D Maximum Degradation in dB	0.	0.	2. < D < 2.5	4. < D < 4.5
BW Minimum 99% Energy Bandwidth in percent	99.	80. < BW < 90.	60. < BW < 70.	50. < BW < 60.
DP Maximum Decision Depth	0	infinity	infinity	infinity
C Minimum Channel Cost	436.2	375. < C < 400.	350. < C < 375.	325. < C < 350.

Note that the maximum degradation suffered increases as channel length increases, while both minimum bandwidth and channel cost decrease with increasing channel length. Thus, venture into the study of channels with length longer than four signalling intervals is strongly recommended, since this may result in the discovery of lower cost channels.

9.2 Suggestions for Further Work

Extensions to this work are numerous. This work assumes the inputs to the partial response signalling system to be impulses of zero duration. Finite duration pulses as inputs are more realistic. In addition,

this study assumes the transmitting filter to be an ideal low-pass filter which is not physically realizable. Non-minimum Nyquist filters like the raised-cosine filter should be considered.

The thesis only considers the binary signalling case. Using an m-ary inputs will decrease the required bandwidth by a factor of $\log_2 m$, although at the expense of lowering the signal to noise ratio for detection for a given average transmitted power. Investigation into the changes and relationships among complexity, degradation and bandwidth requirements for m-ary inputs may yield fruitful results. Some work has been done by the author although the results are not reported here.

There are two interesting theoretical problems embedded in this study which we do not directly tackle. One is the finding of the regions where the minimum 99% energy bandwidth lie, the other involves the joint minimization of both the 99% energy bandwidth and degradation. Both tasks require a non-linear variational calculus approach. Theoretical solutions compared with our simulations may result in deeper understanding.

Due to the large amount of computations required for running the "double" dynamic program, only channels with length up to four are extensively investigated. In reality, channel lengths of ten to twenty may need to be considered, for example, in interference due to multipath echoes. In addition, channels with frequency distortion and non-linearity should be taken into account. These must all await a more sophisticated computational approach.

APPENDIX A

DERIVATION OF THE INPUT AND OUTPUT VARIANCE OF A FILTER

A.1 Variance of M-ary Input

For polar impulses of heights $\pm a/2$, $\pm 3a/2, \ldots \pm(m-1)a/2$ and assuming all different levels are equally likely, the variance is

$$\sigma_x^2 = (2/m)\{(a/2)^2 + (3a/2)^2 + \ldots + ((m-1)a/2)^2\} - m_x,$$

where m_x is the mean value of the input.

As the input is polar, $m_x = 0$; consequently,

$$\sigma_x^2 = (m^2 - 1)a^2/2.$$

This derivation does not actually depend on the polar assumption, but the zero mean value result does.

A.2 Output Variance of a Time-Invariant Filter

The input sequence x(n) is a wide-sense stationary discrete-time random process with autocorrelation $\sigma_x^2\delta(m)$; that is, we assume the input variables to be i.i.d with variance σ_x^2.

Assume that the mean input value is zero so that the mean output is zero also. Let the unit-sample response of the time invariant filter be h(n), then the output sequence is

$$y(n) = \sum_{k=-\infty}^{\infty} h(n - k)x(k) = \sum_{k=-\infty}^{\infty} h(k)x(n - k).$$

The autocorrelation function of the output process is [24]

$$\begin{aligned}\phi_{yy}(n,n + m) &= E[y(n)y(n + m)] \\ &= E[\sum_{k=-\infty}^{\infty} \sum_{r=-\infty}^{\infty} h(k)h(r)x(n - k)x(n + m - r)]. \\ &= \sum_{k=-\infty}^{\infty} h(k) \sum_{r=-\infty}^{\infty} h(r)E[x(n - k)x(n + m - r)].\end{aligned}$$

As $x(n)$ is stationary, $E[x(n - k)x(n + m - r)]$ depends only on the time different $m + k - r$. Hence,

$$\phi_{yy}(n,n + m) = \sum_{k=-\infty}^{\infty} h(k) \sum_{r=-\infty}^{\infty} h(r)\, \phi_{xx}(m + k - r) = \phi_{yy}(m).$$

This means that the output autocorrelation sequence depends only upon the time difference m also.

Substituting $q = r - k$, the above equation becomes

$$\begin{aligned}\phi_{yy}(m) &= \sum_{\ell=-\infty}^{\infty} \phi_{xx}(m - q) \sum_{k=-\infty}^{\infty} h(k)h(q + k) \\ &= \sum_{\ell=-\infty}^{\infty} \phi_{xx}(m - q)R(q).\end{aligned}$$

where $R(q) = \sum_{k=-\infty}^{\infty} h(k)h(q + k).$

For $q = 0$, $R(0) = \sum_{k=-\infty}^{\infty} h^2(k)$ = energy of the linear time-invariant filter.

Substituting

$$\phi_{xx}(m) = \sigma_x^2 \delta(m) \text{ , we get}$$

$$\phi_{yy}(m) = \sigma_x^2 \sum_{\ell=-\infty}^{\infty} \delta(m - q)R(q)$$

$$= \sigma_x^2 R(m).$$

Setting m = 0 leads to

$$\sigma_y^2 = \sigma_x^2 R(0),$$

as $\phi_{yy}(0) = \sigma_y^2 =$ output sequence.

This derivation depends critically on the assumption that the inputs are i.i.d. variables with variance σ_x^2.

APPENDIX B

DERIVATION OF THE ENERGY DENSITY AND THE TOTAL ENERGY OF PRS SYSTEMS

B.1 The Energy Density of PRS Systems

For $K = 3$, $F(D) = 1 + f_1 D + f_2 D^2$

The energy density of the above filter is

$$|F(D)|^2_{D=e^{j\omega}}$$

$$= |1 + f_1 e^{j\omega} + f_2 e^{j2\omega}|^2$$

$$= (1 + f_1 e^{j\omega} + f_2 e^{j2\omega})\ (1 + f_1 e^{-j\omega} + f_2 e^{-j2\omega})$$

$$= 1 + f_1 e^{-j\omega} + f_2 e^{-j2\omega} + f_1 e^{j\omega} + f_1^2$$

$$+ f_1 f_2 e^{-j\omega} + f_2 e^{j2\omega} + f_1 f_2 e^{j\omega} + f_2^2$$

$$= 1 + f_1^2 + f_2^2 + f_1(e^{j\omega} + e^{-j\omega})$$

$$+ f_2(e^{j2\omega} + e^{-j2\omega}) + f_1 f_2(e^{j\omega} + e^{-j\omega})$$

$$= 1 + f_1^2 + f_2^2 + 2f_1 \cos\omega + 2f_1 f_2 \cos\omega + 2f_1 f_2 \cos 2\omega$$

as $e^{-j\beta} + e^{j\beta} = 2\cos\beta$

It is straightforward to extend the above derivation to the general case for $K > 3$.

Thus

$$|H(\omega)|^2 = (1 + f_1 e^{j\omega} + f_2 e^{j2\omega} + \ldots\ldots + f_L e^{jL\omega}) \cdot$$

$$(1 + f_1 e^{-j\omega} + f_2 e^{-j2\omega} + \ldots\ldots + f_L e^{-jL\omega})$$

$$= 1 + f_1 e^{-j\omega} + \ldots\ldots f_L e^{-jL\omega} +$$

$$f_1^2 + f_1 e^{j\omega} + f_1 f_2 e^{-j\omega} + \ldots + f_1 f_k e^{-j(k-1)L} + f_1 f_L e^{-j(L-1)\omega} +$$

$$f_2^2 + f_2 e^{j2\omega} + f_2 f_1 e^{j\omega} + f_2 f_k e^{-j(k-2)L} + f_2 f_L e^{-j(L-2)\omega} +$$

$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

$$f_L^2 + f_L e^{jL\omega} + f_L f_1 e^{j(L-1)\omega} + f_L f_k e^{-j(k-L)\omega} + f_L f_{L-1} e^{j\omega} +$$

$$= 1 + f_1^2 + \ldots\ldots\ldots\ldots\ldots\ldots + f_L^2$$

$$+ f_1(e^{j\omega} + e^{-j\omega}) + \ldots\ldots + f_L(e^{jL\omega} + e^{-jL\omega})$$

$$+ f_1 f_2(e^{j\omega} + e^{-j\omega}) + \ldots\ldots + f_1 f_2(e^{j(L-1)\omega} + e^{-j(L-1)\omega})$$

$$+ \ldots\ldots\ldots\ldots$$

$$+ f_{k-1} f_k(e^{j\omega} + e^{-j\omega}) + \ldots\ldots + f_k f_L(e^{j(L-k)\omega} + e^{-j(L-k)\omega})$$

$$+ \ldots\ldots\ldots\ldots$$

$$+ f_L f_{L-1}(e^{j\omega} + e^{-j\omega}).$$

B.2 The Energy of The Impulse Response of a PRS System

In the time domain, the energy of a sequence [24]:

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n - k)$$

is

$$\sum_{k=-\infty}^{\infty} x_k^2$$

for real x(k).

For the impulse response

$$F(D) = 1 + f_1 D + ... + f_L D^L,$$

the sequence becomes

$$x(n) = \delta(t) + f_1\delta(t - T) + ...$$

$$+ f_L\delta(t - LT)$$

for delay units of T sec. The energy for the impulse response is thus

$$\sum_{i=0}^{L} f_i^2.$$

In the frequency domain, the energy of F(D) for K = 3 can be obtained by substituting $\alpha = \frac{1}{2}$ in eq. (138), i.e.,

$$\text{Energy} = 1 + f_1^2 + f_2^2 + \frac{2f_1}{\pi} \sin\pi + \frac{2f_1 f_2}{\pi} \sin\pi + \frac{2f_2}{\pi} \sin 2\pi$$

$$= 1 + f_1^2 + f_2^2 ,$$

as $\sin k\pi = 0$ for k = integer.

Generalizing the above derivation from B.1, for any $K > 3$, we have

$$\text{Energy of } F(D) = \frac{1}{\pi} \int_{-\pi}^{+\pi} |H(\omega)|^2 d$$

$$= 1 + f_1^2 + \ldots\ldots f_L^2$$

$$= \sum_{i=0}^{L} f_i^2$$

as all terms involving $(e^{jk\omega} + e^{-jk\omega})$ go to zero after integration of $|H(\omega)|^2$ from $-\pi$ to π in the frequency domain. Refer to B.1 for the expression of $|H(\omega)|^2$.

REFERENCES

[1] R. Bellman, Dynamic Programming, Princeton, N.J.: Princeton University Press 1957.

[2] S. V. H. Qureshi and E. E. Newhall, "An adaptive receiver for data transmission over time-dispersive channels," IEEE Trans. Inform. Theory, Vol. IT-19, pp. 448-457, July, 1973.

[3] G. D. Forney, Jr., "Lower bounds on error probability in the presence of large intersymbol interference," IEEE Trans. Commun. Technol. (Corresp.), vol. COM-20, pp. 76-77, Feb. 1972.

[4] G. D. Forney, Jr., "The Viterbi algorithm," IEEE Proceedings, vol. 61, pp. 268-278, March 1973.

[5] G. D. Forney, Jr., "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," IEEE Trans. Inform. Theory, vol. IT-18, pp. 363-378, May 1972.

[6] D. G. Messerschmitt, "A geometric theory of intersymbol interference, part 1: zero-forcing and decision-feedback equalisation," Bell Syst. Tech. J., vol. 52, pp. 1483-1518, Nov. 1973.

[7] D. G. Messerschmitt, "A geometric theory of intersymbol interference, part 2: performance of the maximum-likelihood detector," Bell Syst. Tech. J., vol. 52, pp. 1521-1539, Nov. 1973.

[8] P. Kabal and S. Pasupathy, "Partial response signalling," IEEE Trans. Commun., vol. COM 23, pp. 921-934, Sept. 1975.

[9] A. J. Viterbi, "Convolutional codes and their performances in communication systems," IEEE Trans. Commun. Techn., vol. COM-19, pp. 751-772, Oct. 1971.

[10] H. Kobayashi, "Correlative level coding and maximum-likelihood decoding," IEEE Trans. Inform. Theory, vol. IT-19, pp. 586-594, Sept. 1971.

[11] J. K. Omura, "On Viterbi decoding algorithm," IEEE Trans. Inform. Theory, vo. IT-15, pp. 177-179, Jan. 1969.

[12] R. W. Lucky, J. Salz and E. J. Weldon, Jr., Principles of Data Communication, New York: McGraw-Hill, 1968.

[13] J. M. Wozencraft and I. M. Jacobs, Principles of Communication Engineering, New York: Wiley, 1965, Ch. 4 & 5.

[14] F. R. Magee, Jr., and J. G. Proakis, "Adaptive maximum-likelihood sequence estimation for digital signalling in the presence of intersymbol interference," IEEE Trans. Inform. Theory (Corresp), vol. IT-19, pp. 120-124, Jan. 1973.

[15] F. R. Magee, Jr., and J. G. Proakis, "An estimate of the upper bound on error probability for maximum-likelihood sequence estimation on channels having a finite-duration pulse response," IEEE Trans. Inform. Theory, vol. IT-10, pp. 669-702, Sept. 1973.

[16] E. Bedrosian, "Spectrum-conservation by efficient channel utilization," IEEE Commun. Society Magazine, pp. 20-27, March, 1977.

[17] C. E. Shannon, "Communication in the presence of noise," Proc. of IRE, vol. 37, pp. 10-21, Jan. 1949.

[18] G. Ungerboeck, "Adaptive maximum likelihood receiver for carrier-modulated data-transmission systems," IEEE Trans. on Commun., vol. COM-22, pp. 624-636, May 1974.

[19] Sedreyfus and Law, The Art and Theory of Dynamic Programming New York: Academic Press, 1977, Ch. 2.

[20] W. R. Bennett, Introduction to Signal Transmission New York: McGraw Hill, 1970, Ch. 3.

[21] H. Kobayashi and D. T. Tang, "On decoding of correlative level coding systems and ambiguity zone detection," IEEE Trans. Commun. Techn., vol. COM-19, pp. 467-477, Aug. 1971.

[22] R. R. Anderson and G. J. Foschini, "The minimum distance for MLSE digital data systems of limited complexity," IEEE Trans. Inform. Theory, vol. IT-21, pp. 544-551, Sept. 1975.

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

[24] A. V. Oppenheim and R. W. Schafer, Digital signal processing New Jersey: Prentice-Hall Inc., 1975, ch. 2 & 8.

[25] H. L. Van Trees, Detection, Estimation and Modulation Theory, New York: Wiley, 1968, ch. 2.