

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

**REDUCED COMPLEXITY
RECEIVERS
FOR TRELLIS CODED
MODULATIONS
VIA PUNCTURED TRELLIS CODES**

by
Bonchul Koo

Thesis submitted to the Faculty of the Graduate School of
the New Jersey Institute of Technology in partial fulfillment of
the requirements for the degree of
Master of Science in Electrical Engineering
1989

APPROVAL SHEET

Title of thesis : Reduced complexity receivers for trellis coded modulations
via punctured trellis codes

Name of Candidate : Bonchul Koo
Master of Science in Electrical Eng., 1989

Thesis and Abstract Approved :

Dr. Fidel Morales – Moreno March 15, 1989
Professor Date
Department of Electrical Engineering

Dr. Joseph Frank 3/16/89
Professor Date
Department of Electrical Engineering

Dr. Chung H. Lu 3/15/1989
Professor Date
Department of Electrical Engineering

VITA

Name : Bonchul Koo

Permanent address : 621 YongHyun 5-Dong Nam-Gu Incheon, Korea

Degree and data to be conferred : M.S.E.E., 1989

Secondary education : Jemoolpo high school, Incheon, 1977

Collegiate institutions attended	Date	Degree	Date of degree
----------------------------------	------	--------	----------------

New Jersey Institute of Tech.	09/87	M.S.E.E.	05/89
-------------------------------	-------	----------	-------

Inha University	03/77	B.S.	02/81
-----------------	-------	------	-------

Major : Electrical Engineering

Positions held :

Manager : Korea Telecommunication Authority, Seoul, Korea

ABSTRACT

Title of thesis: REDUCED COMPLEXITY RECEIVERS FOR TRELLIS
CODED MODULATIONS VIA PUNCTURED TRELLIS CODES

Bonchul Koo, Master of Science in Electrical Engineering, 1989

Thesis directed by : Assistant Professor in Electrical Engineering,
Fidel Morales-Moreno.

We introduce a new concept, called *matched punctured trellis encoding*, that simplifies the complexity of Maximum Likelihood Sequence Estimation (MLSE) receivers for combined trellis encoding and modulations with memory. Matched punctured trellis encoding is applied to tamed frequency modulation (TFM) which is a bandwidth efficient correlative - FM scheme. TFM finds applications in satellite, microwave radio, and mobile communications.

Our approach is based on puncturing a rate - $1/2$ matched convolutional code to obtain a rate - $2/3$ mismatched code. A matched code is one that produces trellis coded modulations of minimum complexity. Puncturing these codes to obtain mismatched codes of higher rates increases the complexity of the trellis coded modulations and in return one can achieve greater coding gains. However, the main idea here is that using suboptimum MLSE receivers, with just the complexity of the matched codes, good coding gains can still be achieved. Furthermore, we conclude that the new rate - $2/3$ coded modulations obtained with our approach achieve greater coding gains (for same complexity comparisons) than previously published work.

The new codes are obtained by exhaustive computer search techniques and coding gains of up to 5.73 dB for 32 decoder states are achieved. These new codes are good for use with TFM modulation in an AWGN channel.

Blank Page

To my parents and to my wife

ACKNOWLEDGEMENTS

I am profoundly grateful to Professor Fidel Morales - Moreno for his constant instruction and guidance throughout the research and writing of this dissertation; the numerous discussions with him have always been a major source of stimulation. Thanks are also due to my parents, Jasun Koo and Yeunae Park, for their moral support; and to President of Korea Telecommunication Authority and other officers for all financial support. I wish to say a special thank-you to President Haewook Lee and vice President, Yungseon Hwang and Director of General Administration Group, Hyundae Shin and so on. And finally, to my wife, Namsoon Yoo, her trust in my life and to my son and daughter, Jaemo and Juyeun.

ABSTRACT	1
ACKNOWLEDGEMENTS	2
1. INTRODUCTION	3
1.1 MOTIVATION	3
1.2 THESIS OBJECTIVE	4
1.3 THESIS OVERVIEW	5
1.4 THE CONCEPT OF PUNCTURED CODES	6
2. UNCODED TFM	7
2.1 THE CONCEPT OF TFM	7
2.2 DESCRIPTION OF TFM MODULATOR	7
2.3 PERFORMANCE OF TFM	9
3. TRELLIS CODED TFM	10
3.1 THE CONCEPT OF TRELLIS CODED TFM	10
3.2 EQUIVALENT TRELLIS CODED TFM	13
3.2.1 PRECODED TRELLIS CODED TFM	14
3.3 EUCLIDEAN DISTANCE	14
4. MATCHED CODES FOR TFM	18
4.1 EXAMPLE	18
4.2 MATCHING CONDITIONS	19
5. PUNCTURED TRELLIS CODES	22
5.1 MOTIVATION	22
5.2 METHOD	23
5.2.1 THE CODE STRUCTURE	23
5.2.2 DECODING CONSIDERATIONS	24

5.3 SOLUTION	25
5.3.1 THE STRUCTURE OF PUNCTURED TRELLIS CODE ...	25
5.3.2 OPTIMIZATION APPROACH	26
5.3.3 EXAMPLE	27
5.4 NUMERICAL RESULTS	28
6. CONCLUSION AND DISCUSSION	31
APPENDIX - A SIMPLIFIED FORMULAR FOR NSFED	33
APPENDIX - B COMPUTER PROGRAMS	40
REFERENCES	50
LIST OF FIGURES	53
LIST OF TABLES	55

1. INTRODUCTION

It is known that by using partial response pulse shaping in FM (frequency modulation) signals one can obtain bandwidth efficient modulations schemes. Tamed Frequency Modulation (TFM) [1] can be seen as an instance of partial response FM with Nyquist - 3rd pulse shaping [10]. TFM is a continuous - phase modulation (CPM) scheme that exhibits a very narrow power spectrum with low level sidelobes. Thus, TFM is a good modulation scheme for use in the transmission of digital data in narrow-band channels such as satellite, mobile, and line-of-sight microwave radio. The superior spectrum of TFM, in comparison with that of the better known Minimum Shift Keyed Signal, is achieved by careful control of the phase argument of the modulated carrier with a particular premodulation filter (a specific 3 - tap transversal filter) and a Nyquist - 3 low pass filter in cascade.

1.1 MOTIVATION

The Maximum Likelihood or Viterbi decoding algorithm was discovered and analyzed by Viterbi [20] in 1967. The Viterbi decoding was first shown to be an efficient and practical decoding technique for short constraint length codes. Forney and Omura [21] demonstrated that the algorithm was in fact maximum likelihood. The maximum likelihood decoder would calculate the likelihood of the received data for code symbol sequences on all paths through the trellis. The path with the largest likelihood would then be selected, and the information bits corresponding to that path would

form the decoder output. Unfortunately, the number of paths increases exponentially by constraint length and code rate. With Viterbi decoding, it is possible to greatly reduce the effort required for maximum likelihood decoding by taking advantage of the special structure of the code trellis. In combined encoding ($R = k/l$) and modulation schemes we use the soft decision decoding replacing the Hamming metric as likelihood distribution [8]. In the case of decoding for an AWGN (additive white gaussian noise) channel, the appropriate optimization criterion is to maximize the free Euclidean distance between signals of any two distinct information sequences.

However, even with the Viterbi algorithm, MLSE decoders are very complex for most powerfull trellis coded modulations. Any reduction of the complexity of these receivers is always welcome, as long as performance is not severely deteriorated. This is our motivation, namely, to study the possibility of reducing the complexity of MLSE receivers for trellis coded modulation without severely reducing the performance. In this thesis we do not try to give a general solution, yet, we believe our method can be applied to other modulations as well.

1.2 THESIS OBJECTIVE

The objective of this thesis is to obtain reduced complexity MLSE receivers for trellis coded TFM modulation and at the same time to achieve good coding gains. We do so by puncturing a special class of convolutional codes of $R = 1/2$, called matched codes, to obtain so-called mismatched codes of $R = 2/3$. This is our main idea and starts from the fact that implementation of the Viterbi algorithm

(*Maximum Likelihood decoder*) for high - rate convolutional codes is greatly simplified if the code structure is constrained to be that of a punctured low rate code [6]. Our concept does not only stop there, but also exploits the inherent structural properties of TFM modulation.

In the standard approach of coding with rate - $2/3$ codes, the implementation is complicated by the fact that 2^k paths enter each node in the trellis rather than just two paths as in rate - $1/2$ codes. However, by puncturing a rate - $1/2$ code to obtain a rate - $2/3$ code one can decode the latter as if it were a rate - $1/2$ code, with very little additional complexity. A search for rate - $2/3$ trellis coded TFM schemes, originating from rate - $1/2$ codes, has been made. Yet, there is something more; the codes that we have used here have special attributes, which we explain in the following chapters.

1.3 THESIS OVERVIEW

The development of the thesis is covered in chapters 2 - 5 and conclusions are given in chapter 6. In chapter 2, we establish that Tamed Frequency Modulation is a modulation with memory and thus can be modeled as a Markov process. Especially, we describe TFM by means of a finite - state sequential machine.

TFM scheme is a modulator with memory and is chosen among other memory schemes because of its relative simplicity. Yet, it is complex enough to let us introduce most of our ideas. A main part of this thesis is involved with the understanding of the effects of the interaction between the memory of both the convolutional code and the modulator, on the structure, optimization, realization, and complexity of the

combinations.

In chapter 3, the concept of trellis coded TFM modulation is generalized. Also, we show a technique which explicitly identifies the combined coding and modulation system. The suitability of this identification is that the memory of the modulator can be modeled as a binary encoder. We also introduce the idea that catastrophic encoders can be optimal encoders for TFM without exhibiting catastrophic error propagation. For the code rates of interest, the mathematical detail for the matching or mismatching conditions are also shown in this chapter.

In chapter 5, we introduce a new concept that reduces the complexity of the decoder operation. The concept of puncturing technique is applied to matched encoder for TFM. Mismatched and matched codes are also considered.

Finally, conclusions are given in chapter 6. The proof of some of the formulas and results of this thesis are given in Appendix A.

1.4 THE CONCEPT OF PUNCTURED CODES

In this thesis puncturing technique is used to obtain our goal. Clark and Cain [6] introduced punctured codes by using the notation established by Forney [7]. Now consider a rate-1/2 encoder with each constraint length. The output bits (of this encoder) by two input bits are four. If every fourth encoder output bit (by the second input bit) is deleted (or punctured), this encoder will produce three channel bits for every two data bit; i.e., it will be a rate-2/3 encoder. This rate-2/3 codes (obtained by puncturing technique) are called punctured codes.

2. UNCODED TFM

2.1 THE CONCEPT OF TFM

Tamed Frequency Modulation (TFM) belongs to a class of constant envelope modulation methods among at bandwidth economy with practical equipment. The superior spectrum of TFM, in comparison with that of the better known Minimum Shift Keyed signal, is achieved by careful control of the phase argument of the modulated carrier with a particular premodulation filter (a specific 3 - tap transversal filter) and Nyquist - 3 lowpass filter in cascade. In Fig. 1, we generate TFM. As was mentioned before, TFM belongs to the subclass of the partial response CPM signals.

2.2 DESCRIPTION OF TFM MODULATOR

A TFM signal is represented by

$$s(t) = \sqrt{\frac{2E}{T}} \cos(\omega_c t + \phi(t, \alpha)), \quad (2.1)$$

where $\omega_c = 2\pi f_c$ and f_c is the carrier frequency, ϕ is the continuous phase function bearing the transmitted information. T is the symbol time, and E is the symbol energy. A basic scheme for the generation of $s(t)$ is shown in Fig. 1. With TFM, the allowable change in $\phi(t, \alpha)$ over one bit period is restricted to either $0, +\pi/4, -\pi/4$ or $+\pi/2, -\pi/2$, determined by three consecutive input binary data bits.

The information carrying phase is

$$\phi(t, \alpha) = \pi \int_{-\infty}^t \sum_{i=-\infty}^{\infty} c_n p(\tau - iT) d\tau \quad ; \quad -\infty < t < \infty, \quad (2.2)$$

$p(t)$ is the frequency pulse function and $c_n \in \{+1, -1, +\frac{1}{2}, -\frac{1}{2}, 0\}$. The allowable phase shifts of the modulated carrier during the m th bit period can be expressed as

$$\phi(mT + T, \alpha) - \phi(mT, \alpha) = \frac{\pi}{2} (ab_{m-1} + Bb_m + ab_{m+1}), \quad (2.3)$$

where a is $1/4$ and B is $1/2$, and the input data bits b_{m-1} , b_m , and b_{m+1} at time instants of $(m-1)T$, mT and $(m+1)T$, respectively, are either $+1$ or -1 . The Fourier transform of the frequency pulse function for the TFM is given by

$$G(f) = S(f)H(f). \quad (2.4)$$

$H(f)$ is a low-pass filter satisfying the Nyquist-3rd criterion [10]. $S(f)$ is a 3-tap transversal filter that extends the influence of an input data over three bits period for the phase function $\phi(t)$, i.e., input sequence of binary impulses is shaped by the premodulator. The transversal filter and Nyquist- 3 shaping were originally included to meet conditions which allow the use of a relatively simple quadrature demodulator. According to eq. (2.2) this normalizing $p(t)$ can be expressed as :

$$\int_{-\infty}^{\infty} p(t) dt = 1/2.$$

2.3 PERFORMANCE OF TFM

The TFM signal is assumed to be transmitted over an additive , white and gaussian channel having a one-sided noise power spectral density. A detector which minimizes the probability of erroneous decision must observe the received signal $r(t)$ over the entire time axis and choose the infinitely long sequence b_n . This is referred to as maximum likelihood sequence estimation (MLSE). A sub-optimum detector has been proposed [13] in which the signal is observed for N symbol intervals, to make a decision on a specific data symbol. The limiting case ($N \rightarrow \infty$) of this suboptimum detector is the MLSE detector. Thus, the receiver observes the signal

$$r(t) = s(t) + n(t) \quad ; \quad 0 \leq t \leq NT \quad (2.5)$$

and if we let $N \rightarrow \infty$, it can be shown [13] that an MLSE detector is obtained. Using the union bound which is known to be tight at large SNR [13] it can be shown that an upper bound on the probability of erroneous decision is achieved [5],

$$Pr(e) \leq 2^{N-1} Q\left(\sqrt{\frac{E_b}{N_0}} d_{min,N}\right), \quad (2.6)$$

where E_b is the bit energy (in binary systems E_b equals E , the symbol energy). The $d_{min,N}$ is the normalized minimum Euclidean distance up to time N . Thus, $d_{min,N}$ is the single most important parameter which determines the error-rate performance of digital modulation schemes in an AWGN channel. For this reason we shall use $d_{min,N}$ as a measure of performance.

3. TRELLIS CODED TFM

3.1 THE CONCEPT OF TRELLIS CODED TFM

Fig. 2 shows the combined encoding and TFM communication system model. \mathbf{G} is a rate - k/l binary convolutional encoder represented by a rank - $k, k \times l$ matrix with entries g_{ij} ($i = 1, 2, \dots, k; j = 1, 2, \dots, l$). The code generated by \mathbf{G} is C_G , and is the set of all codewords $\mathbf{b} = \mathbf{a}\mathbf{G}$ where \mathbf{a} ranges through all k -tuples of equiprobable binary sequences. At time t , a k -tuple $\mathbf{a} = (a_t^{(1)}, \dots, a_t^{(k)})$ produces an l -tuple $\mathbf{b} = (b_t^{(1)}, \dots, b_t^{(l)})$, called a *branch* in the trellis of C_G , that is serially transmitted to TFM modulator \mathbf{M} . The TFM modulator \mathbf{M} is shown in Fig. 1. The 3 - tap transversal filter is a partial - response encoder with transfer function ; $\frac{1}{4}(1+2D+D^2)$, where D is the unit - delay operator corresponding to one modulator interval T . It produces a sequence $\{c_n\}$ of quinary symbols , $c_n \in \{+1, -1, +\frac{1}{2}, -\frac{1}{2}, 0\}$,

$$c_n = 2c'_n - 1 \tag{3.1}$$

$$c'_n = \frac{1}{4}(b_n + 2b_{n-1} + b_{n+2}) \tag{3.2}$$

where $b_n \in \{0, 1\}$. The *VCO* (voltage controlled oscillator) in the modulator \mathbf{M} produces the *CPM* signal

$$s(t) = A \cos(2\pi f_c t + \phi(t) + \phi_0) \tag{3.3}$$

where f_c is the carrier frequency, ϕ_0 is an arbitrary constant phase shift that we set to zero with no loss of generality, and $\phi(t)$, the *excess - phase*, is

$$\phi(t) = 2\pi h \int_0^t c_n \sum_n p(x - nT) dx, \quad (3.4)$$

where $\{c_n\}$ is as in (3.1) ; $A = (2E/T)^{1/2}$, where T is the duration of a symbol b_n and E is the energy of a symbol b_n ; and $h = 1/2$ is the modulation index. We also assume that the duration and energy of a symbol a_n are T_b and E_b , respectively, i.e., the rate of \mathbf{G} is $k/l = T/T_b = E/E_b$, which is important for later normalizations. The pulse - shape $p(t)$ in (3.4) satisfies the Nyquist - 3rd criterion [10], i.e.,

$$\int_{nT}^{(n+1)T} p(t) dt = \begin{cases} 1/2 & \text{if } n = 0 \\ 0 & \text{if } n \neq 0. \end{cases} \quad (3.5)$$

The minimum bandwidth solution for $p(t)$ in (3.4) corresponds to TFM as originally introduced in [3]. For simplicity, we will refer to any signal in the class of signals (as defined in (2.1)) as a TFM signal.

The signal at the input of the Maximum Likelihood decoder *V.D.* is $r(t) = s(t) + n(t)$, where $n(t)$ is a Gaussian random process with zero mean and variance $N_0/2$. The MLSE receiver *V.D.* uses the Viterbi algorithm to find the k -tuple of sequences \hat{a} which leads to a signal $\hat{s}(t)$ that is closest to $r(t)$ in the sense of smallest Euclidean distance.

For a given pulse-shape $p(t)$ there is a phase trellis diagram describing the (in-

stantaneous) excess-phase value $\phi(t)$. However, at the symbol transitions, $\phi(t)$ is not a function of $p(t)$, since from (3.4) and (3.5), with $h = 1/2$,

$$\phi(mT) = \pi \sum_{n=0}^{m-1} c_n \int_0^{mT} p(t - nT) dt = \frac{\pi}{2} \sum_{n=0}^{m-1} c_n. \quad (3.6)$$

Phase values that differ by an integer multiple of 2π are not physically distinguishable. We define ϕ_m as the excess-phase modulo 2π , i.e., $\phi_m = \phi(mT)$ modulo -2π . Consequently,

$$\phi_m \in \{0, -\pi/4, +\pi/4, -\pi/2, +\pi/2, -3\pi/4, +3\pi/4, \pi\}.$$

The pattern of phase ϕ_m that results from a sequence of symbols c_n can be described by a finite-state sequential machine. Using ϕ_m and the symbols ($b_{n-1}, b_{n-2}, b_0 = 0$), we can find a 16-state machine representation of TFM modulation. This machine is however nonminimal, i.e., it can be reduced to another (*equivalent*) machine with less number of states. Since the complexity of the combinations of encoding and modulations is used as the optimization constant here, it is important to find a minimal machine representation of TFM. A minimal (hence *irreducible*) 8-state diagram interpretation of TFM can be obtained by considering excess-phase values about the tone frequency $f_1 = f_c - 1/4T$. Thus rewriting (3.3) as

$$s(t) = A \cos(2\pi f_1 t + \theta(t)), \quad (3.7)$$

we can define $\theta(t) = \phi(t) + \pi t/4T$ as the excess-phase about the frequency f_1 . At the symbol transitions $\theta(mT)$ is given by

$$\theta(mT) = \frac{\pi}{2} \sum_{n=0}^{m-1} (c_n + 1) \quad (3.8)$$

independently of the Nyquist-3 pulse-shape $p(t)$.

Consequently,

$$\theta_m \in \left\{ 0, +\frac{\pi}{4}, -\frac{\pi}{4}, +\frac{3\pi}{4}, -\frac{3\pi}{4}, \pi \right\},$$

where, $\theta_m = \theta(mT)$ modulo- 2π , and we have assumed that (3.1) satisfies the initial conditions ; $b_{-1} = b_{-2} = 0$. Using the phase values θ_m as opposed to ϕ_m we obtain the 8-state Mealy machine of Fig. 3.

3.2 EQUIVALENT TRELLIS CODED TFM

For binary convolutional codes C_G , we can show that an encoder \mathbf{G} contains feedback if there is at least one finite input sequence $\{a_n\}$ for which the encoder produces an infinite output sequence $\{b_n\}$, (a finite binary sequence is any sequence $\{x_n\}$ with a finite number of nonzero symbols). To illustrate the above assume \mathbf{G} is initially at the zero encoder state and a finite sequence $\{a_n\}$ is applied at its input.

\mathbf{G} contains feedback if for that input sequence the final encoder state is not the zero state. In this manner we can show that a TFM modulator contains internal feedback. For this we use the machine representation of Fig. 3. Let $\{b_n\} = 1, 0, 0, \dots$, be the input (*finite*) sequence to this machine, and assume the initial state is the zero state. Since the machine never returns to the initial state, it contains internal feedback. A consequence of the feedback inherent in TFM is that sometimes catastrophic

encoders \mathbf{G} will produce noncatastrophic trellis coded TFM combinations. It is said that a trellis coded TFM is to be catastrophic if and only if there exist k -tuples of sequences a_i and a_j with infinite Hamming distance producing signals $s_i(t)$ and $s_j(t)$, $-\infty \leq t \leq \infty$, with finite Euclidean distance.

3.2.1 PRECODED TRELLIS CODED TFM

The precoded TFM modulator is described by the Mealy machine of Fig. 4 and the corresponding *transformed* communications system is shown in Fig. 5. The states in the machine of Fig. 4 are labeled as in Fig. 3, and the branches by the input symbols to the precoder, i.e., by b_n . The relation between \mathbf{G} and \mathbf{W} is established by the precoder $1 + D$, as seen by comparing Fig. 2 and 5. Thus, given an encoder \mathbf{W} for use with a precoded TFM modulator, we can find the corresponding encoder \mathbf{G} for the original TFM modulator \mathbf{M} by applying the precoder rule $1 + D$ to the output of \mathbf{W} . In what follows we assume the equivalent TFM machine of Fig. 4 and the communications system of Fig. 5. In this scheme by precoding the input to \mathbf{M} with the rate-one binary encoder with transfer function a catastrophic encoder G will be transformed to a noncatastrophic encoder G^* .

3.3 EUCLIDEAN DISTANCE

Let the sequences \underline{U}_i and \underline{U}'_i correspond to the channel symbol sequences α and β , respectively, where i is $0, 1, \dots, l - 1$. The corresponding signals $s(t, \alpha)$ and $s(t, \beta)$ are both assumed to be in the same state number at level 0. Let the modulated signal

over the interval $nT \leq t \leq (n+1)T$ be described as

$$s(t, \alpha) = \sqrt{\frac{2E}{T}} \cos(\omega_c t + \phi(t, \alpha)). \quad (3.9)$$

It is also assumed that \underline{U}_i and \underline{U}'_i are such that $\underline{U}_0 \neq \underline{U}'_0$ and such that $s(t, \alpha)$ and $s(t, \beta)$ are in the same state for the first time in level l . The situation described above is an example of an error event of length l intervals. Error events are essential in the performance analysis of the receiver. One of the most important characteristics of an error event is its Euclidean distance. Especially, for large signal - to - noise ratios (E_b/N_0), the error probability behaviour is dominated by the minimum Euclidean distance in the set of all error events.

For the Gaussian channel, path metrics are correlations between the received signal and the hypothesized transmitted signal. With or without convolutional coding, the squared Euclidean distance $d^2(i, j)$ between any two transmitted signals $s_i(t)$ and $s_j(t)$ is

$$d^2(i, j) = \sum_n d_n^2(i, j), \quad (3.10)$$

where,

$$d_n^2(i, j) = \int_{nT}^{(n+1)T} [s_i(t) - s_j(t)]^2 dt \quad (3.11)$$

is the distance contribution attributable to the interval $nT \leq t \leq (n+1)T$. Our interest, however, is in determining the asymptotic performance of such a receiver,

which for the assumed channel requires determining the minimum distance between any two signals that are merged at some time, split, then remerged later. Or, for the finite - memory receivers we seek the minimum distance between pair that split and are of length N , denoted $d_{min,N}$. We also define the free distance

$$d_{free} = \lim_{N \rightarrow \infty} d_{min,N} \quad (3.12)$$

as usual. The appropriate distance metric for this problem is

$$d_{min,N}^2 = \frac{1}{2E_b} \int_0^{NT} [s_i(t) - s_j(t)]^2 dt \quad (3.13)$$

which, for $\omega_{ci} \gg 2\pi/T$, reduces to

$$d_{min,N}^2 \simeq \frac{R}{T} \int_0^{NT} [1 - \cos \Delta\phi(t)] dt \quad (3.14)$$

where, $\Delta\phi(t)$ is the time - varying phase separation between the two signals of length NT seconds. Also, we note that the cumulative squared distance may be recursively computed as

$$d_{min,n+1}^2 = d_{min,n}^2 + \frac{R}{T} \int_{NT}^{(N+1)T} [1 - \cos \Delta\phi(t)] dt, \quad (3.15)$$

and is completely specified by transitions in the pair - state trellis ($d_{min,n}; 0 \leq n \leq N$). The task now is to find the smallest distance between *any* pair - state corresponding to a merger at level $n = 0$ and another merged pair - state at some level n , while

transitioning between pair states in each trellis. This is completely analogous to the shortest route problem and may be solved by applying the dynamic programming solution. The principle of optimality here is that if a sequence pair (α_1, α_2) is to produce the minimum distance event, it will do so via extension of minimum distance pairs to some intermediate pair state, and this holds for all n . Thus, it is sufficient to preserve the information associated with the minimum distance pair for each pair state at each level n , and proceed forward recursively, using the known pair state transitions,

$$\Delta\phi(t) = \phi(t, \alpha) - \phi(t, \beta),$$

and

$$d_{min,N}^2 \simeq R \left[N - \frac{1}{T} \int_0^{NT} \cos[\Delta\phi(t)] dt \right].$$

See Appendix A for more details.

4. MATCHED CODES FOR TFM

Our optimization criterion is maximum NSFED (normalized squared free Euclidean distance) for given code rate and given total number of states in the combined encoding and trellis TFM diagram (or state transition diagram). Let S_V denote the total number of states in the trellis of a combination, and let S_W be the total number of states in the trellis of a code W in Fig. 4. For a given code rate we will also call S_V and S_W the complexity of the combination and the complexity of the convolutional code, respectively.

4.1 EXAMPLE 1

Consider now the rate - 1/2 convolutional encoder of Fig. 6.1 with generator matrix

$$W^* = [1 + D, 1] \tag{4.1}$$

and constraint length $v = 1$. The code C_{W^*} generated by this encoder has a trellis with $S_{W^*} = 2$ states and is shown in Fig. 6.2. When this code is used with a precoded TFM modulator, we find that $S_V=4$ after drawing the combined diagram as we see it in Fig. 7. We can conclude from this example that it is possible to obtain a decoder complexity of $S_V = 4$ states, using convolutional codes of different complexities S_W (i.e. of different constraint lengths). In other words, when the (Viterbi) decoder complexity S_V becomes the fixed parameter, the number of encoder states S_W can take different values.

4.2 MATCHING CONDITIONS

Let C_W be any rate - k/l convolutional code ; let \mathbf{W} be an encoder for C_W ; let S_W be the total number of states in the trellis of C_W ; and let S_V be the total number of states required in the combined coding and TFM trellis :

Lemma 1 (rate-1/2 and rate-2/3 codes)

For codes of rate 1/2 and 2/3 and for every $S_V \geq 4$, there are exactly two distinct classes of convolutional codes producing the required value for S_V ; the class of *mismatched* codes C_W where,

$$S_W = \frac{S_V}{4}, \quad (4.2)$$

and the class of *matched* codes C_{W^*} , with $S_{W^*} = 2S_W$ states, where

$$S_{W^*} = \frac{S_V}{2}. \quad (4.3)$$

Proof : See reference [1].

Thus, once the decoder complexity (S_W) is fixed, the concept of matched and mismatched codes arises very naturally. The structural properties of convolutional codes satisfying (4.3) are given by the following Corollaries to Lemma 1.

Corollary 1 (rate-1/2 codes)

Let W have polynomial entries w_j , $j = 1, 2$. Then , a rate - 1/2 code C_W satisfies (4.3) if and only if ,

$$\deg(w_2) \leq v - 1, \ ; \ v \geq 1 \tag{4.4}$$

where, v is the constraint length of W .

If w_2 is $1 + D$ we have, $\deg(w_2) = \deg(1 + D) = 1$, i.e.. Since the code $[1 + D, 1]$ satisfies (4.4) the encoder in (4.1) (Example) is a matched encoder.

The structure of rate - 2/3 matched (or mismatched) codes is established by Corollary 2. Let

$$v_i = \max_{1 \leq j \leq l} \deg(w_{ij})$$

be the constraint length for input i in the "obvious realization " of W and define

$$v = \sum_i v_i$$

as the overall constraint length of a convolutional encoder [7].

Corollary 2 (rate-2/3 codes)

Let W have polynomial entries W_{ij} , $i = 1, 2$; $j = 1, 2, 3$. A rate - 2/3 code C_W satisfies (4.3) if and only if ;

i) for $v = 2r$, $r = 1, 2, 3, \dots$ (hence , $v_1 = v_2 = v/2$) ,

$$\deg(w_{ij}) = \frac{v}{2} \tag{4.5.a}$$

for all i and for those j where $\deg(w_{1j}) = v/2$, and

ii) for $v = 2r - 1$, $r = 1, 2, 3, \dots$

$$\begin{cases} \deg(w_{ij}) = v_2 - 1 & \text{for those } j \text{ where } \deg(w_{2j}) = v_2, \text{ and} \\ \deg(w_{2j}) = v_1 + 1 & \text{for those } j \text{ where } \deg(w_{ij}) = v_1. \end{cases} \quad (4.5.b)$$

In general, for a given complexity S_V and a given code rate - k/l , we say that W (C_W), with complexity S_W states, is a *matched* encoder (code) for TFM if and only if S_W is the minimum possible value, for that rate, that yields the required S_V , i.e., if and only if the state difference

$$S_{V-W} = S_V - S_W$$

is a minimum for that rate. Otherwise, W (C_W) is called a *mismatched* encoder (code) for TFM. Let C_W be any rate - k/l convolutional code with (overall) constraint length v ; then, C_W is a matched code for a modulator if the corresponding complexity S_V in the combined trellis is the minimum possible rate for that constraint length.

5. PUNCTURED TRELLIS CODES

5.1 MOTIVATION

TFM signals can be detected by either a coherent or a noncoherent demodulator with an MLSE receiver. The MLSE receiver maximizes the likelihood function,

$$\log_e [P_{r(t)|\alpha}(r(t)|\alpha)] \sim - \int_{-\infty}^{\infty} [r(t) - s(t, \alpha)]^2 dt.$$

With respect to the infinitely long estimated sequence α , the maximizing sequence α is the maximum likelihood sequence estimate and $P_{r(t)|\alpha}$ is the probability density function for the observed signal $r(t)$ conditioned on the infinitely long sequence α . It is equivalent to maximize the correlation

$$\sim \int_{-\infty}^{\infty} r(t)s(t, \alpha)dt.$$

The criterion for deciding between two paths through the trellis is to select the one having the larger metric. Thus, the number of computations in decoding performed at each stage increases exponentially with constraint length. The exponential increase in computational burden limits the use of the Viterbi algorithm to relatively small values of the constraint length and code rate. In this chapter we introduce a method to simplify the Viterbi decoders for high - rate convolutional codes.

5.2 METHOD : PUNCTURED CONVOLUTIONAL CODES

In the standard approach to decoding these codes the implementation is complicated by the code structure which has 2^k paths entering each state rather than just two paths as rate $-1/l$. This makes the resulting comparison and selection of the path with the best metric much more difficult. The puncturing technique can solve this problem. Also, we can decode just as one would decode a rate $-1/2$ code, with very little additional complexity.

5.2.1 THE CODE STRUCTURE

Our discussion of punctured code structure uses the notation established by Forney [7]. The code constraint length is defined to be v , which has the minimum number of memory elements. A code is represented by its generator polynomial matrix $G(D)$. The element in the j th row and i th column,

$$G_j^i(D) = g_{0j}^i + g_{1j}^i D + \cdots + g_{vj}^i D^v, \quad (5.1)$$

relates the i th output sequence to the j th input sequence. The punctured code approach will be illustrated using the rate $-2/3$, $v = 2$ code with generator matrix

$$G(D) = \begin{bmatrix} 1 + D & 1 + D & 1 \\ D & 0 & 1 + D \end{bmatrix}. \quad (5.2)$$

The trellis structure for this code is shown in Fig. 8. In decoding this code we use the Viterbi algorithm in a convolutional manner, a 4 - ary comparison per state must be

made for every two information bits. This is in contrast to the much simpler binary comparisons performed in decoding rate - $1/l$ codes.

Now consider the rate - $1/2$, $v = 2$ code with generators $(1 + D + D^2)$, and $(1 + D^2)$. If every fourth encoder output bit is deleted, this code will be a rate - $2/3$ code. In fact, if the bit from the second generator $(1 + D^2)$ is deleted from every other branch, the rate - $2/3$ code in (5.2) is obtained. This code has the trellis shown in Fig.9, where X indicates the deleted bits. Note that the transitions between states and the resulting transmitted bits are identical in Fig. 8 and 9. The transition is through a set of intermediate states since only one bit at a time is shifted into the encoder rather than two. Obviously we can generate the same code in a different manner.

5.2.2 DECODING CONSIDERATIONS

In section 5.2.1 a rate - $2/3$ code was constructed by periodically deleting bits from a rate- $1/2$ code. It was shown that a rate - $2/3$ encoder can be produced from a rate - $1/2$ encoder by mechanical control. This so-called punctured code could also be thought of as an rate - $1/3$ code form with generators $1 + D + D^2$, $1 + D^2$, $1 + D + D^2$ that has the first two bits deleted on one branch and the the third bit deleted on the next branch. The practical value of the punctured code approach is obvious. We can implement a rate - $2/3$ decoder as a rate - $1/2$ decoder with additional technique to stuff erasures in the locations of the deleted bits. After the erasures are stuffed, decoding proceeds just as if the code were a rate - $1/2$ code. In this fashion we

replace the complex 2^k -ary comparisons at each state by binary comparisons. In a high-speed decoder this simplification has a major impact on decoder complexity. The performance will be evaluated on the assumption of an AWGN channel at a large signal - to noise ratio. The asymptotic performance for a rate - k/l code is given by [15].

$$P_e \sim \frac{w_i(d)}{k} Q\left(\sqrt{\frac{2dk}{l} \frac{E_b}{N_0}}\right)$$

where d is the free Hamming distance of the code and $w_i(d)$ is the total input weight of all information sequences which produce weight d paths.

5.3 SOLUTION : PUNCTURED TRELLIS CODES

A new concept, called *punctured trellis encoding*, is introduced in this section. In chapter 4, the matched encoder (code) was fully described. The condition of punctured trellis code is that a rate - $1/2$ code must be matched for TFM to produce a rate - $2/3$ punctured trellis code. The mean of punctured trellis here is of a trellis diagram of combined coding and TFM modulation. Punctured trellis of rate - $2/3$ system is equal to that of rate - $1/2$ except deleting function. Also, we can use a rate - $1/2$ decoder trellis for a rate - $2/3$ decoder trellis in decoding the information sequences. This is our main idea.

5.3.1 THE STRUCTURE OF PUNCTURED TRELLIS CODES

In section 5.2 the general structure of punctured code has been described. The punctured trellis code for TFM needs one more condition ; the rate - 1/2 encoder (code) must be matched for TFM.

The punctured code approach will be illustrated using the $R = 2/3, v = 1$ code with generator matrix

$$G(D) = [1 + D , 1 , 1 + D]. \quad (5.3)$$

The trellis structure for this code is shown in Fig. 10. Now, consider $R = 1/2 , v = 1$ code with generator $[1 + D, 1]$. Let two input bits be ; $a_1, a_2 : 0 , 0 ; 0 , 1 ; 1 , 0 ; 1 , 1$. Since the rate of this code is 1/2, the number of output bits is two for one input bit. In the same manner we can produce four output bits by every two input bits. Let us assume the output bits are b'_1 , b''_1 by a_1 and b'_2 , b''_2 by a_2 . If every fourth encoder output bit, in other words, b''_2 is deleted, this encoder will produce three channel output bits ; b'_1 , b''_1 , b'_2 , for every two input data bits ; a_1 , a_2 .

A punctured rate - 2/3 encoder produced from a rate - 1/2 encoder has a simple structure which can be easily obtained from the rate - 1/2 structure. The simple structure is $[g_1(D) , g_2(D)]$ of a rate - 1/2 matched encoder for TFM and $[g_1(D) , g_2(D) , g_1(D)]$ of a rate - 2/3 punctured mismatched trellis encoder produced from the above rate - 1/2 encoder at the same constraint length.

5.3.2 OPTIMIZATION APPROACH

The optimization criterion is maximum NSFED for a given code rate and given total number of states in the combined encoding and modulation state-trellis diagram. With a code rate of $R = k/l$ and a constraint length of v the total number of states of a matched encoder system is $S_V = 2S_{W^*} = 2^{v+1}$. Rate - 2/3 punctured trellis codes generated from rate - 1/2 matched codes is always mismatched for TFM. The decoder operation of a rate - 2/3 code is the same as that of a rate - 1/2 code. But the Euclidean distance between two signals is that of the rate - 2/3 code. Thus, we can use the decoder trellis of the rate - 1/2 code for decoding the rate - 2/3 system.

5.3.3 EXAMPLE 2

In equation (5.3) we showed the technique to produce punctured trellis codes. The rate - 1/2 code, $v = 1$ convolutional encoder $W^* = [1 + D, 1]$ in (4.1) is a matched encoder for TFM, see Fig. 5. Thus, the total number of states is $S_V = 4$. From this encoder the two different generators for the corresponding rate - 2/3 encoder are obtained as $[1 + D, 1, 1 + D]$. The encoder polynomial matrix of this code is

$$W = \begin{bmatrix} 1 & 1 & 1 \\ D & 0 & 1 \end{bmatrix}.$$

This encoder is a mismatched encoder, $S_W = 4$ since Corollary 2 is not satisfied here. Therefore, the total number of states of this code is $S_V = 4S_W = 8$. We can now obtain the rate - 2/3 encoder obtained from matched rate - 1/2 encoder by using the

puncturing technique. Then, we can also use the structure of the rate - 1/2 code in Fig. 11 for decoding the trellis of the rate - 2/3 code. The optimum decoder trellis of a rate - 1/2 code can be used for decoding a rate - 2/3 code scheme. Therefore, the total number of states of punctured trellis of a rate - 2/3, $v = 1$ is $S_V = 4$, $S_V = 2S_W$ in Fig. 11. Also, comparisons of each state are reduced to 2, yet, comparisons of optimum decoder trellis are 4 in Fig. 12.

5.4 NUMERICAL RESULTS

At this point we note that, as long as $p(t)$ satisfy (3.5), Lemma 1 does not depend on the particular pulse - shape used. This is so because the excess - phase values $\theta(mT)$ are independent of $p(t)$, see (3.8) and Fig. 3 and 5. Consequently, any code matched (mismatched) to, say, precoded TMSK, is also matched (mismatched) to any other precoded TFM modulator. The codes obtained will be best for use with any other TFM scheme. For a given complexity S_V and a given code rate, our goal is to find the mismatched codes, produced by puncturing technique, \mathbf{W} of that rate that produce the best (maximum NSFED) coded TFM schemes of that complexity. Our definition of "*best*" is maximum NSFED for each constraint length.

We will see that, in general, the results obtained with the unpunctured method are the ones obtained if we used the "*best*" matched codes. We can obtain the best punctured codes by the following step;

- 1) Use the puncturing technique to obtain the punctured trellis codes via matched codes for TFM.

- 2) Investigate all possible rate - 1/2 matched codes for TFM.
- 3) Collect both the best rate -1/2 matched and best rate - 2/3 punctured trellis mismatched codes for each code rate and the same complexity.
- 4) Choose a code which produces the greatest NSFED. The chosen codes are the best codes of rate - 2/3.

An extensive search was performed to find the best punctured trellis code generators for several constraint lengths v . The search was over all generators where there are only two different generators of $\{ G_1^*(D), G_2^*(D), G_1^*(D) \}$. This formula has implementation advantages particularly at high speeds in that codes may be implemented as a single rate - 1/2 code with periodically sampled generators. The free Euclidean distance found for the punctured trellis codes compare very favorably with those published in reference [1].

In Table I we reported the results of the computer search. The reported codes are the best rate - 2/3 punctured trellis codes. The value for NSFED and coding gains are given in the Table.

For a given complexity S_V and a given code rate our goal is to find those codes \mathbf{W} of that rate that produce the best maximum NSFED coded TFM schemes of that complexity. We now search for the best code \mathbf{W} of that rate, matched or mismatched, that maximizes the distance NSFED and produces a coded TFM with required complexity S_V . In this search, we have used the algorithm given in [11]. The best codes are reported in Table I and II, with corresponding values for the distance NSFED and

coding gains (relative to the uncoded modulation). Note that the generator polynomials of the reported codes are in binary notation. In Table II we can see that the best rate - 2/3 punctured trellis mismatched codes W outperform the published best codes of that rate for all complexity. The punctured trellis encoder G^* can be easily obtained from the rate - 1/2 encoder (which is a matched encoder). Finally, let us assume that we use this new optimized method. Especially, assume the code rate is 1/2 and 2/3 and complexity S_W is fixed to 2 states, i.e., $S_W=2$. Where, the number of states of rate - 1/2 and 2/3 code trellis is equal, $S_V = 4$ with polynomial matrix of $[1 + D, 1]$ and $[1 + D, 1, 1 + D]$, see Fig. 11. The rate - 1/2 encoder is matched for TFM but the rate - 2/3 encoder is mismatched. The complexity of the rate - 2/3 decoder is $S_V = 4$ states, and NSFED is 2.733, achieving a coding gain of 2.74. We showed how to obtain rate - 2/3 punctured codes from rate - 1/2 non-punctured matched codes in Table III, and the results of regular encoder generator matrix in Table IV.

6. CONCLUSION AND DISCUSSION

The purpose of this thesis is to discuss a situation in which difficult Viterbi decoder design problem can be simplified by suitably changing the code structure, and to develop a theory that allows the optimization of the punctured trellis codes. The trellis coded modulations obtained with this theory are optimum in that the distance, NSFED is maximized for a given code rate and given total number of states in the combined trellis. We have applied this theory to codes of rate - $2/3$ with receiver complexities S_V of from 4 to 32 states.

We found that the best codes can be obtained in the set of the so-called punctured trellis codes for TFM. Also, we showed a way, called puncturing technique, to produce best punctured trellis codes. We can conclude that by using puncturing technique we can obtain rate - $2/3$ punctured trellis codes that have a characteristic of simplifying MLSE receiver structure.

We can also conclude that, because TFM modulation contains internal feedback, sometimes a catastrophic encoder can produce a noncatastrophic combination, as predicted in the chapter 3. We found that a rate - $2/3$ punctured trellis codes W do not satisfy the matching conditions stated in (4.5.a) and (4.5.b). In Table III and IV we compare the best rate - $2/3$ codes W obtained in this thesis.

We showed that we can use the rate - $1/2$ decoder trellis for a rate - $2/3$ decoder trellis in decoding received information sequences. Since the rate - $1/2$ decoder trellis

is equal to rate - $2/3$ except of the deleted information, the number of the rate - $2/3$ decoder trellis states is also equal to that of non-punctured matched rate - $1/2$ scheme.

From a more practical point of view, greater coding gains have been obtained, for same complexity comparisons, than in previously reported work. Indeed, for rate - $2/3$ codes, a coding gain (relative to the uncoded modulation) of 2.74 has been obtained with a decoder complexity of just $S_V = 4$ states. With full complexity we obtained slightly greater coding gains than in optimum decoder in Table II. Also, there are 4 comparisons at each state over the optimum decoder trellis in Fig. 12, yet, we reduced 4 comparisons of optimum decoder trellis to 2. We can, also, conclude that we can reduce the receiver complexity and complexity comparisons to the half by using the method introduced in this thesis.

In conclusion, by using the punctured trellis coded modulations we have obtained half of the receiver complexity relative to the non-punctured optimum encoding without sacrificing performance. Also, with other suitable choices of Nyquist's *3rd* pulse shapes $p(t)$, we can reduce the bandwidth occupancy of trellis codes. Also, it will be concluded that the performance of the system is approximately determined by the minimum Euclidean distance for small symbol error probabilities.

In this thesis we did not try to generalize a theory to apply to other modulation schemes, yet, we believe this new concept can be applied to other modulations.

APPENDIX A

SIMPLIFIED FORMULAR FOR NSFED

We use the following simplified method to calculate the Euclidean distance. Let the modulated signal over the interval $nT \leq t \leq (n+1)T$ be described as

$$s(t, \alpha) = \sqrt{\frac{2E}{T}} \cos(\omega_c t + \phi(t, \alpha) + \phi_0) \quad (1)$$

where, $\phi(t, \alpha)$ is the phase modulation in response to the data sequence

$\mathbf{b} = (b_1, b_2, \dots, b_n)$, E is the energy per symbol, and T is the symbol interval, ϕ_0 is an arbitrary constant phase shift that we set to zero with no loss of generality, where, α is the data sequence

$$\alpha = (\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n).$$

Thus,

$$\begin{aligned} \phi(t, \alpha) &= 2\pi h \int_0^t \sum_{i=-\infty}^n \alpha_i p(t - iT) dt \\ &= 2\pi h \sum_{i=-\infty}^{\infty} \alpha_i \int_0^t p(t - iT) dt \\ &= 2\pi h \sum_i \alpha_i q(t - iT), \end{aligned} \quad (2)$$

where,

$$p(t) = \begin{cases} C_i/T & \text{if } iT \leq t \leq (i+1)T, \quad i = 0, 1, \dots, L-1 \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

and

$$q(t) = \begin{cases} 0 & \text{if } t \leq 0 \\ q(LT) & \text{if } t > LT, \end{cases}$$

and

$$q(t) = \int_0^t h(\lambda) d\lambda. \quad (4)$$

The pulse-shape $p(t)$ in (2) satisfies the *Nyquist's 3rd* criterion, i.e.,

$$q(t) = \int_{nT}^{(n+1)T} p(t) dt = \begin{cases} 1/2 & \text{if } n = 0 \\ 0 & \text{if } n \neq 0. \end{cases}$$

We define $E/E_b = R$ bit rate, and $d_{min,N}^2$ as the normalized squared free Euclidean distance

$$d_{min,N}^2 = \sum_{n=0}^{N-1} d_n^2(\alpha, \beta) = \frac{1}{2E_b} \int_0^{NT} [s(t, \alpha) - s(t, \beta)]^2 dt \quad (5)$$

where,

$$d_n^2(\alpha, \beta) = \frac{1}{2E_b} \int_{nT}^{(n+1)T} [s(t, \alpha) - s(t, \beta)]^2 dt. \quad (6)$$

Then,

$$d_{min,N}^2 = \frac{1}{2E_b} \int_0^{NT} [s^2(t, \alpha) - 2s(t, \alpha)s(t, \beta) + s^2(t, \beta)] dt. \quad (7)$$

Now,

$$\begin{aligned}
\int_0^{NT} s(t, \alpha)s(t, \beta)dt &= \int_0^{NT} \frac{2E}{T} \cos(2\pi f_c t + \phi(t, \alpha)) \cos(2\pi f_c t + \phi(t, \beta)) dt \\
&= \frac{2E}{T} \int_0^{NT} \frac{1}{2} \cos(\phi(t, \alpha) - \phi(t, \beta)) dt \\
&\quad + \frac{2E}{T} \int_0^{NT} \frac{1}{2} \cos(2\pi f_c t + \phi(t, \alpha) + \phi(t, \beta)) dt \\
&\simeq \frac{2E}{T} \int_0^{NT} \cos(\phi(t, \alpha) - \phi(t, \beta)) dt.
\end{aligned} \tag{8}$$

Define $\Delta\phi(t) = \phi(t, \alpha) - \phi(t, \beta)$, hence,

$$d_{min, N}^2 \simeq R \left[N - \frac{1}{T} \int_0^{NT} \cos[\Delta\phi(t)] dt \right],$$

or

$$d_{min, N}^2 = \sum_{n=0}^{N-1} \left[R \left(1 - \frac{1}{T} \int_{nT}^{(n+1)T} \cos[\Delta\phi(t)] dt \right) \right]. \tag{9}$$

Now,

$$\begin{aligned}
\phi(t, \alpha) &= 2\pi h \sum_{i=-\infty}^n \alpha_i q(t - iT) \\
&= 2\pi h \sum_{i=n-L+1}^n \alpha_i q(t - iT) + 2\pi h q(LT) \sum_{i=-\infty}^{n-L} \alpha_i,
\end{aligned} \tag{10}$$

but where, $q(LT) = \frac{1}{2}$ for positive pulse $p(t) > 0$. Hence, for positive $p(t)$,

$$\phi(t, \alpha) = 2\pi h \sum_{n-L+1}^n \alpha_i q(t - iT) + \pi h \sum_{i=-\infty}^{n-L} \alpha_i, nT \leq t \leq (n+1)T \quad (11)$$

and

$$\phi(t, \alpha) = \Omega_i t + \theta_i$$

$$\Omega_i t = 2\pi h \sum_{i=n-L+1}^n \alpha_i q(t - iT)$$

$$\theta_i = \pi h \sum_{i=-\infty}^{n-L} \alpha_i,$$

so,

$$\begin{aligned} \Delta\phi(t) &= 2\pi h \left[\sum_{i=n-L+1}^n \alpha_i q(t - iT) + \frac{1}{2} \sum_{i=-\infty}^{n-L} \alpha_i \right] \\ &\quad - 2\pi h \left[\sum_{i=n-L+1}^n \beta_i q(t - iT) + \frac{1}{2} \sum_{i=-\infty}^{n-L} \beta_i \right] \\ &= 2\pi h \left[\sum_{i=n-L+1}^n (\alpha_i - \beta_i) q(t - iT) + \frac{1}{2} \sum_{i=-\infty}^{n-L} (\alpha_i - \beta_i) \right]. \end{aligned} \quad (12)$$

But, $\alpha_i - \beta_i = 0, i = -\infty, \dots, -1$ and $\gamma_i = \alpha_i - \beta_i$, then,

$$\phi(t) = 2\pi h \left[\sum_{i=n-L+1}^n \gamma_i q(t - iT) + \frac{1}{2} \sum_0^{n-L} \gamma_i \right]. \quad (13)$$

Therefore,

$$d_{min,N}^2 = RN - \frac{R}{T} \sum_{n=0}^{N-1} \int_{nT}^{(n+1)T} \cos \left[2\pi h \left(\sum_{i=n-L+1}^n \gamma_i q(t-iT) + \frac{1}{2} \sum_{i=0}^{n-L} \gamma_i \right) \right] dt. \quad (14)$$

Now,

$$\begin{aligned} d_{min,N}^2 &= \sum_{n=0}^{N-1} \left[R \left(1 - \frac{1}{T} \int_{nT}^{(n+1)T} \cos[\phi(t)] dt \right) \right] \\ &= \left[R - \frac{R}{T} \int_0^T \cos[\phi(t)] dt \right] + \left[R - \frac{R}{T} \int_T^{2T} \cos[\phi(t)] dt \right] + \dots \\ &\quad + \left[R - \frac{R}{T} \int_{(N-1)T}^{NT} \cos[\phi(t)] dt \right], \end{aligned} \quad (15)$$

but,

$$\begin{aligned} \phi(t) &= 2\pi h \left[\sum_{i=n-L+1}^n \gamma_i q(t-iT) + \frac{1}{2} \sum_{i=0}^{n-L} \gamma_i \right] \\ &= (\Omega_i - \rho_i)t + (\theta_i - \zeta_i), \end{aligned} \quad (16)$$

where,

$$\begin{aligned} (\Omega_i - \rho_i)t &= 2\pi h \sum_{i=n-L+1}^n \gamma_i q(t-iT) \\ (\theta_i - \zeta_i) &= \pi h \sum_{i=0}^{n-L} \gamma_i, \end{aligned} \quad (17)$$

and

$$\phi(t, \alpha) = \Omega_i t + \theta_i$$

$$\phi(t, \beta) = \rho_i + \zeta_i, \quad nT \leq t \leq (n+1)T.$$

So,

$$\begin{aligned} & \int_n^{(n+1)T} \cos[(\Omega_i - \rho_i)t + (\theta_i - \zeta_i)] dt \\ &= \frac{\sin[(\Omega_i - \rho_i)t + (\theta_i - \zeta_i)]}{\Omega_i - \rho_i} \Big|_n^{(n+1)T} \end{aligned}$$

Then,

$$\begin{aligned} d_{min,N}^2 &= RN - \frac{R \sin[(\Omega_i - \rho_i)t + (\theta_i - \zeta_i)]}{T(\Omega_i - \rho_i)} \Big|_0^T - \dots \\ &\quad - \frac{R \sin[(\Omega_i - \rho_i)t + (\theta_i - \zeta_i)]}{T(\Omega_i - \rho_i)} \Big|_{(N-1)T}^{NT}, \end{aligned} \tag{18}$$

and since

$$\begin{aligned} & \frac{\sin[(\Omega_i - \rho_i)t + (\theta_i - \zeta_i)]}{\Omega_i - \rho_i} \Big|_{nT}^{(n+1)T} \\ &= \frac{\sin[(\Omega_i - \rho_i)(n+1)T + (\theta_i - \zeta_i)]}{\Omega_i - \rho_i} \\ &\quad - \frac{\sin[(\Omega_i - \rho_i)nT + (\theta_i - \zeta_i)]}{\Omega_i - \rho_i} \\ &= \frac{\sin \Delta \phi[(n+1)T] - \sin \Delta \phi[nT]}{\phi[(n+1)T] - \phi[nT]}. \end{aligned} \tag{19}$$

Hence,

$$d_{min,N}^2 = R \left[N - \frac{1}{T} \sum_{n=0}^{N-1} \frac{\sin(\Delta\phi[(n+1)T]) - \sin(\Delta\phi[nT])}{\phi[(n+1)T] - \phi[nT]} \right]. \quad (20)$$

In practical we can simplify the formulation as the following:

Let the difference of two phases at $(n+1)T$ be $\Omega_\alpha - \Omega_\beta$, and that of the two phases at nT be $\theta_\alpha - \theta_\beta$. where,

$$d_n^2(\alpha, \beta) = 1 - \int_0^1 \cos[(\Omega_{\alpha,n+1} - \Omega_{\beta,n+1})t + (\theta_{\alpha,n} - \theta_{\beta,n})] dt.$$

Hence, the normalized squared free Euclidean distance is

$$NSFED = R \sum_N d_n^2(\alpha, \beta),$$

where R is the rate of a code.

APPENDIX B

This program is to calculate the Euclidean distance between CPM signals.

This program is of an example of encoder $[1+D, 1, 1+D]$ with rate $2/3$ and constraint length 1.

```

#include "stdio.h"
#define P_n 8
#define Y_0 out_0=a1^b1
#define Y_1 out_1=a1
#define Y_2 out_2=a2^b2
#define prt_0 printf("\t%d",out_0)
#define prt_1 printf("\t%d",out_1)
#define prt_2 printf("\t%d",out_2)
main()
{
    static int e_1[4]={0,0,1,1};
    static int e_2[4]={0,1,0,1};

    int i,B,C,D,a1,a2,b1,b2,c1,c2,d1,d2;
    int out_0,out_1,out_2;

    B=0;
    for(i=0 ; i<4 ; i++){
        b1=B;
        a1=e_1[i];
        Y_0;
        Y_1;

        b2=e_1[i];
        a2=e_2[i];

        Y_2;

        prt_0;
        prt_1;
        prt_2;
        putchar('\n');
    }
    B=1;
    for(i=4 ;i<8 ;i++){

        b1=B;
        a1=e_1[i-4];
        Y_0;
        Y_1;
        b2=e_1[i-4];
        a2=e_2[i-4];
        Y_2;

        prt_0;
        prt_1;
        prt_2;
        putchar('\n');
    }
}

```

0
1
1
1
1
1
0
0

0
1
1
0
0
1
1
1

1
1
0
0
1
1
1
0

This program is to calculate the Euclidean distance between CPM signals with rate 2/3 and constraint length 1, encoder [1+D,1,1+D].

```

#include "stdio.h"
#include <math.h>
#define P_n 8
#define PI 3.1415926

float state(pt1,pt2,pt3)
int pt1,pt2,pt3;
{
    int value;
    float r;

    value=4*pt1+2*pt2+pt3;
    switch(value)
    {
        case 0:
            r=0.;
            break;
        case 1:
            r=PI/4.;
            break;
        case 2:
            r=PI;
            break;
        case 3:
            r=3.*PI/4.;
            break;
        case 4:
            r=1.75*PI;
            break;
        case 5:
            r=2.*PI;
            break;
        case 6:
            r=1.25*PI;
            break;
        case 7:
            r=PI;
            break;
    }
    return(r);
}

main()
{
    int dst[P_n],dnd[P_n],drd[P_n];
    int i,j,k,j1,k1,j2,k2,j3,k3,j4,k4,N;
    int A,B,A1,B1,A2,B2,A3,B3,A4,B4,A5,B5;
    float dmin,ED1,ED2,ED3,x,d1,d2,d3;
    float phase_of_j,phase_of_k;
    float V0,V1,V2,V3;
    float y1,y2,y3,y4,pj1,pj2,pj3,pk1,pk2,pk3;

    for(i=0; i<P_n; i++){

```

```

    scanl (" %d %d %d ", &dst[i], &dnd[i], &drd[i]);
}
dmin=1000.;

for(i=0;i<P_n;i++){
  for(j=0;j<3;j++){
    for(k=j+1;k<4;k++){

      A=j+i*4;
      B=k+i*4;

      if(A>=P_n)
        A=A%P_n;
      if(B>=P_n)
        B=B%P_n;

      phase_of_j=state(dst[i],dnd[i],drd[i]);
      phase_of_k=state(dst[i],dnd[i],drd[i]);
      pj1=state(dnd[i],drd[i],dst[A]);
      pk1=state(dnd[i],drd[i],dst[B]);

      if(phase_of_j>pj1) phase_of_j=phase_of_j-2.*PI;
      if(phase_of_k>pk1) phase_of_k=phase_of_k-2.*PI;

      y1=pj1-phase_of_j;
      y2=pk1-phase_of_k;
      y3=0.;
      y4=pj1-pk1;
      x=y1-y2;
      if(fabs(x) < 0.5)
        d1=0.;
      else
        d1=1.-(sin(y4))/(x);

      pj2=state(drd[i],dst[A],dnd[A]);
      pk2=state(drd[i],dst[B],dnd[B]);

      if(pj1>pj2) pj1=pj1-2.*PI;
      if(pk1>pk2) pk1=pk1-2.*PI;
      y1=pj2-pj1;
      y2=pk2-pk1;
      y3=y4;
      y4=pj2-pk2;
      x=y1-y2;
      if(fabs(x) < 0.5)
        d2=1.- cos(y3);
      else
        d2=1.-(sin(y4)- sin(y3))/(x);

      pj3=state(dst[A],dnd[A],drd[A]);
      pk3=state(dst[B],dnd[B],drd[B]);

      if(pj2>pj3) pj2=pj2-2.*PI;
      if(pk2>pk3) pk2=pk2-2.*PI;
      y1=pj3-pj2;
      y2=pk3-pk2;
      y3=y4;
      y4=pj3-pk3;
      x=y1-y2;
      if(fabs(x) < 0.5)
        d3=1.- cos(y3);
      else

```

```
d3=1.-(sin(y4)- sin(y3))/(x);
```

```
V0=d1+d2+d3;
```

```
for(jl=0;jl<4;jl++){
  for(kl=0;kl<4;kl++){

    A1=jl+A*4;
    B1=kl+B*4;

    if(A1>=P_n)
      A1=A1%P_n;
    if(B1>=P_n)
      B1=B1%P_n;

    pj1=state(dnd[A],drd[A],dst[A1]);
    pk1=state(dnd[B],drd[B],dst[B1]);

    phase_of_j=state(dst[A],dnd[A],drd[A]);
    phase_of_k=state(dst[B],dnd[B],drd[B]);

    if(phase_of_j>pj1) phase_of_j=phase_of_j-2.*PI;
    if(phase_of_k>pk1) phase_of_k=phase_of_k-2.*PI;
    y1=pj1-phase_of_j;
    y2=pk1-phase_of_k;
    y4=pj1-pk1;
    y3=phase_of_j-phase_of_k;
    x=y1-y2;
    if(fabs(x) < 0.5)
      d1=1.- cos(y3);
    else
      d1=1.-(sin(y4)- sin(y3))/(x);

    pj2=state(drd[A],dst[A1],dnd[A1]);
    pk2=state(drd[B],dst[B1],dnd[B1]);

    if(pj1>pj2) pj1=pj1-2.*PI;
    if(pk1>pk2) pk1=pk1-2.*PI;
    y1=pj2-pj1;
    y2=pk2-pk1;
    y3=y4;
    y4=pj2-pk2;
    x=y1-y2;
    if(fabs(x) < 0.5)
      d2=1.- cos(y3);
    else
      d2=1.-(sin(y4)- sin(y3))/(x);

    pj3=state(dst[A1],dnd[A1],drd[A1]);
    pk3=state(dst[B1],dnd[B1],drd[B1]);

    if(pj2>pj3) pj2=pj2-2.*PI;
    if(pk2>pk3) pk2=pk2-2.*PI;
    y1=pj3-pj2;
    y2=pk3-pk2;
    y3=y4;
    y4=pj3-pk3;
    x=y1-y2;
    if(fabs(x) < 0.5)
      d3=1.- cos(y3);
```

```

else
    d3=1.-(sin(y4)- sin(y3))/(x);

V1=d1+d2+d3;
if(i==A1&&i==B1){
    ED1=V0+V1;
    printf("ED1=%f a=%d a1=%d \n",ED1,A,A1 );
    printf("\t      b=%d b1=%d \n", B,B1);
    if(dmin > ED1)
        dmin=ED1;
    ED1=1000.;
}
else if(i!=A1&&i!=B1&&A1==B1)
    ED1=1000.;
else
    ED1=V0+V1;
ED1=ED1;
for(j2=0; j2<4; j2++){
    for(k2=0 ; k2<4; k2++){

        A2=j2+A1*4;
        B2=k2+B1*4;

        if(A2>=P_n)
            A2=A2%P_n;
        if(B2>=P_n)
            B2=B2%P_n;

        pj1=state(dnd[A1],drd[A1],dst[A2]);
        pk1=state(dnd[B1],drd[B1],dst[B2]);

        phase_of_j=state(dst[A1],dnd[A1],drd[A1]);
        phase_of_k=state(dst[B1],dnd[B1],drd[B1]);

        if(phase_of_j>pj1) phase_of_j=phase_of_j-2.*PI;
        if(phase_of_k>pk1) phase_of_k=phase_of_k-2.*PI;
        y1=pj1-phase_of_j;
        y2=pk1-phase_of_k;
        y3=phase_of_j-phase_of_k;
        y4=pj1-pk1;
        x=y1-y2;
        if(fabs(x) < 0.5)
            d1=1.- cos(y3);
        else
            d1=1.-(sin(y4)- sin(y3))/(x);

        pj2=state(drd[A1],dst[A2],dnd[A2]);
        pk2=state(drd[B1],dst[B2],dnd[B2]);

        if(pj1>pj2) pj1=pj1-2.*PI;
        if(pk1>pk2) pk1=pk1-2.*PI;
        y1=pj2-pj1;
        y2=pk2-pk1;
        y3=y4;
        y4=pj2-pk2;
        x=y1-y2;
        if(fabs(x) < 0.5)
            d2=1. - cos(y3);
        else
            d2=1.-(sin(y4)- sin(y3))/(x);

        pj3=state(dst[A2],dnd[A2],drd[A2]);
        pk3=state(dst[B2],dnd[B2],drd[B2]);
    }
}

```

```

if(pj2>pj3) pj2=pj2-2.*PI;
if(pk2>pk3) pk2=pk2-2.*PI;
    y1=pj3-pj2;
    y2=pk3-pk2;
    y3=y4;
    y4=pj3-pk3;
    x=y1-y2;

if(fabs(x) < 0.5)
    d3=1.- cos(y3);
else
    d3=1.-(sin(y4)- sin(y3))/(x);

V2=d1+d2+d3;

if(i==A2&&i==B2){
    ED2=ED1+V2;
    printf("ED2=%.3f \n",ED2);
    if(ED2 < 3.5){
        puts(" Forget this code");
        exit();
    }
    if(dmin > ED2)
        dmin=ED2;
    ED2=1000.;
}
else if(i!=A2&&A2==B2&&i !=B2)
    ED2=1000.;
else
    ED2=ED1+V2;
ED2=ED2;
for(j3=0; j3<4; j3++){
    for(k3=0; k3<4; k3++){

        A3=j3+A2*4;
        B3=k3+B2*4;
        if(A3>=P_n) A3=A3%P_n;
        if(B3>=P_n) B3=B3%P_n;

/*
        if(i==A3&&i==B3){
            /*
            pj1=state(dnd[A2],drd[A2],dst[A3]);
            pk1=state(dnd[B2],drd[B2],dst[B3]);

            phase_of_j=state(dst[A2],dnd[A2],drd[A2]);
            phase_of_k=state(dst[B2],dnd[B2],drd[B2]);

            if(phase_of_j>pj1) phase_of_j=phase_of_j-2.*PI;
            if(phase_of_k>pk1) phase_of_k=phase_of_k-2.*PI;
            y1=pj1-phase_of_j;
            y2=pk1-phase_of_k;
            y3=phase_of_j-phase_of_k;
            y4=pj1-pk1;
            x=y1-y2;
            if(fabs(x) <0.5)
                d1=1.- cos(y3);
            else
                d1=1.-(sin(y4)- sin(y3))/(x);

            pj2=state(drd[A2],dst[A3],dnd[A3]);
            pk2=state(drd[B2],dst[B3],dnd[B3]);

            if(pj1>pj2) pj1=pj1-2.*PI;
            if(pk1>pk2) pk1=pk1-2.*PI;

```

```

        y1=pj2-pj1;
        y2=pk2-pk1;
        y3=y4;
        y4=pj2-pk2;
        x=y1-y2;
    if (fabs(x) <0.5)
        d2=1.- cos(y3);
    else
        d2=1.-(sin(y4)- sin(y3))/(x);

    pj3=state(dst[A3],dnd[A3],drd[A3]);
    pk3=state(dst[B3],dnd[B3],drd[B3]);

    if(pj2>pj3) pj2=pj2-2.*PI;
    if(pk2>pk3) pk2=pk2-2.*PI;
        y1=pj3-pj2;
        y2=pk3-pk2;
        y3=y4;
        y4=pj3-pk3;
        x=y1-y2;

    if (fabs(x) <0.5)
        d3=1.- cos(y3);
    else
        d3=1.-(sin(y4)- sin(y3))/(x);

    V3=d1+d2+d3;

    if(i==A3&&i==B3){

        ED3=ED2+V3;
        printf("ED3=%.3f \n",ED3);

        if(ED3 < 3.5){

            exit();

        }

        {
            dmin=ED3;
            printf("dmin=%.3f\n", 2*dmin/3);
            ED3=1000.0;
        }
        else if(i!=A3&&A3==B3&&i!=B3)
            ED3=1000.;
        else
            ED3=ED2+V3;

        ED3=ED3;

    for(j4=0; j4<4; j4++){
        for(k4=0; k4<4; k4++){

            A4=j4+A3*4;
            B4=k4+B3*4;

            if(A4>=P_n) A4=A4%P_n;
            if(B4>=P_n) B4=B4%P_n;

            if(i==A4&&i==B4){

                pj1=state(dnd[A3],drd[A3],dst[A4]);
                pk1=state(dnd[B3],drd[B3],dst[B4]);

                phase_of_j=state(dst[A3],dnd[A3],drd[A3]);
                phase_of_k=state(dst[B3],dnd[B3],drd[B3]);

```



```

if(phase_of_j>pj1) phase_of_j=phase_of_j-2.*PI;
if(phase_of_k>pk1) phase_of_k=phase_of_k-2.*PI;

y1=pj1-phase_of_j;
y2=pk1-phase_of_k;
y3=phase_of_j-phase_of_k;
y4=pj1-pk1;
x=y1-y2;

if(fabs(x) < 0.5)
    d1=1. - cos(y3);
else
    d1=1. -(sin(y4) - sin(y3))/(x);

pj2=state(drd[A3],dst[A4],dnd[A4]);
pk2=state(drd[B3],dst[B4],dnd[B4]);

if(pj1>pj2) pj1=pj1-2.*PI;
if(pk1>pk2) pk1=pk1-2.*PI;

y1=pj2-pj1;
y2=pk2-pk1;
y3=y4;
y4=pj2-pk2;
x=y1-y2;

if(fabs(x) < 0.5)
    d2=1. - cos(y3);
else
    d2=1. - (sin(y4) - sin(y3))/(x);

pj3=state(dst[A4],dnd[A4],drd[A4]);
pk3=state(dst[B4],dnd[B4],drd[B4]);

if(pj2>pj3) pj2=pj2-2.*PI;
if(pk2>pk3) pk2=pk2-2.*PI;

y1=pj3-pj2;
y2=pk3-pk2;
y3=y4;
y4=pj3-pk3;
x=y1-y2;

if(fabs(x) < 0.5)
    d3=1. - cos(y3);
else
    d3=1. - (sin(y4) - sin(y3))/(x);

V4=d1+d2+d3;

ED4=ED3+V4;

puts("*****");
printf("ED4=%.3f\n",ED4);

if(ED4 < 3.5){ /* =2.3*3/2 */
    puts("Forget this code");
    exit();
}
if(dmin > ED4)
    dmin=ED4;
puts("*****");
}
else
    ED3=ED3;

```

```
}  
}  
}  
}  
}  
}  
}  
}  
}  
}  
}
```

```
puts(" We got the result");  
printf("dmin=%.3f \n",2*dmin/3.);  
puts("This code is greater than the previous");
```

```
}
```

REFERENCES

- [1] F. Morales-Moreno, W. Holubowics, and S. Pasupathy, "Optimization of Trellis Coded $(1 + D)^2 - FM$ via Matched Codes" Technical Report no. CCSPR-001-88
The Center for Communication and Signal Processing Research
Department of Electrical Engineering
New Jersey Institute of Technology , Newark , NJ , USA
submitted to the *IEEE Trans. Comm.* , August 1988.
- [2] F. Morales-Moreno, W. Holubowics, and S. Pasupathy, "Performance of Trellis Encoded Tamed Frequency Modulation", *IEEE* , 38th Veh. Tech. Conf. pp. 12-17 1988.
- [3] F. Jager and C.B. Dekker, "Tamed Frequency Modulation, a novel method to achieve spectrum economy in digital transmission," *IEEE Trans. Comm.*, vol. COM-26, pp. 534-542, May 1978.
- [4] K.S. Chung and L.E. Zegers, "Generalized Tamed Frequency Modulation," *Philips J. Res.*, vol. 37, no. 4, pp. 165-177, 1982.
- [5] T. Aulin and C.E.W. Sundberg, "Continuous Phase Modulation - Part I : Full response signaling," *IEEE Trans. Comm.*, vol. COM-29, pp. 196-209, March 1981.
- [6] J.B. Cain, G.C. Clark, Jr., and J.M. Geist, "Punctured Convolutional Codes of

- Rate $(n - 1)/n$ and Simplified Maximum Likelihood Decoding," *IEEE Trans. Inform.* vol. IT-25, No. 1, pp. 97-100 Jan. 1979.
- [7] G.D. Forney, Jr., " Structural Analysis of Convolutional Codes via Dual Codes," *IEEE Trans. Infor. Theory*, vol. IT-19, pp. 512-518, July 1973.
- [8] G. Ungerboeck, "Channel Coding with Multilevel- Phase Signals," *IEEE Trans. Infor.Theory*, vol. IT-28, pp. 55-67, January 1982.
- [9] F. Morales-Moreno and S. Pasupathy, " Optimal Trellis Codes and Generalized MSK signals," *Proc.Global Telecommunications Conference*, Atlanta, GA, pp. 22.2.1-22.2.5, Nov. 1984.
- [10] S. Pasupathy," Nyquist's third Criterion," *Proceedings of the IEEE*, vol. 62, pp. 860 – 861, June 1974.
- [11] S.G. Willson and M.G.Mulligan, " An improved algorithm for evaluating trellis phase codes," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 846 – 850, Nov. 1984.
- [12] J.G. Proakis, *Digital Communications*, McGraw-Hill Book Co., 1983.
- [13] A.J. Viterbi and J.K.Omura,"Principle of Digital Communication and Coding," McGraw-Hill Book Co., 1979.
- [14] T.Aulin," Viterbi Detection of Continuous Phase Modulated Signals," in *NTC'80 Confer. Rec.*, pp. 14.2.1-14.2.7, Houston, Dec. 1980.
- [15] A.J.Viterbi," Convolutional Codes and their Performance in Communication Systems," *IEEE Trans. Commun. Technol.*, vol. COM-19,

Oct. 1971. pp. 751-771

- [16] G. Ungerboeck, " Trellis- Coded Modulation with Redundant Signal Sets ; Part - I," *IEEE Commn. Magazine*, vol. 25. pp. 5-11, Feb. 1987.
- [17] C.E. Sundberg," Continuous Phase Modulation," *IEEE Comm. Magazine*, vol. 24, No. 4, April 1986.
- [18] T. Aulin, C.E. Sundberg, and A. Svensson," Viterbi Detectors with Reduced Complexity for Partial Response Continuous Phase Modulation," *IEEE Globecom.*, pp. A7.6.1-9, 1981.
- [19] J. A. Heller, I. M. Jacobs," Viterbi Decoding for Satellite and Space Communication," *IEEE Trans. Comm. Technol*, vol. COM-19, No. 5, Oct. 1971.
- [20] A. J. Viterbi, " Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, " *IEEE Trans. Inform. Theory*, vol. IT - 13, Apr. 1967, pp. 260 - 269.
- [21] J.K. Omura, "On the Viterbi decoding algorithm," *IEEE Trans. Inform. Theory* (Corresp.), vol. IT-15, Jan. 1969, pp. 177-179.

LIST OF FIGURES

Fig. 1. TFM Modulator M

Fig. 2. Combined coding and modulation communication system

Fig. 3. The finite state machine of TFM modulation. The branches are labeled by the input bits to the TFM modulator

Fig. 4. Combined Coding and Precoded TFM modulation communications system model par

Fig. 5. The finite state machine of precoded TFM modulation. The branches are labeled by the input bits to the precoded TFM modulator

Fig. 6. The trellis of Coded modulation for encoder $[1 + D, 1]$

a. Convolution encoder

b. Trellis diagram of convolutional encoder

Fig. 7. Trellis of combined coding and TFM modulation

for encoder $[1 + D, 1]$ with $R = 1/2, v = 1$

Fig. 8. Trellis diagram for a $R = 2/3, v = 2$ encoder in eq.(5.2)

Fig. 9. Trellis diagram of a $R = 2/3, v = 2$ code produced by

periodically deleting bits from a $R = 1/2, v = 2$ code

Fig. 10. Trellis diagram of the punctured trellis encoder $[1 + D, 1, 1 + D]$

with $R = 2/3, v = 1$

Fig. 11. Trellis diagram of the encoder $[1 + D, 1, 1 + D]$ combined coding and TFM

modulation produced by periodically deleting bits from $R = 1/2, v = 1$ encoder
 $[1 + D, 1]$

Fig. 12. Trellis diagram (original) of combined coding and modulation of encoder
 $[1 + D, 1, 1 + D]$ with $R = 2/3, v = 1$.

LIST OF TABLES

- [1] **Table I.** Best rate - $2/3$ punctured trellis codes
- [2] **Table II.** Coding gains (dB) of best rate - $2/3$ punctured trellis codes and best rate - $1/2$ non-punctured codes
- [3] **Table III.** Generator matrix and encoder matrix of $2/3$ punctured trellis codes
- [4] **Table IV.** Best rate - $2/3$ punctured trellis codes and best rate - $1/2$ non-punctured codes

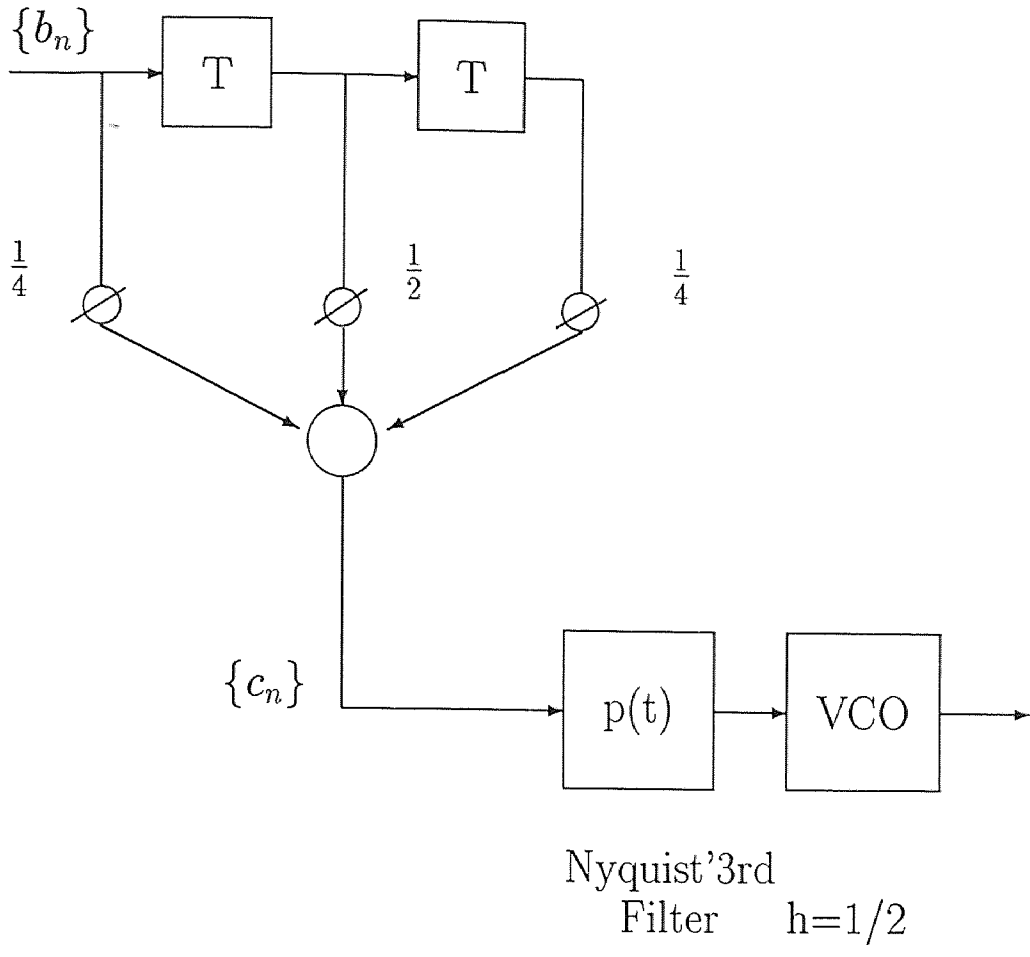


Fig. 1. TFM modulator M

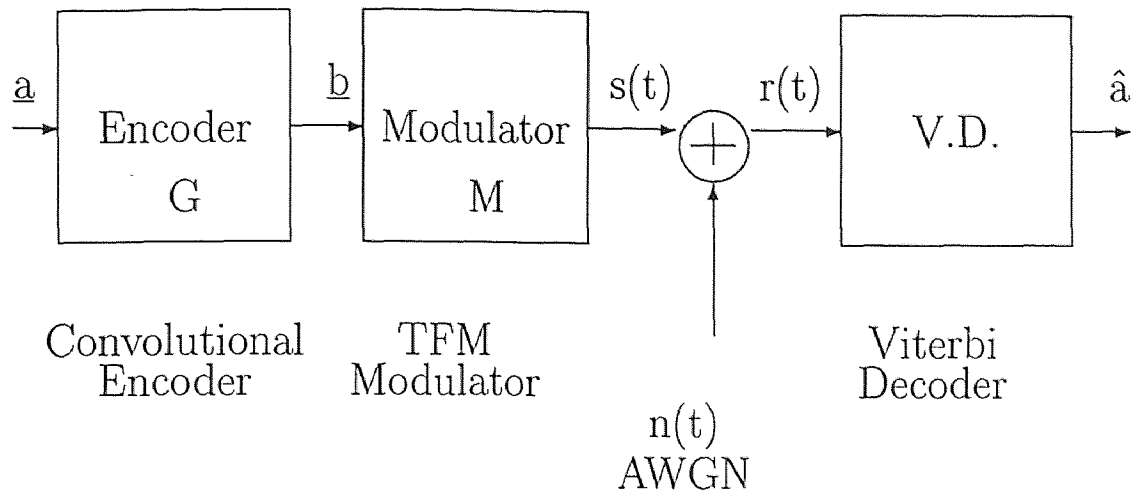


Fig. 2. Combined coding and modulation communication system.

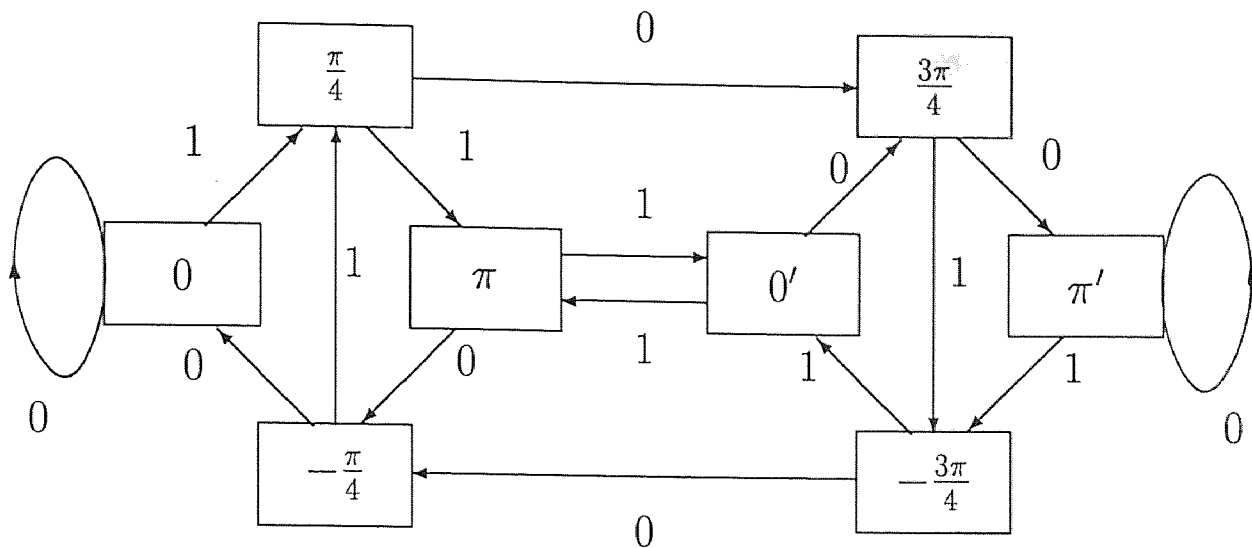


Fig. 3. The finite state machine of TFM modulation. The branches are labeled by the input bits to the TFM modulator.

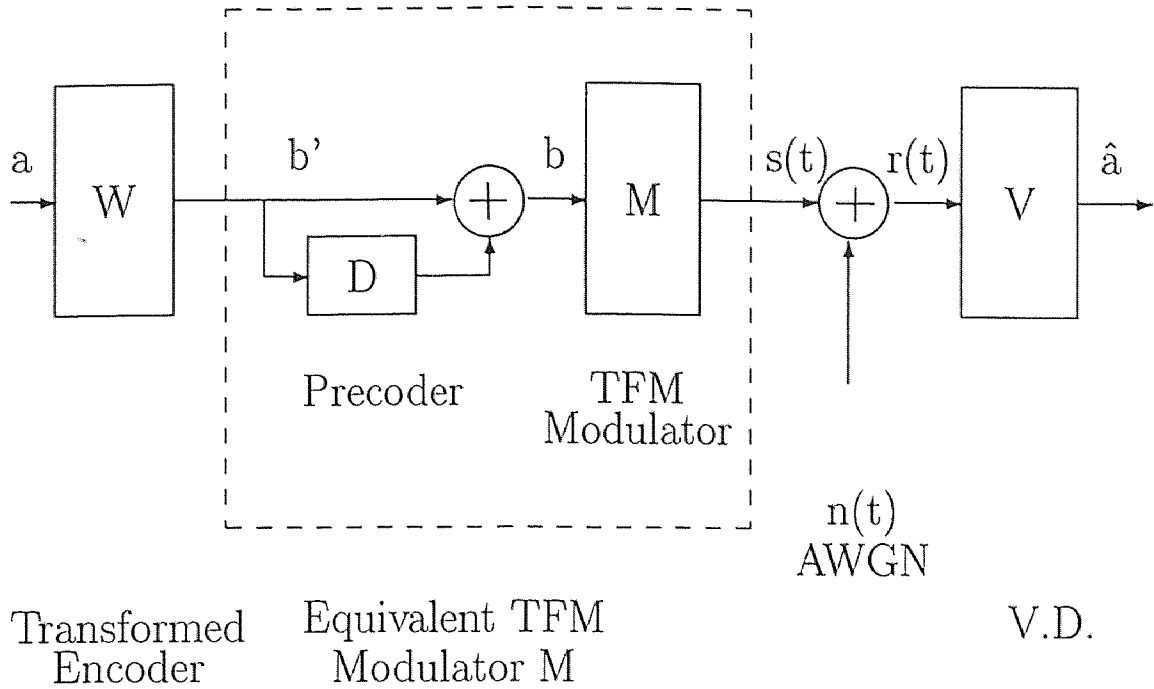


Fig. 4. Combined coding and precoded TFM modulation communication system model.

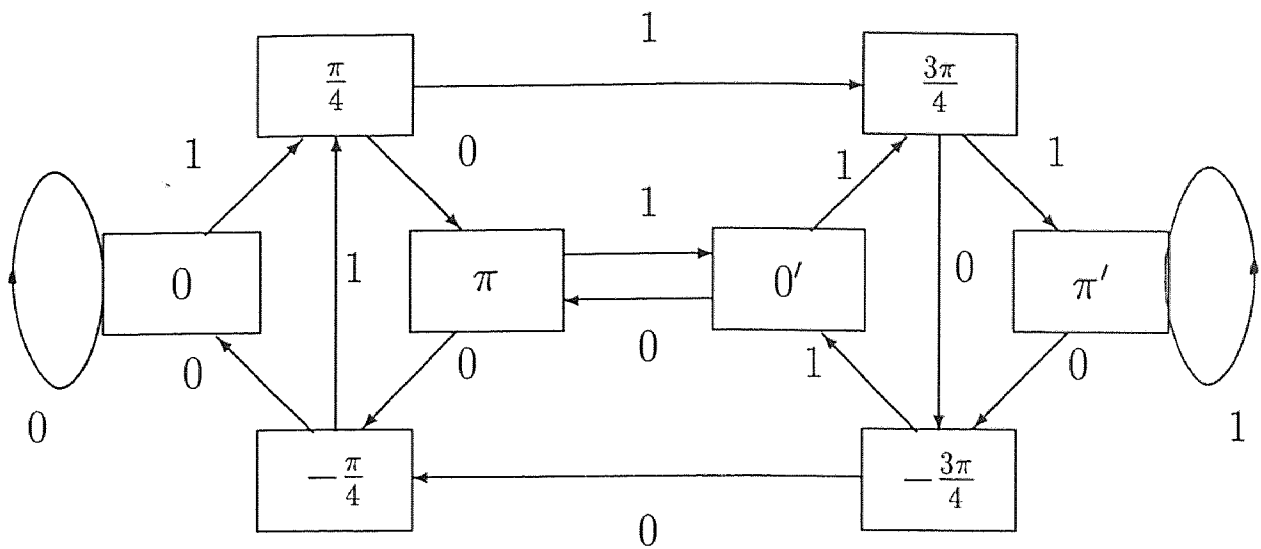


Fig. 5. The finite state machine of precoded TFM modulation. The branches are labeled by the input bits to the precoded TFM modulator.

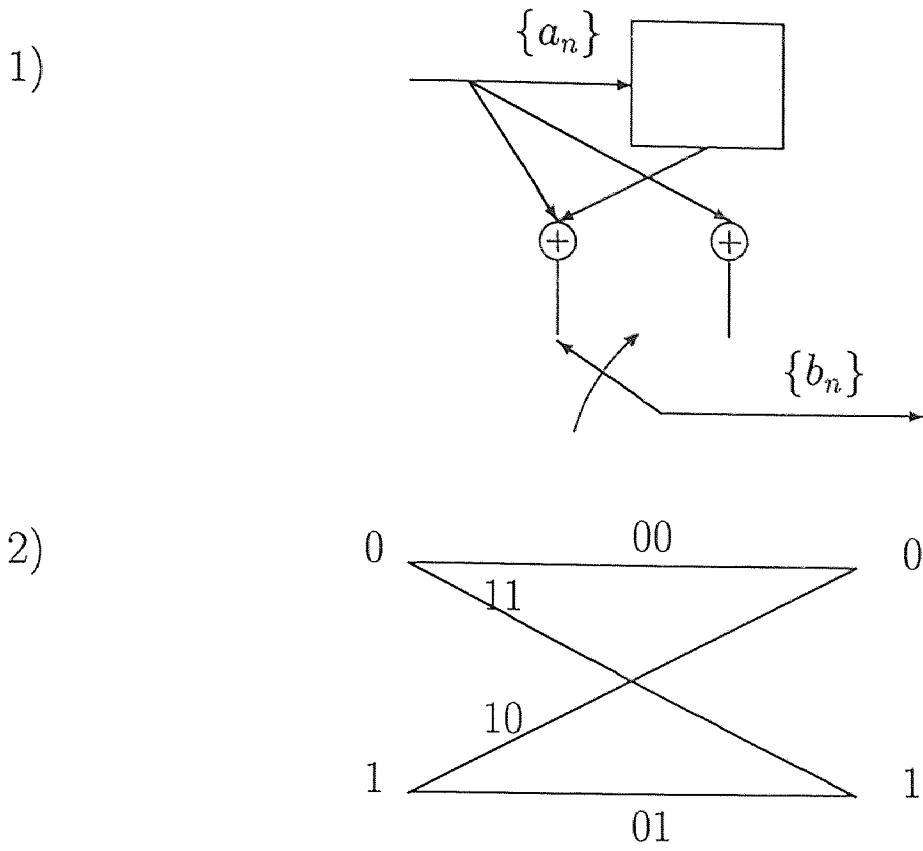


Fig. 6. Trellis coded modulation for the encoder $[1+D, 1]$

1. Convolutional encoder,
2. Trellis diagram of convolutional encoder.

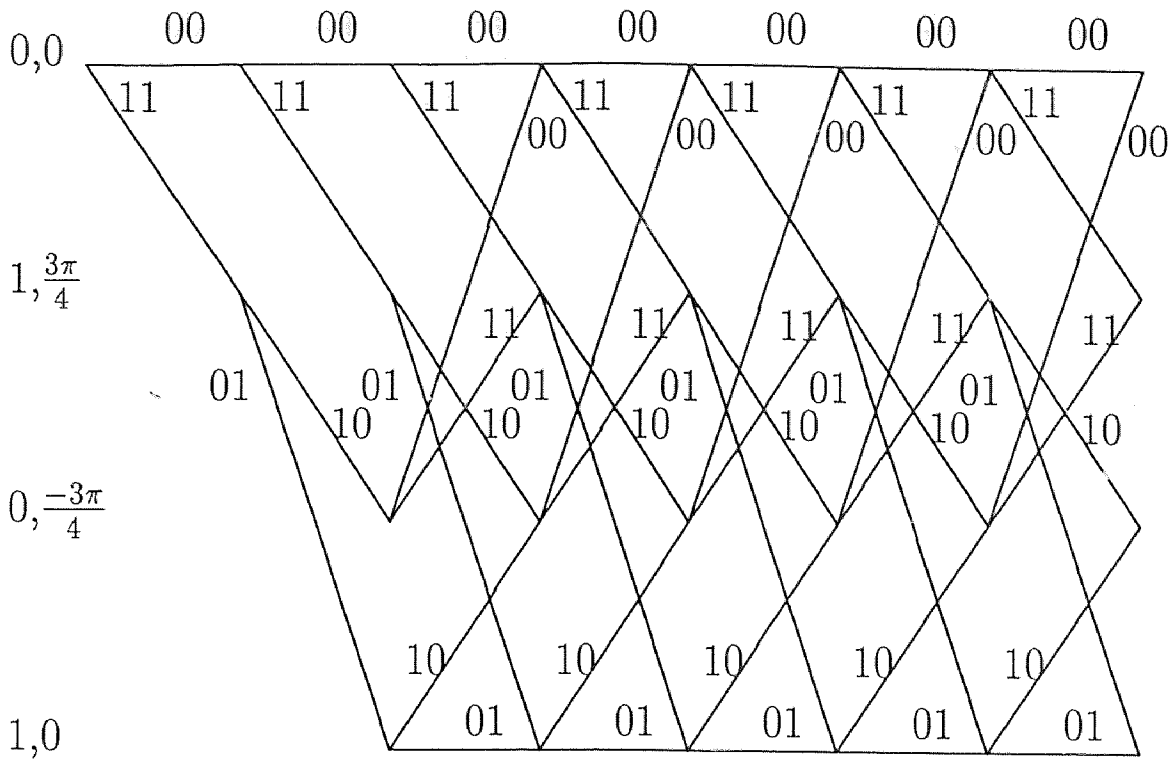


Fig. 7. Trellis of combined coding and TFM modulation for the encoder $[1 + D, 1]$ with $R = 1/2, v = 1$.

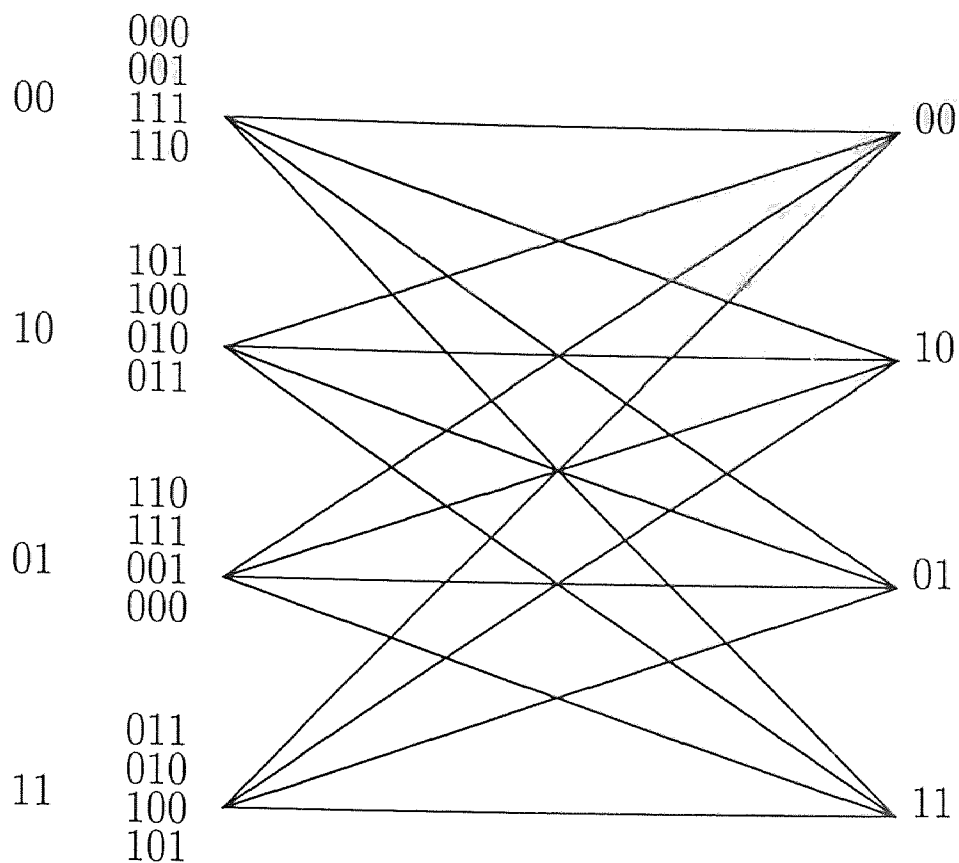


Fig. 8. Trellis diagram for the encoder in Eq. (5.2) with $R = 2/3, v = 1$.

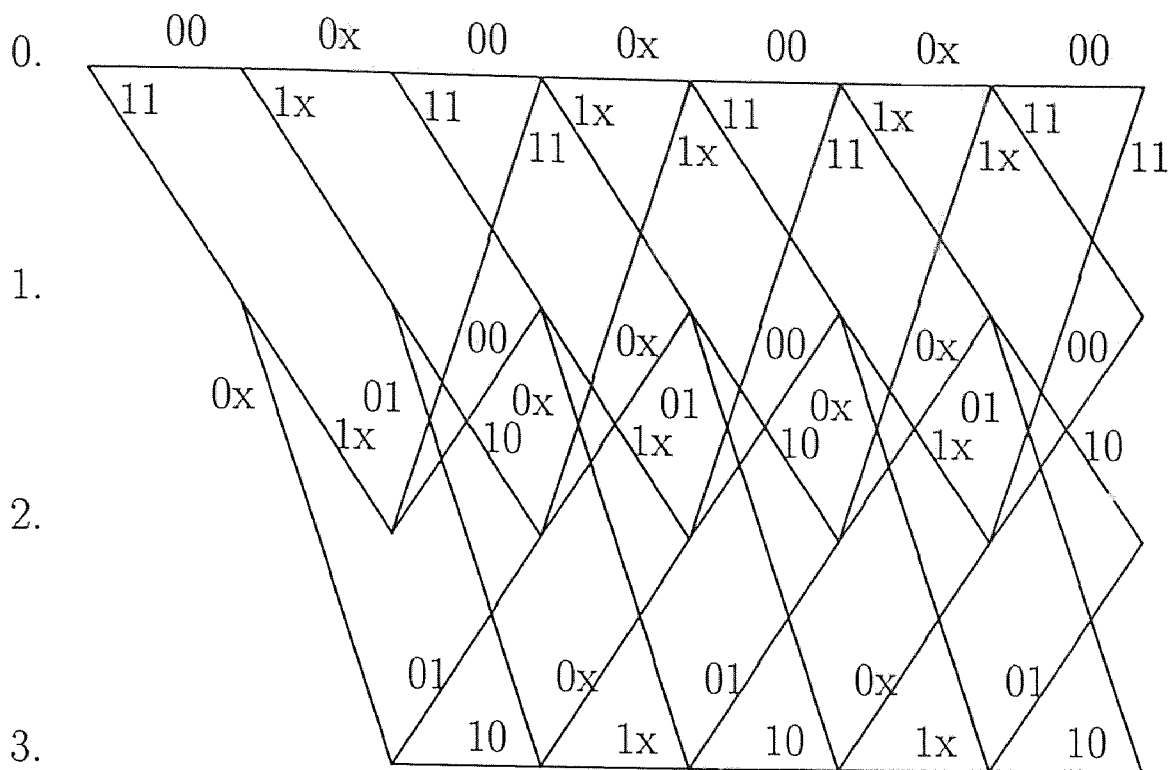


Fig. 9. Trellis diagram of $R = 2/3, v = 2$ code produced by periodically deleting bits from $R = 1/2, v = 2$ code of Fig.8. x is a deleted bit.

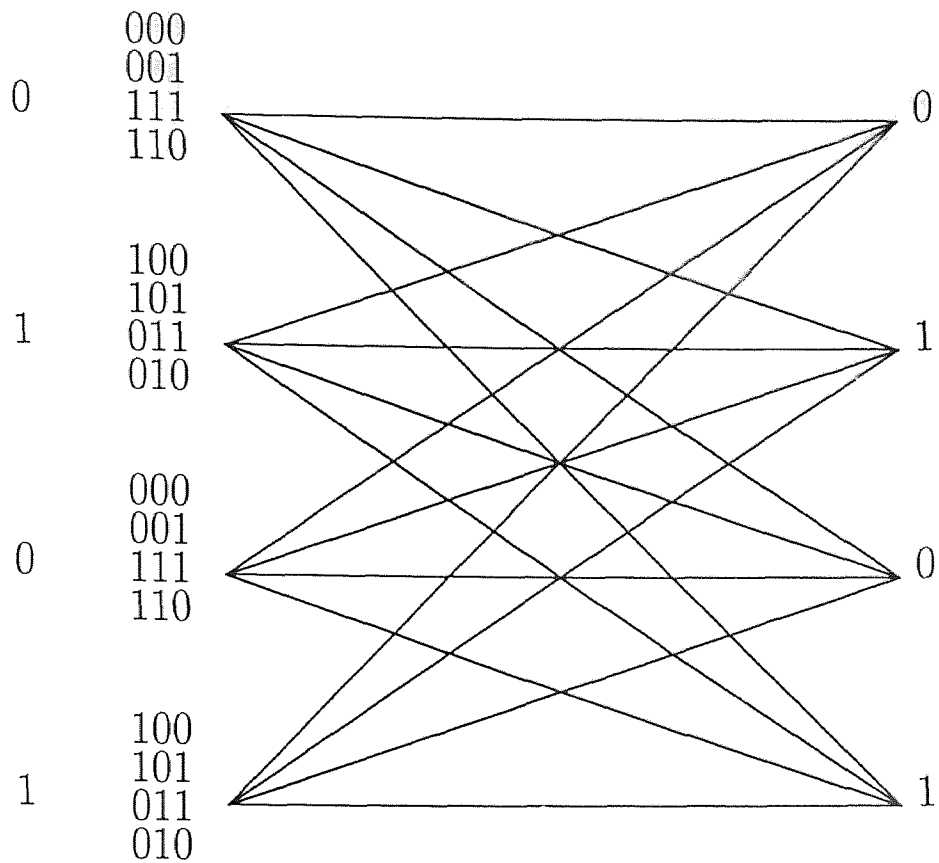


Fig. 10. Trellis diagram of the punctured trellis encoder $[1 + D, 1, 1 + D]$ with $R = 2/3, v = 1$.

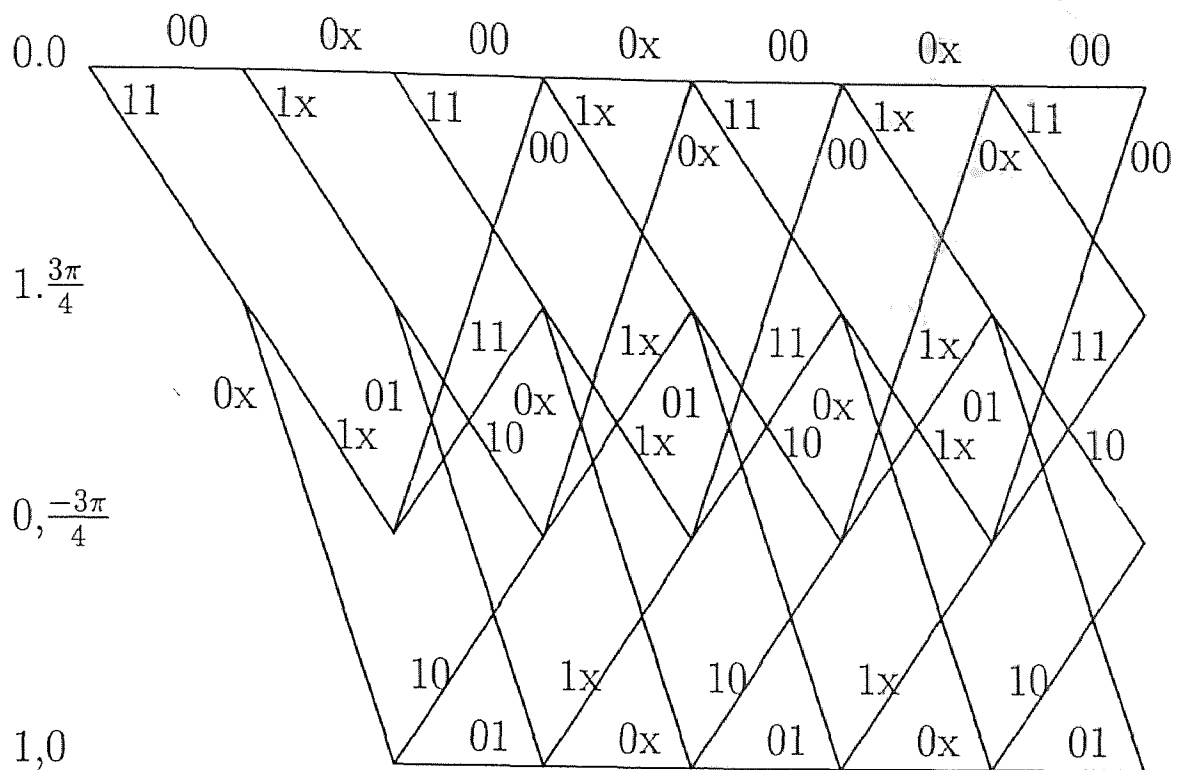


Fig. 11. Trellis diagram of $[1 + D, 1, 1 + D]$ combined coding and TFM modulation produced by periodically deleting bits from the $R = 1/2, v = 1$ code $[1 + D, 1]$.

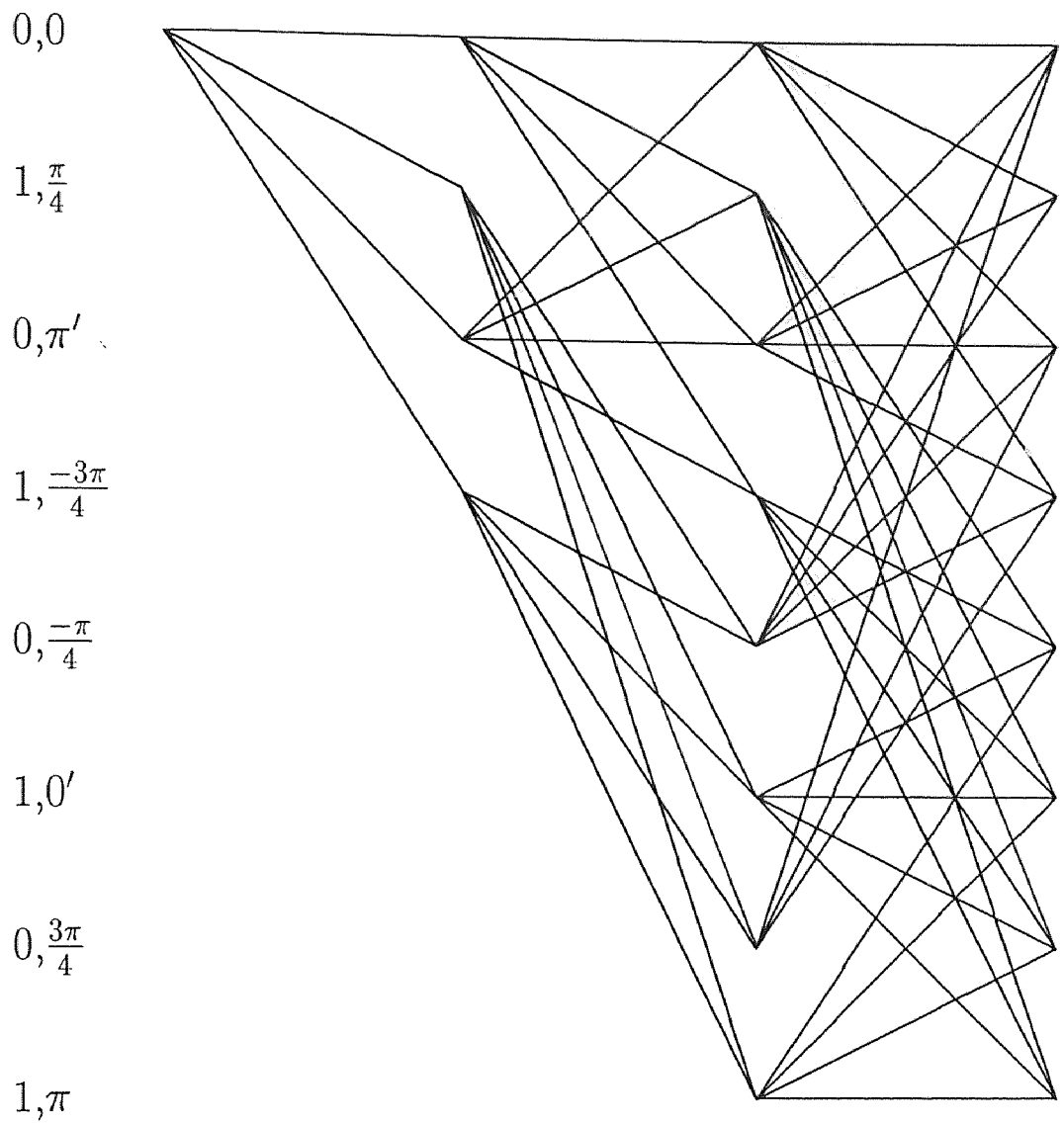


Fig. 12. Trellis diagram (original) of combined coding and modulation of encoder $[1 + D, 1, 1 + D]$ with $R = 2/3, v = 1$.

Table I. Best rate - 2/3 punctured trellis codes

S_V	Generators-2/3 in binary form	NSFED	Coding Gain
4	11 10 11	2.733	2.74
8	101 110 101	3.938	4.33
16	1111 1010 1111	4.864	5.25
32	10111 10110 10111	5.441	5.73

Table II. Coding gains (dB) of best rate - 2/3
punctured trellis codes and
best rate - 2/3 non-punctured codes

S_V	New Results	Previously reported work	Best decoder
	Best rate-2/3 Punctured Codes	Best rate-2/3 non-punctured Codes*	
4	2.74	2.00	2.74 (8 states)
8	4.33	2.74	4.33 (16 states)
16	5.25	4.33	5.65 (32 states)
32	5.73	5.65	6.15 (64 states)

* Codes obtained in[1]

Table III. Generator matrix and encoder matrix
of rate - 2/3 punctured trellis codes

S_V	Generators in binary form	Encoder in binary form
4	11 10 11	10 10 10 01 00 10
8	101 110 101	11 10 00 00 01 11
16	1111 1010 1111	110 110 110 011 000 110
32	10111 10110 10111	111 100 100 001 011 111

Table IV. Best rate - 2/3 punctured trellis codes
and rate - 1/2 matched codes

S_V	Generators-1/2 in binary form	Generators-2/3 in binary form
4	11 10	11 10 11
8	101 110	101 110 101
16	1111 1010	1111 1010 1111
32	10111 10110	10111 10110 10111