General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Technical Report

AN IMPROVED ALGORITHM FOR EVALUATING TRELLIS PHASE CODES

Submitted to:

Communications Technical Consultation Office
National Aeronautics and Space Administration
Lewis Research Center
21000 Brookpark Road
Cleveland, OH 44135

Attention: E. F. Miller
W. A. Whyte

Submitted by:

M. G. Mulligan
Graduate Research Assistant

S. G. Wilson
Associate Professor

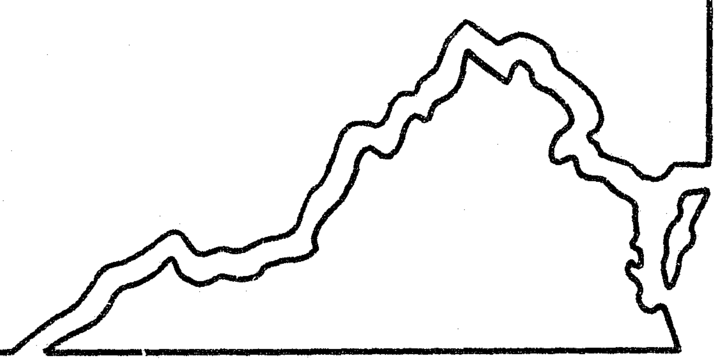
(NASA-CR-169354) AN IMPROVED ALGORITHM FOR EVALUATING TRELLIS PHASE CODES (Virginia Univ.) 18 p HC A02/MF A01 CSCL 20N

N83-16600

Unclas
G3/32 35302

Report No. UVA/528200/EE82/104

September 1982



COMMUNICATIONS SYSTEMS LABORATORY
DEPARTMENT OF ELECTRICAL ENGINEERING
SCHOOL OF ENGINEERING AND APPLIED SCIENCES
UNIVERSITY OF VIRGINIA

AN IMPROVED ALGORITHM FOR EVALUATING TRELLIS PHASE CODES

Michael G. Mulligan
Stephen G. Wilson

Communication Systems Laboratory
Department of Electrical Engineering
University of Virginia
Thornton Hall
Charlottesville, VA 22901
(804)924-6091

Abstract

A method is described for evaluating the minimum distance parameters of trellis-phase codes, including CPFSK, partial-response FM and, more importantly, coded CPM (continuous-phase modulation) schemes. The algorithm provides dramatically faster execution times and lesser memory requirements than previous algorithms. Results of sample calculations and timing comparisons are included.

AN IMPROVED ALGORITHM FOR EVALUATING TRELLIS PHASE CODES

I. INTRODUCTION

Trellis phase codes are digital phase-modulated (constant-envelope) signals whose phase trajectories undergo smooth phase transitions which can have memory induced by partial-response modulation and/or convolutional pre-coding, [1], [2]. This combination of smoothness and memory provides constant-envelope designs which have intrinsically good power spectra and enhanced energy efficiency. The term "trellis" arises from the finite-state trellis description which is possible for rational modulation indices, as would be the case in any practical implementation.

Study of such signals for use on the white Gaussian noise channel with coherent detection is based on finding the minimum distance, in the L_2-norm, between any pair of signals $S_1(t)$ and $S_2(t)$. This is a considerably more difficult problem than finding the minimum Hamming distance of a convolutional code since the group property is lacking and each "transmitted" sequence may have a unique set of distances to all "received" sequences. Thus to exhaustively determine minimum distance for signals of length N symbols with M-ary modulation requires examination of M^{2N} pairs of sequences of length N, or NM^{2N} calculations of distance increments, which can be viewed as a basic unit of computation.

If one considers a phase tree in which the phase trajectories of all signals are plotted as a function of time, it is possible to imagine two signals having the same trajectories up to a certain point, their paths diverging then meeting again at some later time. It is desired to determine the minimum distance between any pair of signals which have split and remerged.

Previous approaches to finding the minimum distance have in fact explored all pairs of paths in the phase tree which split at the initial tree level . Pruning rules have been added so that if the cumulative distance exceeds the present minimum distance for any merged pair, then descendants of the current pair need not be considered. Pairs which are still contenders for establishing the free distance, defined as the absolute minimum distance for all pairs of signals, are kept alive by putting them on a memory stack [3] or by use of a forward/backward search in the tree, [4]. The former is fast relative to the latter, but can be quite consumptive of memory. Neither, however, exploits the important finite-state nature of the signal (for rational modulation indices), which is the intent of this paper.

Specifically, we show an algorithm for finding the distance to depth N whose complexity is upper-bounded by $\mathcal{O}(2^{2\nu} \cdot M^2 \cdot q \cdot N)$ units of computation, which is linear in N although the multiplier constant may be large. In the above 2^{ν} is the number of "data states" in the encoder/modulator, M is the number of signalling options in each interval, and q is the denominator of the modulation index, $h = p/q$. Furthermore, memory resources necessary are very manageable, $\mathcal{O}(2^{2\nu} \cdot q)$ in contrast to the potentially large stack size required for previous tree-searching algorithms, [3].

We shall describe and illustrate the algorithm in the context of convolutionally-coded CPFSK, [2], [4], although we note that the algorithm may be applied to any finite-state phase-coded modulation scheme. These range from MSK on the simple side to M-ary partial-response FM [1] and multi-h codes [5] on the more complicated side. A necessary restriction is that h be rational. Obviously any irrational h can be closely approximated by a rational h, with sufficiently large q, making the trellis size large. We don't feel this is any practical limitation, however, for actual implementations of modems will be restricted to rational h with small p and q for the same complexity reasons.

II. BACKGROUND

Let the modulated signal over the interval $nT < t < (n+1)T$ be described as

$$S(t, \overline{b}) = \left(\frac{2E}{T}\right)^{1/2} \cos\left(\omega_c t + \phi(t, \overline{b})\right) \tag{1}$$

where $\phi(t, \overline{b})$ is the phase modulation in response to the data sequence $b = (b_1, b_2, \ldots b_n)$. E is the energy per symbol, and T is the symbol interval. We may write the phase in general as

$$\phi(t, \overline{b}) = f(b_n, b_{n-1}, \ldots b_{n-\nu+1}, \theta_n) = f(S_n, \theta_n, b_n) \tag{2}$$

where S_n is the "data state" comprised of a finite number of previous data symbols and θ_n is the "phase state", which amounts to the cumulative phase induced by all data symbols whose influence has ceased (see e.g. [1], [4]). In the unmodulated CPFSK case for example,

$$\theta_n = \sum_{j=1}^{n-1} b_j \, h\pi \tag{3}$$

The "state" of the modulator is defined as $X_n = (S_n, \theta_n)$, and the structure of the coding/modulation scheme induces a known state-transition equation. For $h = p/q$, a finite-state representation having $2^\nu \cdot q$ states follows, with ν playing the role of "constraint length", or memory length of the combined coder/modulator.

To decode such a signal in maximum likelihood fashion, the Viterbi algorithm may be applied to a trellis having $2^\nu \cdot q$ states. For the Gaussian channel, path metrics are correlations between the received signal and the hypothesized transmitted signal. 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 which are merged at some time, split, then remerge later. Or, for finite-memory receivers, we seek the minimum distance between pairs which split and are of length N, denoted d_N. We also define the free distance

$$d_{free} = \lim_{N\to\infty} d_N \tag{4}$$

as usual.

The appropriate distance measure for this problem is

$$d_{12}^2(N) = \int_0^{NT} [S_1(t) - S_2(t)]^2 dt \tag{5}$$

which reduces to, for $\omega_c >> 2\pi/T$,

$$d_{12}^2(N) = \left(\frac{2E}{T}\right) \int_0^{NT} [1 - \cos \Delta\phi(t)]dt \tag{6}$$

where $\Delta\phi(t)$ is the time-varying phase separation between the two signals of length NT seconds. Observe that the value for d_{free} depends on h, the modulation index; $\phi(t,\overline{b})$, the phase modulation charactersitic; and the "memory" of the encoding process which translates into longer remerger times.

III. ALGORITHM DESCRIPTION

In order to examine all pairs of sequences we form a pair-state trellis. Each node in the trellis is defined by a triple consisting of S_{1_n} and S_{2_n}, the two data states, and $\Delta\phi_n = \theta_{1_n} - \theta_{2_n}$, the phase difference at the start of the current interval. It is only necessary to specify the modulo 2π phase difference which in fact allows us to use a finite-state model and a trellis. It may be readily shown that there are q such phase differences, where again $h = p/q$. Thus if there are 2^ν data states we now have $q \cdot 2^{2\nu}$ pair-states in the trellis. The M^2 transitions from states at level n to states at level n+1 are defined by the encoder/modulator structure and have incremental distances associated with them.

We note that the cumulative squared distance may be recursively computed as

$$d^2_{12}\,(n+1) = d^2_{12}\,(n) + \left(\frac{2E}{T}\right) \int_{nT}^{(n+1)T} [1 - \cos \Delta\phi(t)]dt \qquad (7)$$

and is completely specified by transitions in the pair-state trellis. 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 the pair-state trellis. This is completely analogous to the shortest-route problem and may be solved by application of the dynamic programming solution in the form of the Viterbi algorithm, [6]. The "principle of optimality" here is that if a sequence pair (b_1, b_2) is to produce the minimum distance event, it will do so via extensions 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.

To locate the minimum distance for each pair state j we fetch previous distances for the M^2 pair states which transition to j, add the incremental distances associated with those transitions, make roughly M^2 comparisons to find the minimum, and save this distance. If it is desired to locate the minimum-producing sequence pair, path maps can be stored and updated as well. Since there are $2^{2\nu} \cdot q$ pair-states and M^2 branches/state we say the complexity is $\mathcal{O}(2^{2\nu} \cdot q \cdot M^2 \cdot N)$ for an N-level trellis.

Actually the same pruning rules used in earlier implementations may be applied to speed the calculation. Specifically, some initial (optimistic) estimate of d^2_{free} is entered, and if the minimum distance of a pair leading to a pair-state merge in the trellis (of which there are $2^{2\nu}$ at every level) falls below the initial estimate, the d^2_{free} value is decreased to the newly found value. d^2_{free} is decreased further if other subsequent merges give lesser distances. Simultaneously, whenever the cumulative distance to a <u>non-merged</u> pair-state exceeds the current d^2_{free}, this pair-state need not be extended in the next trellis level, for descendants of this pair-state cannot ultimately produce the d^2_{free} event.

The distance profile as a function of n, $d^2_{min}(n)$, is also easily obtained by locating the <u>minimum</u> of distances over all survivor pairs to states at level n.

To initialize the algorithm, we assign zero distance to all merged pair-states in the trellis, and large initial distance to all others. In effect this allows the search for pair sequences beginning with any merged state, and thus the examination of all possible starting states is incorporated automatically. In contrast, the tree-searching procedures have simply rooted the tree search in each possible initial state and found a global distance minimum over this set of states.

A flow-chart of the algorithm is shown in Figure 2. To give better feel for the level of complexity, we list in Table 1 the trellis parameters for a variety of pertinent cases. Application of the algorithm to seemingly disparate cases is rather easy: all that must be done is identify a proper state-description and its associated state transitions, and to define a case-specific subroutine which computes a table of incremental distances for various pair-state transitions.

The comparison between the trellis-based algorithm described here and the tree-searching algorithms used previously is not as one-sided (exponential versus linear in N) as the above complexity relations indicate. Pruning rules in both algorithms speed the execution substantially, and in fact, [1] reports search time much faster than exponential in N. Still the trellis algorithm will be faster since it exploits pair-state merges while the tree-based algorithms do not. In addition, our memory requirements are generally much smaller, being known in advance as well, in contrast to the less predictable stack-size requirement of tree-searching methods. Specific comparisons are offered in the next section.

IV. EXAMPLES AND DISCUSSION

Our present interest is in evaluating the use of convolutional coding combined with CPFSK and other CPM methods as a combined power/bandwidth efficient scheme. The large number of possible code and modulation parameters makes an efficient distance computation essential.

First we consider the example of the rate 1/2, constraint-length 4 (ν = 3) encoder of Figure 3. The two output code symbols are treated as a single quaternary symbol in the set {-3, -1, 1, 3} with natural-binary mapping assumed, i.e. 00 $\to$ -3, 11 $\to$ +3. We have earlier found this scheme to be optimal in its class for h $\lesssim$ 1/4. The free distance as a function of h for this code is plotted in Figure 3, where we see monotone-increasing d^2_{free} up to h slightly beyond 1/4. Thereafter other merger events dominate the distance, and this code is no longer optimal, [4]. For certain values of h, 1/4 and 1/2 in this case, the distance has isolated small values owing to unusually short merger events. These are known as weak modulation indices, and other codes perform better at these h.

Also shown in Figure 4 and 5 are the distance results for r = 2/3, ν = 2 coded octal CPFSK (the optimal small h code in this class [7]) and for 4-ary uncoded CPFSK, for which the distance result was already known. Both schemes fit into the trellis phase code framework and are also readily evaluated by this procedure. These distance plots were obtained by computing distance for all rational h with h $\leq$ 1/2 and q $\leq$ 50. Due to this discretization, the distance near the weak modulation indices is not represented exactly.

Also readily available from the algorithm is the "distance profile," $d_{min}^2(n)$ versus n for various h, or vice-versa. This reveals how much receiver memory is necessary so that all unmerged pairs have distance exceeding d^2_{free}. For example, for the $r = 1/2$ code, with $h < 1/4$, 11 symbol delay is sufficient, while for $r = 2/3$, $h < 1/8$, 12 symbol delay is adequate.

Table II lists execution times for the algorithm described here and a tree-searching procedure similar to that of [3]. Both were programmed in FORTRAN by reasonably competent programmers, using similar data types and structures. We see the new method is faster as predicted, although the tree method's pruning rules allow it to be reasonably close. The speed advantage diminishes as q increases due to the increasing trellis size. In this case, many of the pair-states are "dead", but merely checking this fact reduces the efficiency somewhat. As a general conclusion we can say the trellis algorithm has its biggest gain when the tree is in fact mergeable into a small trellis, which occurs for smaller memory lengths and q small.

Bibliography

[1] Aulin, T., Rydbeck, N., and Sundberg, C-E.W., "Continuous-Phase Modulation, Parts I and II," IEEE Transactions on Communications, March, 1981.

[2] Wilson, S. G., and Pizzi, S. V., "Results on Convolutional Coding Combined with Continuous-Phase Modulation," Int'l Symposium on Information Theory, Les Arcs, France, June, 1982. (in submission to IEEE Trans. on Information Theory).

[3] Aulin, T., Lindell, G., and Sundberg, C-E., "Minimum Euclidean Distance for the Best Combinations of Short Rate 1/2 Convolutional Codes and Binary CPFSK Modulation," Technical Report TR-160, November 1981.

[4] Pizzi, S. V., "Modulation and Coding for Power and Bandwidth Efficiency," Ph.D. dissertation, University of Virginia, May, 1982.

[5] Anderson, J. B., and Taylor, D. P., "A Bandwidth-Efficient Class of Signal Space Codes," IEEE Transactions on Information Theory, November, 1978.

[6] Viterbi, A. J., "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Trans. on Information Theory, April, 1967.

[7] Mulligan, M. G., Masters thesis research in progress, University of Virginia.

Signal Type	# pair-states		branches/state
MSK, h = 1/2	2	(2)	4
4-ary CPFSK, h = 1/4 (uncoded)	4	(4)	16
4-ary 2 RC, h = 1/4	64	(40)	16
8-ary CPFSK, h = 1/8	8	(8)	64
r = 1/2, ν = 2, coded 4-ary CPFSK, h = 1/4	64	(40)	4
r = 2/3, ν = 2, coded 8-ary CPFSK, h = 1/8	128	(80)	16
binary {4/8, 5/8} multi-h code	8	(8)	4

Table I.

Pair-State Trellis Parameters for Several Trellis Phase Codes

Case	r	h	ν	Trellis Algorithm*	Tree Algorithm*	$d^2_{free}/2E_b$
I	1/2	1/4	3	13 sec.	47 sec.	5.24
II	"	1/5	3	16	32	4.30
III	"	3/16	3	27	39	3.50
IV	"	1/16	3	21	36	0.48
V	2/3	1/8	2	21	--	2.56

* Run in FORTRAN on PDP 11/03 microcomputer.

Table II.

Comparison of Execution Times for Tree and Trellis Algorithms

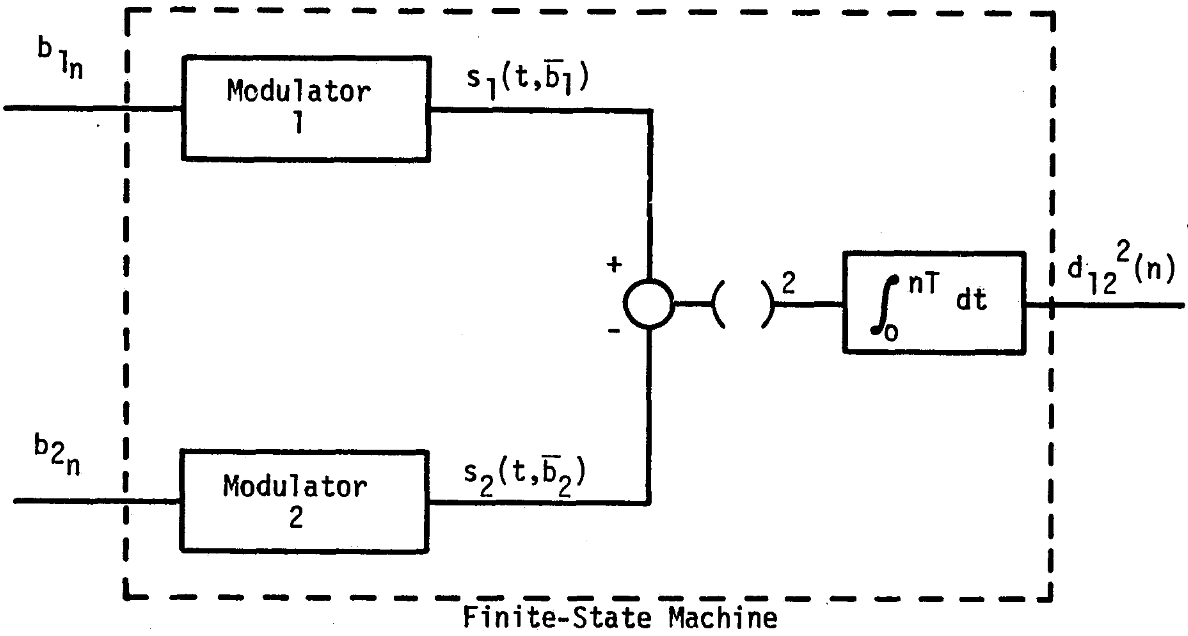
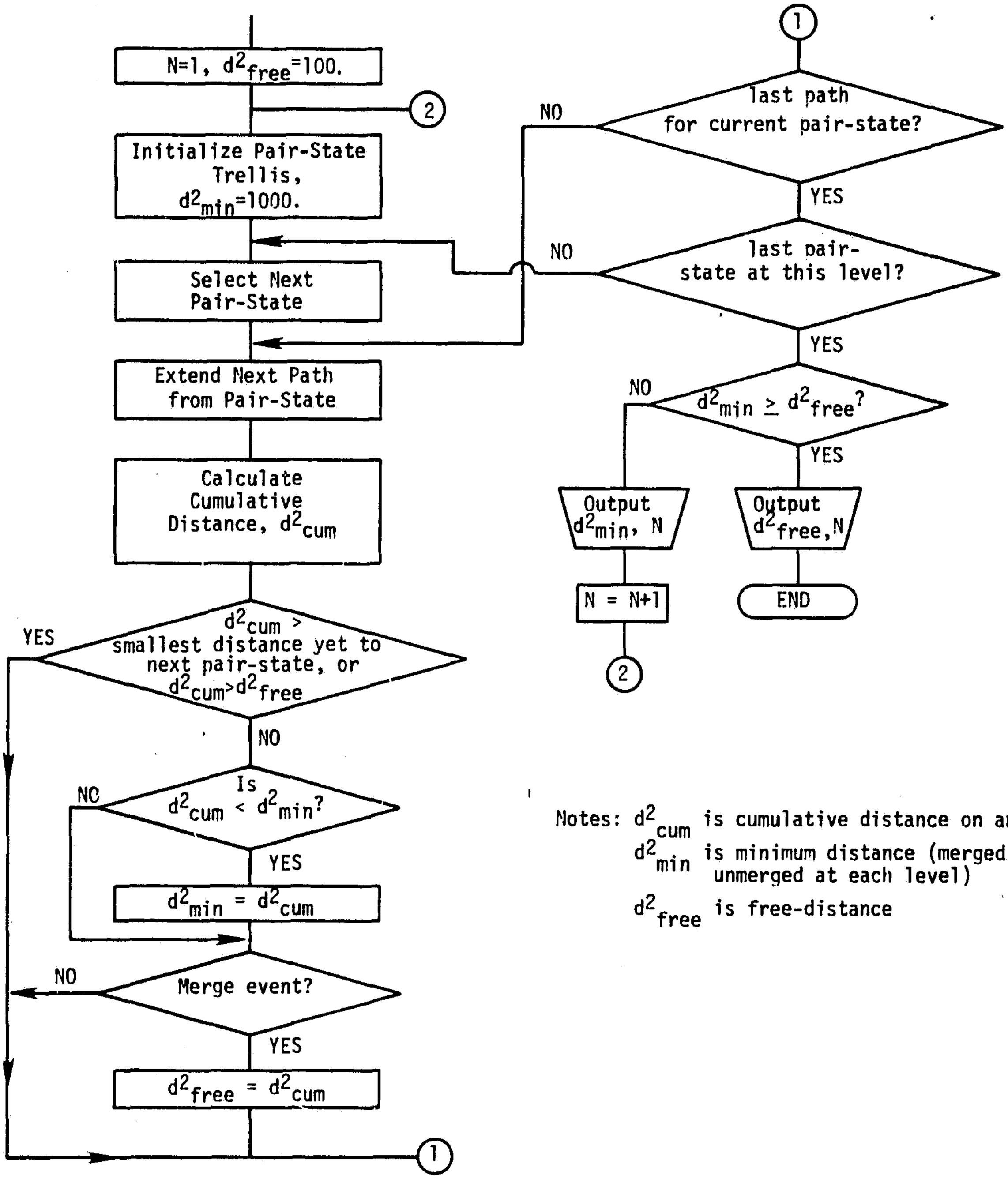


Figure 1. Conceptual Description of Distance-Finding Problem: Vector-Input Sequence Produces Scalar-Output Distance According to Finite-State (Trellis) Description.



Notes: d^2_{cum} is cumulative distance on any path
d^2_{min} is minimum distance (merged or unmerged at each level)
d^2_{free} is free-distance

Figure 2. Flow-Chart for Distance Calculation

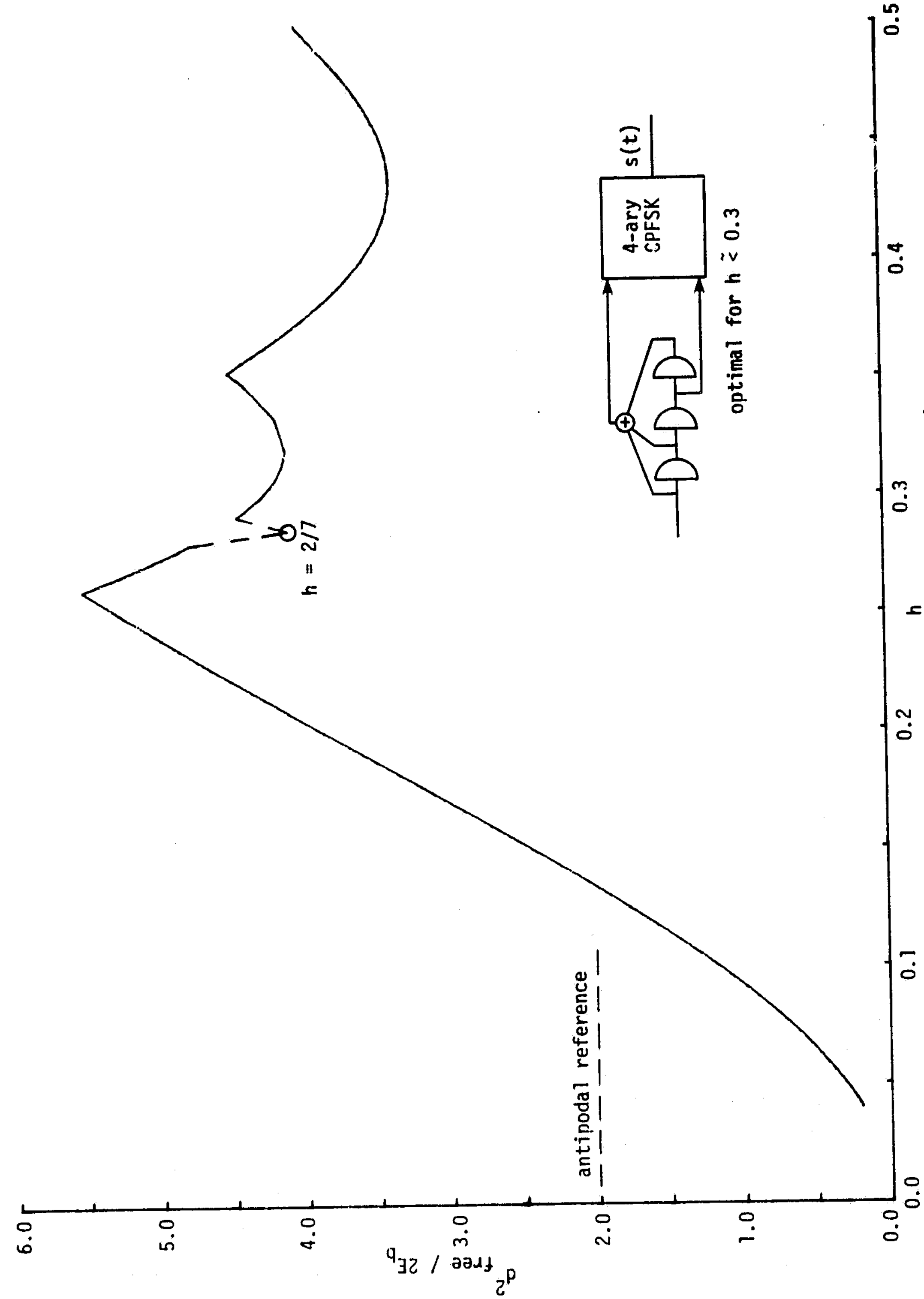


Figure 3. Distance versus h for r = 1/2, ν = 3, coded 4-ary CPFSK

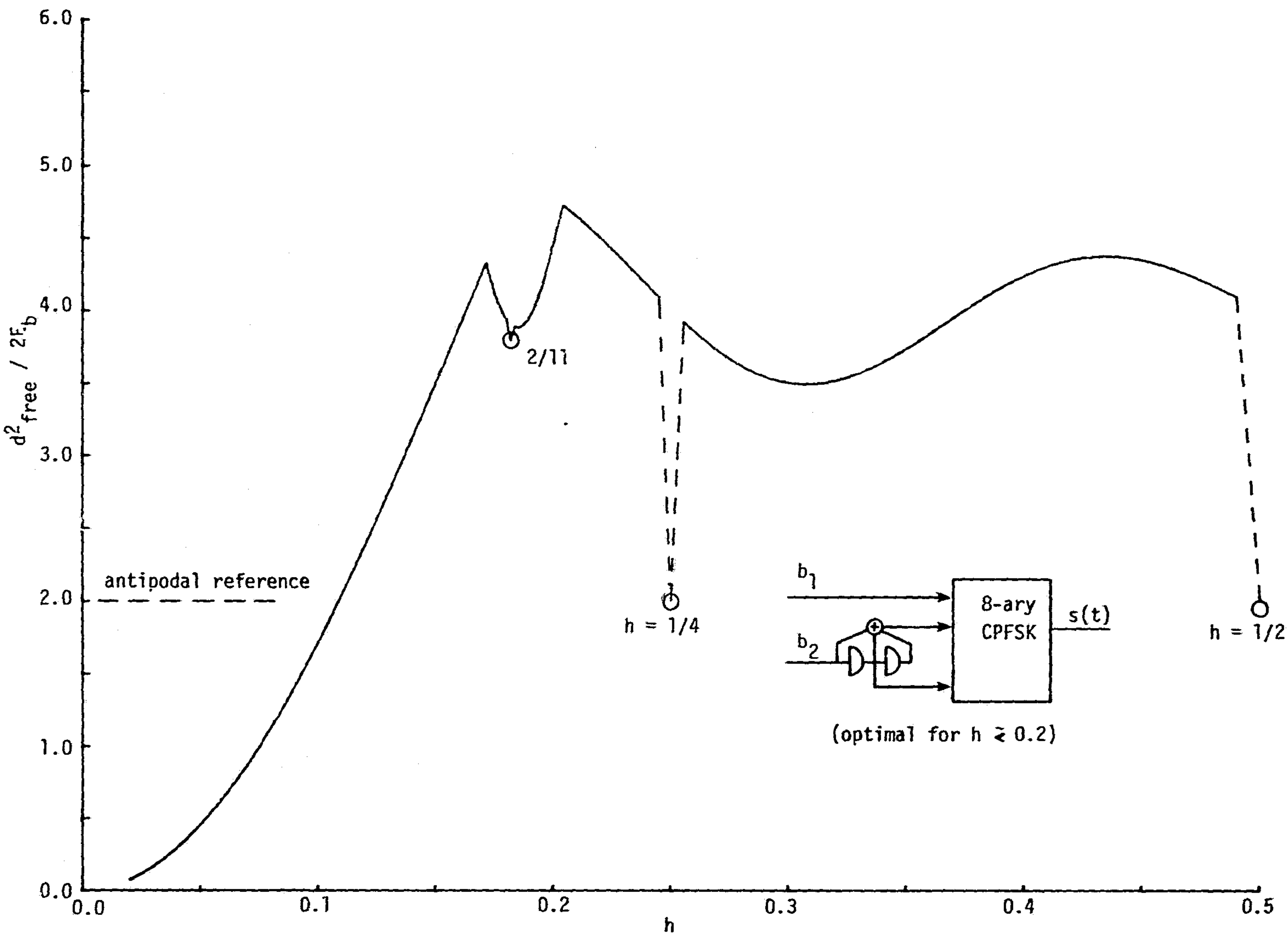


Figure 4. Distance versus h for r = 2/3, ν = 2, coded octal CPFSK

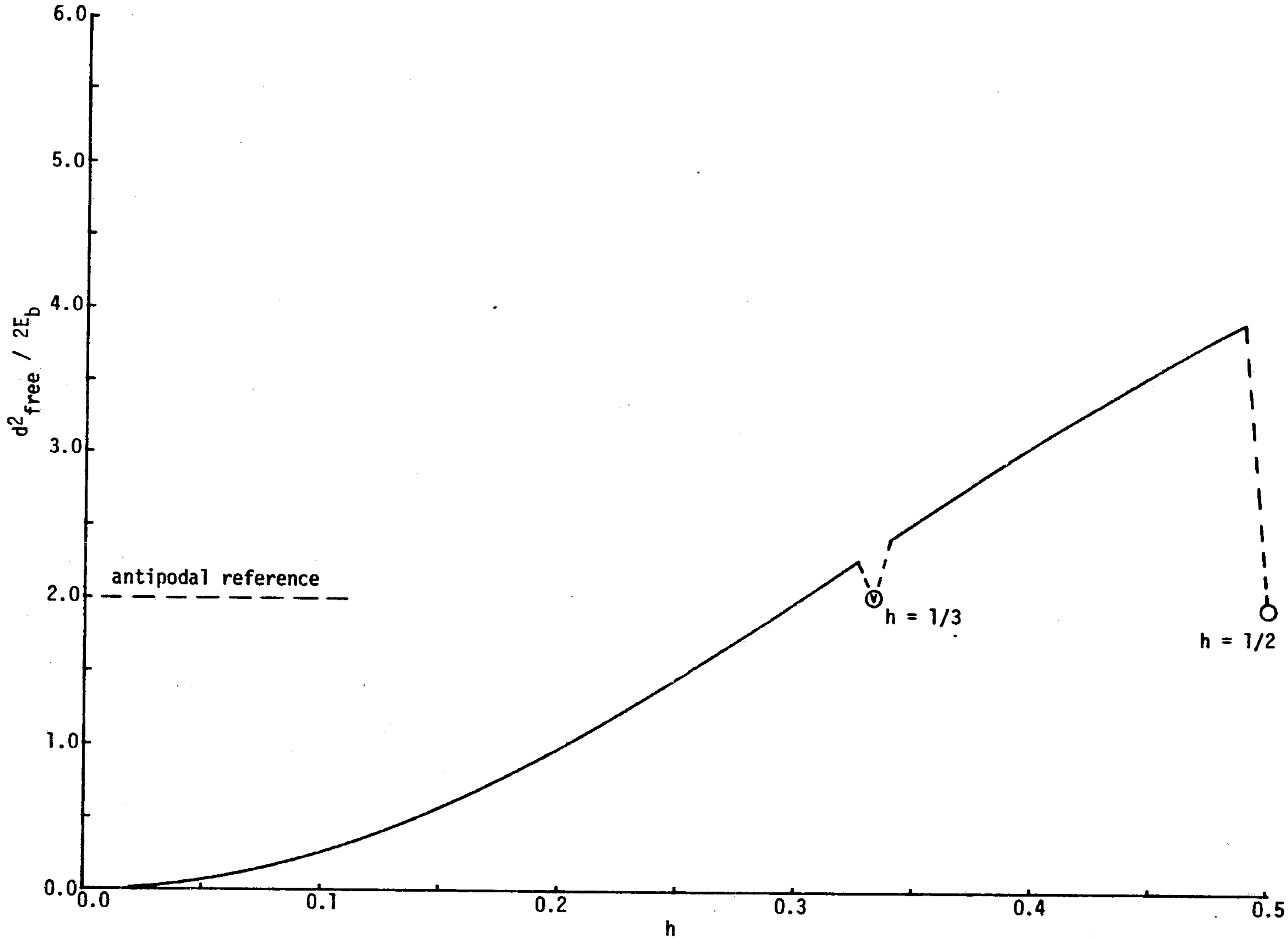


Figure 5. Distance versus h for 4-ary CPFSK, uncoded