SEPARABLE CONCATENATED CODES WITH ITERATIVE MAP DECODING FOR RICIAN FADING CHANNELS

J.H. Lodge and R.J. Young
Communications Research Centre,
3701 Carling Ave., Ottawa, Canada. K2H 8S2
(tel) 613-998-2284, (fax) 613-990-6339

ABSTRACT

Very efficient signalling in radio channels requires the design of very powerful codes having special structure suitable for practical decoding schemes. In this paper, powerful codes are obtained by combining comparatively simple convolutional codes to form multi-tiered "separable" convolutional codes. The decoding of these codes, using separable symbol-by-symbol maximum *a posteriori* (MAP) "filters", is described. It is known that this approach yields impressive results in non-fading additive white Gaussian noise channels. Interleaving is an inherent part of the code construction and consequently these codes are well suited for fading channel communications. Here, simulation results for communications over Rician fading channels are presented to support this claim.

1. INTRODUCTION

In practice, very efficient signalling in radio channels requires more than the design of very powerful codes. It requires designing very powerful codes that have special structure so that practical decoding schemes can be used with excellent (but not necessarily truly optimal) results. Examples of two such approaches include the concatenation of convolutional and Reed-Solomon coding, and the use of very large constraint-length convolutional codes with reduced-state decoding. In this paper, an alternate approach is introduced. The initial simulation results are very encouraging.

The work discussed in this paper was motivated by concepts introduced in [1] for the decoding of concatenated convolutional codes. In that paper it is shown that symbol-by-symbol MAP decoding for the inner code allows soft decisions to be passed to the outer decoder, resulting in impressive performance. The inner decoding algorithm can be thought of as a type of nonlinear filter that accepts as its input a noisy signal. Then it makes use of the structure inherent in the inner code to produce a noisy output "decoded" signal (that is hopefully less corrupted in some sense than the original input signal). Here we apply a similar philosophy to the decoding of separable convolutional codes. A "separable code" is defined to be a concatenated code where component codes and interleaving are chosen and combined in such a way that any codeword of the resulting composite code has the special property that it can be subdivided into valid codewords corresponding to any one of the component codes by appropriately grouping the output bits into code symbols [2][3].

The organization of this paper is as follows. In Section 2 some of the background behind the concept is summarized. We discuss the system model and MAP "filtering" for convolutional codes. Separable convolutional codes, and the use of separable MAP "filters" for decoding these codes, are described in Section 3. Simulation results for communication over Rician fading channels are presented in Section 4.

2. BACKGROUND

The symbol-by-symbol MAP algorithm can be used for codes that can be represented by a trellis of finite duration. For the system model shown in Figure 1, we provide a brief summary of the symbol-by-symbol MAP algorithm as given in [4] and the appendix of [5]. The simple time-invariant 4-state trellis, shown in Figure 2, is used to illustrate the concepts. This trellis corresponds to a rate-1/3 convolutional code. In general, the trellis may be time-varying with the number of states, M_t, being a function of the time index t. It is assumed that at the start and the end of the time interval of interest, the coder is in the zero state. Any given input sequence ${}_iD_{i'}$ of binary (e.g., 0 or 1) k-vectors, that satisfies the above end conditions, will correspond to a particular path through the trellis that is described by a sequence of states

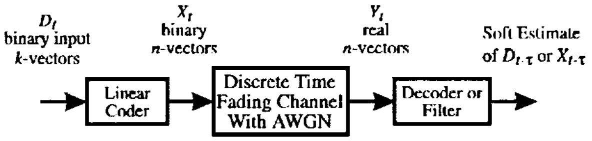

Figure 1. A block diagram of the system model.

$$ {}_{i-1}S_{i'} = \{S_{i-1} = 0,\ldots,S_t = m,\ldots,S_{i'} = 0\} \qquad (1)$$

where $S_t \in \{0,\ldots,M_t\text{-}1\}$.

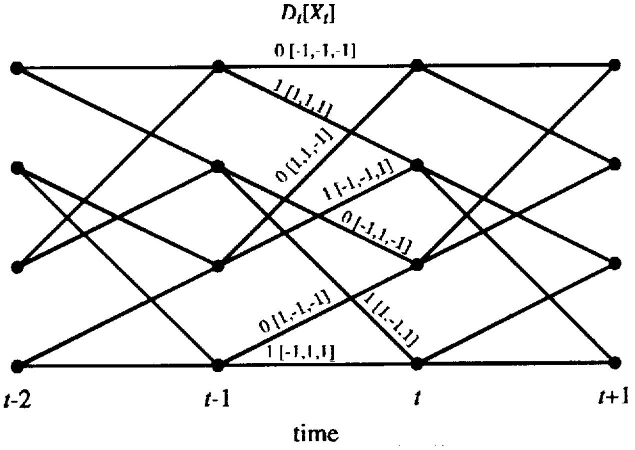

Figure 2. A trellis corresponding to a rate 1/3, 4 state convolutional code.

For each path through the trellis the coder produces a particular channel input sequence

$$_{i}X_{i'} = \{X_i, \ldots, X_t, \ldots, X_{i'}\}, \tag{2}$$

where X_t is an n-vector denoted by

$$X_t = [x_{t_1}, \ldots, x_{t_n}] \tag{3}$$

of binary (e.g., -1 or +1) elements. In the example trellis of Figure 2, k=1, n=3 and M=4 for all t. For notational convenience, the functional dependence of X_t on S_{t-1} and S_t is only shown when required. The corresponding channel output sequence is given by

$$_{i}Y_{i'} = \{Y_i, \ldots, Y_t, \ldots, Y_{i'}\}, \tag{4}$$

where Y_t is an n-vector denoted by

$$Y_t = [y_{t_1}, \ldots, y_{t_n}] \tag{5}$$

with the real-valued elements having conditional probability density functions given by

$$p(y_{t_j} | x_{t_j}) = \left(2\pi\sigma^2\right)^{-1/2} \exp\left[-(y_{t_j} - G_{t_j} x_{t_j})^2 / 2\sigma^2\right] \tag{6}$$

where G_{t_j} is the time-varying gain of the fading channel. Clearly this model is appropriate for antipodal signalling over a flat fading channel with additive thermal noise, typical of mobile satellite communications, under the assumption that the demodulator is able to accurately determine the gain and phase of the fading channel.

Now consider the problem of determining the *a posteriori* probabilities (APP) of the state transitions

$$p_t(m',m) = \Pr\{S_{t-1} = m';\ S_t = m |_{i}Y_{i'}\} = \frac{p(S_{t-1} = m';\ S_t = m;_{i}Y_{i'})}{p(_{i}Y_{i'})} \tag{7}$$

Throughout the paper, we shall refer to probability densities such as the numerator in (7) as a "probability", with the understanding that dividing it by $p(_{i}Y_{i'})$ makes it a true probability. Following [4], we use the joint probability

$$\sigma_t(m',m) = p(S_{t-1} = m';\ S_t = m;_{i}Y_{i'}), \tag{8}$$

recognizing that $p_t(m',m)$ can be computed from $\sigma_t(m',m)$ by either dividing by the constant $p(_{i}Y_{i'})$ or equivalently by the sum of all possible joint transition probabilities at the time t. It can be shown [4] that the above joint probabilities can be expressed as the product of three independent probabilities;

$$\sigma_t(m',m) = \alpha_{t-1}(m')\gamma_t(m',m)\beta_t(m), \tag{9}$$

where

$$\gamma_t(m',m) = p(S_t = m;\ Y_t | S_{t-1} = m') \tag{10}$$

$$\alpha_t(m) = \sum_{m'=0}^{M_{t-1}-1} \alpha_{t-1}(m')\gamma_t(m',m) \tag{11}$$

$$\beta_t(m) = \sum_{m'=0}^{M_t-1} \beta_{t+1}(m')\gamma_{t+1}(m,m') \tag{12}$$

Here we refer to $\gamma_t(m',m)$ as the branch probability and it is given by

$$\gamma_t(m',m) = \Pr\{S_t = m | S_{t-1} = m'\} \prod_{j=1}^{n} p\left(y_{t_j} | x_{t_j}(m',m)\right) \tag{13}$$

where the first term on the right-hand side is usually a straightforward function of the probability distribution of the input data and the coder structure. The second term on the right-hand side is a product of conditional symbol probability densities as given in equation (6). The branch probabilities account for the "present" n-vector of channel outputs, while the "past" channel outputs are accounted for by the forward recursion defined by equation (11), and the future channel outputs are accounted for by the backward recursion in equation (12).

Consider applying these techniques to obtain the *a posteriori* probabilities of the **coded** bits (i.e., the elements of X_t) rather than on the **information** bits (i.e., the elements of D_t). If the coded bits are assumed to be independent, with p_0 and p_1 being the probability that any given bit is a 0 or 1, respectively, then

$$p(x_{t_j} = 0;\ _{i}Y_{i'}) = p(x_{t_j} = 0;\ y_{t_j}) = p(y_{t_j} | x_{t_j}) p_0 \tag{14}$$

However, the coded bits are not independent due to the structure imposed by the coder. Consequently, we would like to use the MAP processing to determine the probabilities, $p(x_{t_j}=0; {}_iY_{i'}|C)$, where the conditioning on C refers to the knowledge of the coding structure. This can easily be done by defining the set of all transitions for which $x_{t_j}=0$;

$$A = \{(m',m): x_{t_j}(m',m) = 0\}, \qquad (15)$$

and then summing over the joint transition probabilities to obtain the joint probability

$$p(x_{t_j} = 0; {}_iY_{i'}|C) = \sum_{(m',m)\in A} \sigma_t(m',m). \qquad (16)$$

The noisy codeword enters the MAP "filter" as a vector of independent probabilities, and then is output from the filter with the probabilities (which are no longer independent) being refined according to the structure of the code. A similar procedure can be used for determining the probability that the information bit d_{t_j} is zero by replacing the set A by

$$A' = \{(m',m): d_{t_j}(m',m) = 0\}. \qquad (17)$$

In this paper, we distinguish between the terms "MAP filter" and "MAP decoder", with the former computing the *a posteriori* probabilities of the coded bits and the latter the *a posteriori* probabilities of the decoded bits. (Clearly for systematic codes, the *a posteriori* probabilities of the information bits are a subset of the probabilities for the coded bits.) If hard decisions are performed on the output of the MAP filter, the minimum average probability of coded bit error is achieved. However, the resulting word may not be a valid code word. A good choice for a valid codeword can be obtained by iterating the filtering operation until a valid code word is obtained. Of course, the assumption of independent probabilities by the MAP algorithm is erroneous when the algorithm is used iteratively.

3. SEPARABLE CONVOLUTIONAL CODES AND ITERATIVE MAP FILTERING

Recall that a separable code is defined to be a concatenated code where component codes and interleaving are chosen and combined in such a way that any codeword of the resulting composite code has the special property that it can be subdivided into valid codewords corresponding to any one of the component codes by appropriately grouping the output bits into code symbols. Next, we describe a technique that results in a very large powerful convolutional code by appropriately combining smaller component convolutional codes.

The first important observation is that convolutional encoders are linear and shift-invariant [6]. Therefore a sum of valid codewords, each with a different delay, is still a valid codeword. The second is that time-division interleaving can be implemented as is illustrated in Figure 3. Note that this structure does not destroy the shift-invariant property, unlike most interleaving schemes. Therefore this type of combined encoder/interleaver can be used as a building block for the type of composite code that is desired. This concept is illustrated in Figure 4 for a two-tier example code. Each tier contains a number of identical coders with inputs interconnected to the coder outputs of the previous tier. The interconnection must be done such that the codewords arriving from the previous tier are linearly combined through the current tier in such a way that the outputs can be subdivided into valid codewords for the previous tier. For example, in Figure 4, c_{11}, c_{12} and c_{13} are three valid codewords for code CE1. In general, these three codewords may not be identical to the two codewords generated by the first tier.

Here, we develop such an interconnection using a

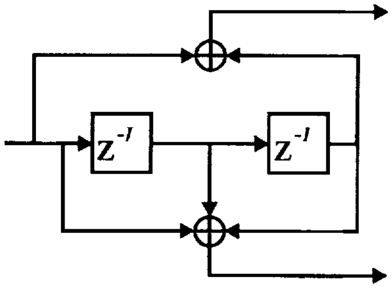

Figure 3. An example convolutional encoder including I-fold time division interleaving.

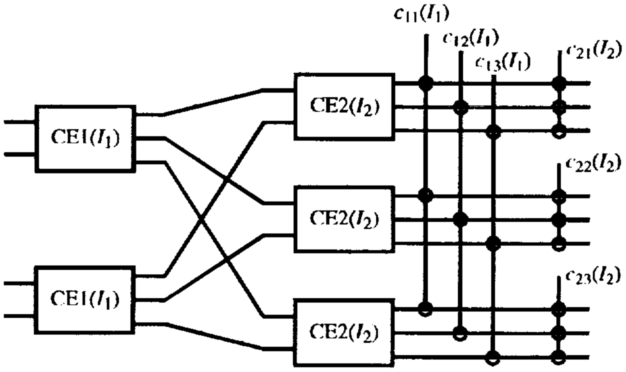

Figure 4. Two-tier coding with rate 2/3 component codes. CE1(I_1) is an encoder with I_1-fold interleaving for code 1. CE2(I_2) is an encoder with I_2-fold interleaving for code 2. c_{1q} is the qth valid codeword for code 1, with I_1-fold interleaving. c_{2q} is the qth valid codeword for code 2, with I_2-fold interleaving.

recursive approach. Starting with a rate k_1/n_1 convolutional coder at the first tier, we wish to add a second tier consisting of rate k_2/n_2 coders. In our interconnection there will be k_2 coders at tier 1 and n_1 coders at tier 2. The concatenation of tier 1 and tier 2 is treated as a supercoder of rate $k'_2/n'_2 = k_1k_2/n_1n_2$. To connect a third tier of rate k_3/n_3 coders we repeat the above process. There will be k_3 supercoders at tier 2 and n'_2 coders at tier 3 and after the interconnection this will produce a supercoder of rate $k'_3/n'_3 = k'_2k_3/n'_2n_3$. In general, interconnecting tier i to tier i+1 requires k_{i+1} supercoders at tier i and n'_i coders at tier i+1. This concatenation is treated as a supercoder of rate k'_ik_{i+1}/n'_in_{i+1} for subsequent interconnections. The final supercoder resulting from concatenating N tiers of convolutional coders has a rate

$$\frac{k'_N}{n'_N} = \frac{\prod_{i=1}^{N} k_i}{\prod_{i=1}^{N} n_i}. \tag{18}$$

The actual interconnection of tier i to tier i+1 is straightforward. If we denote the jth coder at stage i as $c_{i,j}$, then our interconnection strategy is to connect the mth output of supercoder $c_{i,j}$ to the jth input of coder $c_{i+1,m}$.

The individual codewords from the convolutional coders are dispersed as they propagate through subsequent tiers. In order to facilitate MAP filtering, we must be able to construct valid codewords from each tier. Let us denote the output sequence of n'_N bits as

$$\{b(0), b(1), b(2), \ldots, b(n'_N - 1)\}.$$

Then, the mth code symbol from the ith tier is

$$\{b(m), b(m+p), b(m+2p), \ldots, b(m+(n_i-1)p)\}$$

where

$$p = \begin{cases} \prod_{j=i+1}^{N} n_j, & \text{for } i < N \\ 1, & \text{for } i = N \end{cases} \tag{19}$$

and

$$m = \left\{0, 1, 2, \ldots, \frac{n'_N}{n_i} - 1\right\}. \tag{20}$$

Note that each of the component codewords (appropriately interleaved) is present at the output. The purpose of the interleaving is to make the distance of the composite code approximately proportional to the product of the distances of the component codes. Usually, it is desirable to choose the interleaving factors for the tiers to be mutually prime.

In multidimensional signal processing, digital filtering is often performed using "separable" filters. That is, in order to avoid excessive computational requirements, one-dimensional filtering is performed sequentially in each of the N dimensions, rather than performing a single massive N-dimensional digital filter. In this paper, we investigate the analogous approach for the decoding of multi-tiered codes. That is, MAP filters will be used sequentially for each tier. Consider the two-tier case first. MAP filtering can be performed on the codewords corresponding to the first tier giving a new set of refined probabilities, taking into account only the structure of the first component code. These new probabilities are then further refined by MAP filtering the codewords corresponding to the second tier to complete a single filtering cycle. This process can be iterated any number of times. The extension to the cases with more than two tiers is obvious. In the multidimensional signal processing case, iterating the filtering does not make sense because the filters are linear. However, in the separable coding case, the filters are highly nonlinear and additional filtering cycles can significantly improve the performance. In the final cycle, decoding with the MAP algorithm (defined at the end of section 2) should be used in order to recover the information bits.

In processing a continuous stream of received bits, some form of block processing is necessary because receiver memory and delay are not unlimited. However, by nature, convolutional codes are not ideally suited to block processing. Our strategy is to overlay a two segment processing window onto the incoming stream. The first segment of the window identifies the portion of bits that will be decoded and the second segment acts as a view into the future for the processing. After each decoding process is completed, the window is moved forward to the position just past the last decoded bit. The forward and backward recursions of the MAP processing are performed over the entire window, however, the decoding phase does not output bits from the future segment.

There is memory that must be carried forward from one block to the next. This memory consists of the forward recursion probability vector for each cycle of each interleaved tier. The number of probability vectors carried forward is therefore given by

$$\#\text{ of }\underline{\alpha}'\text{s} = N_c \sum_{i=1}^{N} I_i \tag{21}$$

where N_c is the number of cycles of MAP processing, N is the number of tiers and I_i is the interleaving factor at the ith tier. The forward recursion probability vector is initialized at time 0 so that state 0 is probability 1, as given by

$$\alpha_0(i)=\begin{cases}1, & i=0\\ 0, & i=1,2,\ldots,M-1\end{cases} \tag{22}$$

where M is the number of states in the trellis. At the start of processing each block, the backward probability vector at time t_e, corresponding to the end of the block, is initialized such that all state probabilities are equal, as given by

$$\beta_{t_e}(i)=\frac{1}{M},\ i=0,1,\ldots,M-1. \tag{23}$$

Obviously, these are not likely to be the true backward recursion probabilities at this time, however, we do not decode bits from this segment of the sequence. If the future block is chosen large enough, then by the time the recursion reaches the segment that will be decoded, the backward recursion probabilities should be close to their true value.

For convenience in the MAP processing, we restrict the number of bits in the present and future blocks to be a multiple of a fundamental block size. We define this fundamental block size, B, as

$$B=n'_N\prod_{i=1}^{N}I_i\,. \tag{24}$$

Then, the number of bits in the present block is PB and the number of bits in the future block is FB. In order to minimize decoding overhead, P should be chosen to be much larger than F. Also, F must be chosen to be large enough to allow the backward recursion probabilities to reach their true values by the time they reach the segment to be decoded.

4. SIMULATION RESULTS AND DISCUSSION

The performance of MAP processing of signals transmitted though Rician fading channels was investigated by software simulation. The 2-tier concatenation of 16 state, rate 2/3 systematic codes shown in Figure 4 was used with I_1 and I_2, the interleave factors, being 15 and 16 respectively. The complete simulation model is shown in Figure 5. Random bits are encoded with the concatenated encoders and then passed to a 9×240 block interleaver. The concatenated encoders provide good code symbol interleaving but do not interleave the individual bits of the code symbol; the function of the block interleaver is to provide interleaving of the bits. The size of the interleaver was chosen to be equal to the fundamental block size of the simulation as described by equation (24). The output of the block interleaver is passed to the fading channel using antipodal signalling. The fading filter was designed with a 10% raised cosine frequency response and its 3 dB bandwidth was defined to be the fading bandwidth. For Rician fading channels, the k-factor is defined to be the ratio, in dB, of the average fading path power to the direct (ie., line-of-sight) path power. The output of the fading channel and the fading process itself are passed to individual block deinterleavers so that the fading process samples remain time aligned with the received signal samples. The received signal samples are then processed by the MAP algorithm which uses the channel information. The magnitude and phase of the fading process are removed from the received signal samples prior to the MAP processing. In addition, the knowledge of the time varying signal-to-noise ratio is used to correctly transform the samples to bit probabilities.

Bit error rate performance results were generated for an AWGN channel and Rician fading channels with fading bandwidths equal to 0.03 of the symbol rate and k-factors of -10 dB and -5 dB. For an assumed bit rate of 4800 bps and binary signalling, the above fading rate would be approximately 140 Hz. The simulation results can be seen in Figure 6. Interestingly, the strength and diversity of the code results in better performance with fading than without it, in low signal-to-noise conditions, due to the additional power in the fading bandwidth. While these results are quite encouraging, it should be noted that it is assumed here that the demodulator is capable of perfectly estimating the thermal noise spectral density, and the time-varying channel state (i.e.,

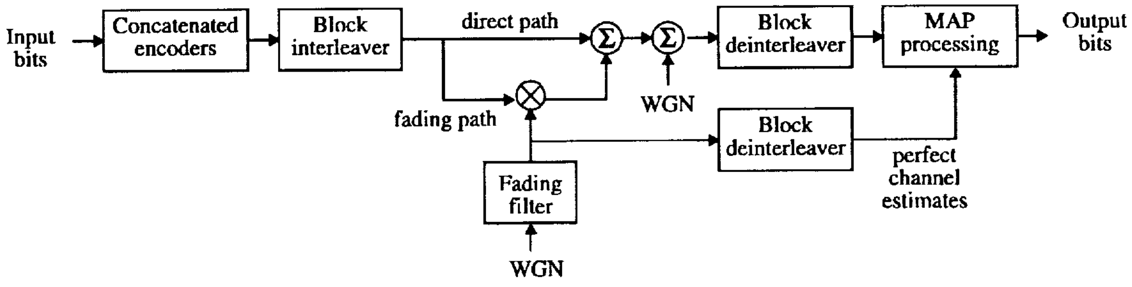

Figure 5. A block diagram of the simulation model.

magnitude and phase). Clearly, this is an optimistic assumption and consequently future work will be required to develop demodulators capable of providing the MAP decoders with the necessary inputs, and evaluating the resulting performance losses. One possible approach is to use reference symbols [7] to estimate the parameters of the fading channel. As a point of reference, Figure 7 shows the performance of the commonly used constraint length 7 rate 1/2 convolutional code, with ideal interleaving, perfect channel state information, and MAP decoding. Of course, this code can be decoded with much less delay and computational effort than the more powerful separable code.

As would be expected with such powerful coding techniques, the decoding process is quite computationally intensive. Therefore, the development of efficient implementation techniques is an important area for future work. For some codes, it is possible that simpler algorithms (e.g., [1]) can replace the MAP processing without severely degrading the performance.

While there still remain a number of areas for future work, the initial simulation results indicate that the iterative use of MAP "filters" for the decoding of separable convolutional codes can offer extremely power efficient transmission for those applications that can tolerate the large computational requirements, large block lengths, and long decoding delays that are typical of such powerful coding techniques.

REFERENCES

[1] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," Proc. GLOBECOM'89, Dallas, Texas, pp. 47.1.1-47.1.7, Nov. 1989.

[2] J.H. Lodge, P. Hoeher and J. Hagenauer, "The decoding of multidimensional codes using separable MAP "filters" ," Proc. Queen's University 16th Biennial Symp. on Communications, pp.343-346, May 1992.

[3] J. Lodge, R. Young, P. Hoeher, and J. Hagenauer, "Separable MAP 'Filters" for the Decoding of Product and Concatenated Codes," IEEE International Conference on Communications ICC'93, Geneva, Switzerland, May 1993.

[4] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," IEEE Trans. Inform. Theory, vol. IT-20, pp. 284-287, Mar. 1974.

[5] G.D. Forney, "The Viterbi algorithm," Proc. IEEE, vol. 61, pp. 268-278, Mar. 1973.

[6] G.D. Forney, "Convolutional codes I: Algebraic structure," IEEE Trans. Inform. Theory, vol. IT-16, pp. 720-738, Nov. 1970.

[7] M.L. Moher and J.H. Lodge, "TCMP - A modulation and coding strategy for Rician fading channels," IEEE J. Sel. Area. Comm., Vol. 7, pp. 1347-1355, December 1989.

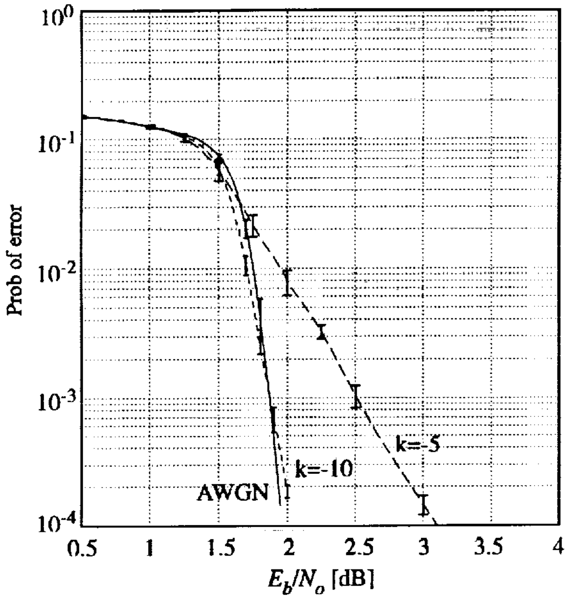

Figure 6. The average bit error rate versus the energy-per-bit-to-noise-spectral-density ratio for a 2-tier concatenated code in Rician fading environments, with four cycles of MAP processing.

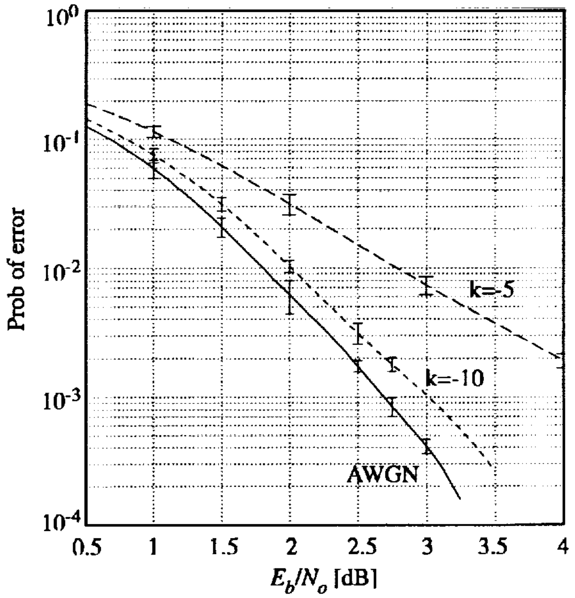

Figure 7. The average bit error rate versus the energy-per-bit-to-noise-spectral-density ratio for a single rate 1/2 code in Rician fading environments.