Symbol-Based Turbo Codes

Mark Bingeman and Amir K. Khandani

***Abstract*—We present a new turbo-coding method which parses the input block into n-bit *symbols* and interleaves on a *symbol-by-symbol basis*. This is used in conjunction with different modulation techniques to take advantage of tradeoffs between bit-error rate performance, code-rate, spectral efficiency, and decoder complexity. The structure of the encoder and decoder of these codes, which we call *symbol-based turbo codes*, are outlined. The bit error rate performance of a few specific codes are examined. A discussion on decoder complexity is also included. *Symbol-based turbo codes* are good candidates for low delay transmission of speech and data in spread spectrum communication systems.**

***Index Terms*—CDMA, modulation, turbo codes.**

I. Introduction

TURBO CODES, first introduced by Berrou *et al.* in [2], consist of a parallel concatenation of recursive systematic convolutional (RSC) codes with interleaving between parallel branches. We propose a new turbo-coding method which parses the input block into n-bit *symbols* and interleaves on a symbol-by-symbol basis. The systematic and encoded data streams are then mapped to a 2^n-point signal set. After this work was completed, we became aware of [3] which uses a similar symbol-based method applied to low-density parity check codes.

By parsing the input data block into n-bit symbols, we are in essence merging (or compressing) n sections of the encoder trellis into one. Fig. 1 shows two stages of the trellis diagram of an RSC encoder with generator polynomial $(13, 11)_8$. The solid lines and dotted lines correspond to the input bits of 0 and 1, respectively. Fig. 1 also shows the trellis diagram when two stages of the original trellis are merged together.

In general, there are 2^n branches leaving each state in an n-stage merged trellis diagram which correspond to the 2^n possible n-bit binary values. Note that when the number of branches leaving each state is greater than the number of states in the merged trellis, parallel transitions occur. These parallel transitions correspond to short error events which can not be broken by a symbol-by-symbol interleaver, and consequently, severely degrade the performance. These cases are avoided by selecting $M \geq n$, where M is the number of memory elements.

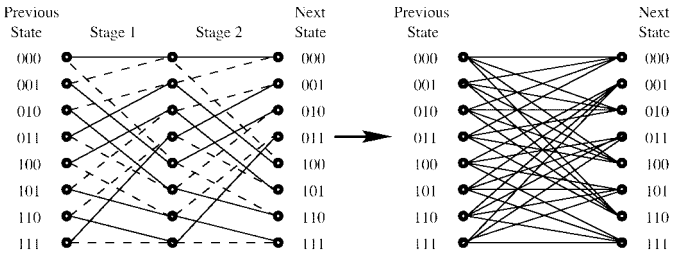


Fig. 1. Basic RSC and merged trellis diagrams.

TABLE I
Various Modulation Vectors for N-Bit Symbols

Technique	J	Modulation Vectors
n-BPSK	n	$\vec{m}_i = (\pm\sqrt{E_b}, \pm\sqrt{E_b}, \cdots)$
Orthogonal†	2^n	$\vec{m}_0 = (\sqrt{E_s}, 0, 0, \cdots, 0)$ $\vec{m}_1 = (0, \sqrt{E_s}, 0, \cdots, 0), \cdots$
Bi-orthogonal†	2^{n-1}	$\vec{m}_i = (\pm\sqrt{E_s}, 0, 0, \cdots, 0)$ $(0, \pm\sqrt{E_s}, 0, \cdots, 0), \cdots$
M-PSK	2	$\vec{m}_i = (\sqrt{E_s}\cos\theta_i, \sqrt{E_s}\sin\theta_i)$ $\theta_i = 2\pi i/2^n$

† Orthogonal and biorthogonal modulation schemes can also be implemented using Hadamard matrices and BPSK modulation

Consider a signal set with 2^n points and J dimensions. The jth component of modulation symbol is denoted by $m_{i,j}$ for $i = 0, 1, \cdots, 2^n - 1$ (the decimal value corresponding to the n-bit symbols), and $j = 1, 2, \cdots, J$. The J-dimensional modulation vector corresponding to a particular n-bit symbol is denoted by $\vec{m}_i = (m_{i,1}, m_{i,2}, \cdots, m_{i,J})$. Table I shows the modulation vectors for a few specific modulation techniques.

The modified BCJR algorithm [2], [4] is used for decoding, but some changes are required to make the constituent decoders operate on n-bit symbols using the n-stage merged trellis. The corresponding logarithm of the likelihood ratio (LLR) values are defined as

$$\Delta(d_k = i) = \log\left(\frac{\Pr\{d_k = i|\vec{R}\}}{\Pr\{d_k = 0|\vec{R}\}}\right), \qquad i = 0, 1, \cdots, 2^n - 1 \tag{1}$$

Manuscript received July 22, 1998. The associate editor coordinating the review of this letter and approving it for approval was Prof. K. Kiasaleh. This work was supported by the Natural Sciences and Engineering Research Council of Canada and by Canadian Institute of Telecommunications Research. This work was presented in part at the CITR Annual Research Conference, Toronto, Ont., Canada, August 1997, and also in the Proceedings of 19th Biennial Symposium on Communications, Queens University, Kingston, Ont., Canada, June 1998.

The authors are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ont., Canada (e-mail: msbingem@shannon.uwaterloo.ca; khandani@shannon.uwaterloo.ca).

Publisher Item Identifier S 1089-7798(99)08044-8.

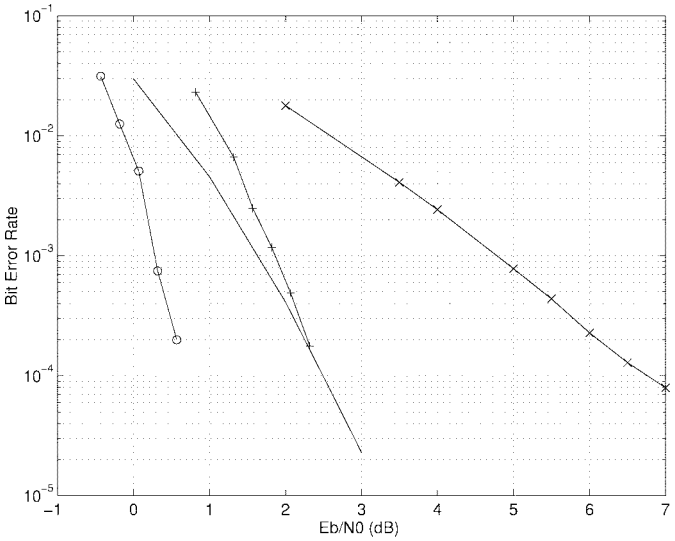


Fig. 2. BER performance of symbol-based turbocode ($N = 192$) versus convolutional code ($K = 9$) using orthogonal modulation.

where $\vec{R}$ denotes the received data streams. This definition for the LLR values allows for the easy conversion between symbol LLR values and symbol *a posteriori* probability (APP) values.

Assuming an additive white Gaussian noise (AWGN) channel, the branch metrics are given by [5]

$$\Pr\{x_k|d_k, S_k, S_{k-1}\} = e^{-(1/N_0)(x_k - b^s(d_k, S_{k-1}, S_k))^2} \quad (2)$$

$$\Pr\{y_k|d_k, S_k, S_{k-1}\} = e^{-(1/N_0)(y_k - b^p(d_k, S_{k-1}, S_k))^2} \quad (3)$$

where $b^{s|p}(d_k, S_{k-1}, S_k)$ is the modulator output associated with the branch from state S_{k-1} to state S_k at step k if the corresponding input d_k is equal to i. The $b^{s|p}()$ values correspond to the $m_{i,j}$ values discussed earlier. These equations can be generalized for any signal set by calculating them for each dimension and multiplying the resulting terms together.

The expression for the intrinsic information [2], [5] is also modified to take into account the modulation scheme used. A general expression for the intrinsic information is as follows:

$$\frac{1}{N_0}\sum_{j=1}^{J}[2x_{k,j}(m_{i,j} - m_{0,j}) + (m_{0,j}^2 - m_{i,j}^2)]. \quad (4)$$

II. Numerical Results

The BER performance of symbol-based turbocodes with various modulation techniques were obtained by simulation for both an additive white Gaussian noise (AWGN) channel, and a Rayleigh fading channel. The discrete model of the AWGN channel is given by the expression $\vec{y}_k = \vec{x}_k + \vec{z}_k$, where $\vec{x}_k$ is the transmitted symbol and $\vec{z}_k$ is a Gaussian random vector with independent and identically distributed (i.i.d.) components with mean zero and variance $N_0/2$. Similarly, the discrete model of the Rayleigh fading channel is given by the expression $\vec{y}_k = a_k\vec{x}_k + \vec{z}_k$, where $\vec{x}_k$ and $\vec{z}_k$ are the same as above, and the a_k's are i.i.d. random variables with a Rayleigh distribution of the form $f(a) = 2ae^{-a^2}$ for $a \geq 0$. For the Rayleigh fading channel, it is assumed that the receiver can determine the multiplicative fading factors.

The RSC encoders used in the simulations have four memory elements ($M = 4$) and generator polynomials $(37, 21)_8$.

TABLE II
Relative Computational Complexity and Memory Requirements of the Symbol-Based Turbocode Decoder

Symbol Size	1	2	3	4
Memory	1.00	0.69	0.69	0.86
Complexity	1.00	1.05	1.43	2.18

Assuming a data rate of 9.6 kb/s, the IS-95 delay specification of 20 ms corresponds to a maximum block length of 192 bits. This block length was used for all of the simulations. For symbol sizes $n > 4$, the RSC encoders with $M = n$ memory elements are used to avoid parallel transitions in the merged trellis diagram.

Symbol-based turbocodes with orthogonal modulation have rates of $n/3(2^n)$. For a symbol size $n = 6$, the resulting code has the same spectral efficiency as the IS-95 up-link (composed of a rate $1/3$ convolution code with 2^6-ary orthogonal signaling). Fig. 2 shows the BER performance of these two schemes for both an AWGN channel and a Rayleigh fading channel. It is observed that the symbol-based turbocode performs about 1.4 dB better at a BER of 10^{-3} for an AWGN channel, and about 2.9 dB better for a Rayleigh fading channel.

It should be mentioned that orthogonal modulation is in general a proper candidate for noncoherent signaling (as is the case in the IS-95 up-link). However, we have limited our investigation to coherent reception. This means that the curves in Fig. 2 should be considered only as an indication of the relative strength of the two schemes when they are used over a noncoherent channel. In practice, coherent systems can take advantage of biorthogonal modulation which will be discussed later.

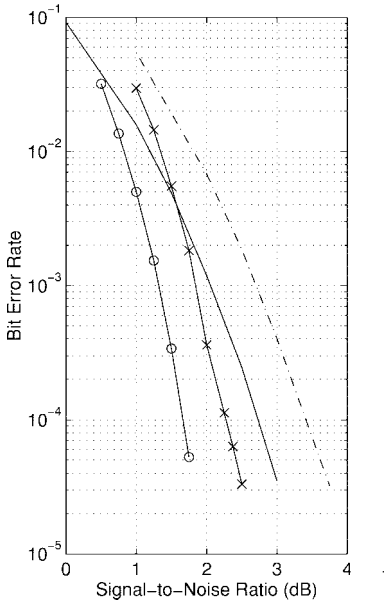
The symbol-based turbocodes with M-ary PSK modulation have rates of $n/6$, where n is the symbol size. The BER performance of these codes are compared with rate $1/3$ and rate $1/2$ convolutional codes with constraint length $K = 9$ for both AWGN and Rayleigh fading channels in Fig. 3. For a BER of 10^{-3}, the symbol-based turbocodes with M-ary phase-shift keying (PSK) modulation performs about 0.7 dB better than convolutional codes of the same rate and a constraint length of 9 over an AWGN channel, and about 2.2 dB better over a Rayleigh fading channel.

The third symbol-based turbocode simulated uses biorthogonal modulation. The rates of these codes are $n/3(2^{n-1})$. Fig. 4 shows the BER performance of these codes over an AWGN channel. Note that the symbol-based turbocode with symbol size $n = 1$ corresponds to the conventional binary turbo code.

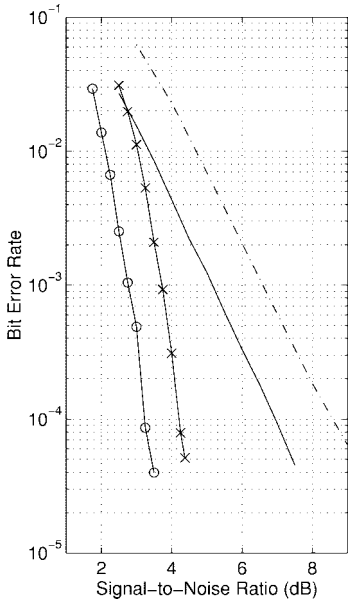
III. Complexity and Memory Requirements

The merging of n trellis stages results in a reduction in the effective block length by $1/n$. A shorter effective block length translates into fewer stages for the forward and backward recursions, and consequently, less α and β values need to be stored. On the other hand, the merged trellis have more branches leaving each state, and more LLR values need to be calculated and stored. Furthermore, the branch metrics require more computations and memory storage.

Table II shows the relative random access memory (RAM) requirements of the symbol-based turbocode decoder as com-



(a)



(b)

Fig. 3. BER performance of symbol-based turbo codes ($N = 192$) with M-ary PSK modulation versus convolutional code ($K = 9$). (a) AWGN channel. (b) Rayleigh fading channel. $\circ$: symbol-based turbo code, rate 1/3; $\times$: symbol-based turbo code, rate1/2; $-$: convolutional code, rate 1/3, $K = 9$; $- \cdot -$: convolutional code, rate 1/2, $K = 9$

pared to a traditional turbocode. This table clearly shows that the symbol-based turbocode decoder requires less RAM as compared to a traditional turbocode. Furthermore, the read-only memory (ROM) used to store the interleaver structure is also reduced by a factor of $1/n$.

Table II also shows the relative computational complexity of the symbol-based turbo-code decoder as compared with the traditional rate $1/3$ turbo-code decoder for $M = 4$ memory elements. The relationship between computational complexities per iteration versus that of the traditional turbo

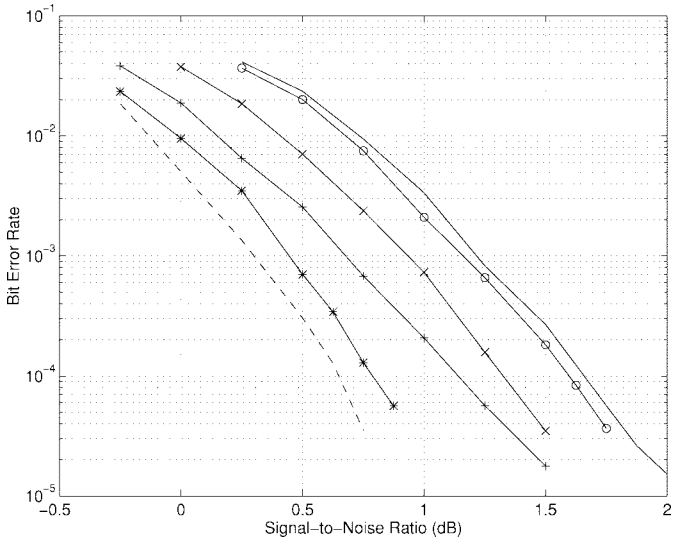


Fig. 4. BER performance of symbol-based turbo codes with biorthogonal modulation over an AWGN channel ($N = 192$).

code, $n = 1$, is roughly given by $2^{n-1}/n$. Therefore, the computational complexity per iteration for symbol sizes $n = 1, 2$ are approximately the same. However, we have observed that symbol-based turbo codes with BPSK modulation and symbol size $n = 2$ require approximately five iterations to obtain the same BER performance of the traditional turbo code with 10 iterations. Overall, symbol-based turbo codes with $n = 2$ result in an improved performance for the same number of decoding iterations[1] (or equivalently a reduction in the number of iterations for the same performance), as well as a reduction in the required memory size with respect to the conventional turbo codes $(n = 1)$ at no extra cost. For larger symbol sizes $n = 3, 4$, the increase in computational complexity is partially compensated by the reduction in the number of iterations required.

REFERENCES

[1] M. Bingeman and A. K. Khandani, "Design, performance analysis and decoding of combined turbocode and modulation," in *Proc. 19th Biennial Symp. on Communications*, Queens University, Kingston, Ont., Canada, June 1998, pp. 36–40.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbocodes," in *IEEE Int. Conf. on Communications 1993 (ICC'93)* Geneva, Switzerland, May 1993, pp. 1064–1070.

[3] M. Davey and D. MacKay, "Low-density parity check codes over $GF(q)$," *IEEE Commun. Lett.*, vol. 2, pp. 165–167, June 1998.

[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. 20, pp. 284–287, Mar. 1974.

[5] P. Robertson, "Illuminating the structure of code and decoder of parallel concatenated recursive systematic turbocodes," in *IEEE Global Telecommunications Conf. (GLOBECOM'94)*, San Francisco, CA, Dec. 1994, pp. 1298–1303.

[1] Recently, we became aware of the work of P. Sauvé and F. R. Kschischang, "Decoding turbocodes by multibit probability propagation," in *Proc. 19th Biennial Symp. on Communications*, Queens University, Kingston, Ont., Canada, June 1998, pp. 89–93. The authors independently arrive at a similar conclusion. They, however, consider only the improvement obtained in the performance for a fixed computational complexity and do not discuss the more important issue of the reduction in the memory size.