

A simple construction of low rate convolutional codes with application to low rate turbo-like code design

Durai Thirupathi and Keith M. Chugg

Communication Sciences Institute

Dept. of Electrical Engineering

University of Southern California, Los Angeles 90089-2565

Abstract — In this paper, we introduce a simple method to construct very low rate systematic recursive convolutional codes from a standard rate $1/2$ code. These codes in turn are used in parallel concatenation to obtain very low rate turbo codes. This construction always leads to a class of convolutional codes for which the signals on trellis transition are a subset of complete bi-orthogonal signal set for any given code rate. The resulting codes are rate compatible and require low complexity decoders. By applying this construction methodology to the rate $1/2$ turbo code used in 3G standard, we show that an additional coding gain of approximately 1.2 dB is achievable at a bit error rate of 10^{-5} .

I. INTRODUCTION

It is a well known fact that, for a given input block size, in an additive white Gaussian noise (AWGN) channel, reduction in the code rate beyond a certain threshold has diminishing returns [1]. Furthermore, if low rate convolutional codes are constructed by specifying the generator polynomial for each coded bit, then a substantial encoding complexity is incurred. In addition, for a convolutional code with fixed number of states, it may be difficult to achieve additional coding gain [5]. A considerable amount of research has been done in this area in the past few decades [13,15,16,17]. These codes find their application in systems where bandwidth expansion has no additional penalty (e.g. spread spectrum systems) or in systems which are power limited with abundance of bandwidth to utilize (e.g. ultra wide band systems).

After the discovery of turbo codes [10], very low rate turbo-like codes such as super-orthogonal turbo codes (SOTC) [3], turbo Hadamard codes (THC) [2], etc. have been introduced. The disadvantage of the SOTC is that the decoding complexity increases dramatically with decrease in code rate. On the other hand, for practical input block sizes (such as 1024 bits etc.) concatenated codes that are constructed using weaker constituent codes are outperformed by the ones that are constructed with stronger constituent codes [4]. So, in designing very low rate turbo-like codes, one of the crucial issues is to design strong enough constituent convolutional codes with decoding complexity of the overall code being almost independent of the code rate. It will also be useful if these codes can be designed with some in-built structures that can be utilized in other front end receiver processing tasks such as synchronization.

In this paper, we introduce a technique to construct such a class of low rate convolutional codes. These codes are then used to build very low rate turbo codes. Since these codes are

This work was supported in part by the Army Research Office (DAAD 19-01-1-0477).

constructed from rate $1/2$ parent codes, the complexity of the resulting encoder and decoder is very low. The performance of this code is about 0.6 dB away from the corresponding constrained channel capacity [1].

The rest of the paper is organized as follows. Section II briefly reviews the class of super-orthogonal convolutional codes. Section III describes the algorithm that is used in this work to construct low rate codes. Numerical results and conclusions are given in Sections IV and V, respectively.

II. BRIEF REVIEW ON SUPER-ORTHOGONAL CODES

Block codes are characterized by the parameters (n, k, d_{min}) where n is the length of the codeword, k the input alphabet size and d_{min} the minimum distance between any two codewords. The ratio k/n is the code rate. Orthogonal binary block codes are characterized by $(2^k, k, 2^{k-1})$ and the corresponding code rate is $k/2^k$. Here, the size of the code (i.e. the number of codewords in the code) is also 2^k . In other words, for a given k tuple, the encoding process is selection of one of the 2^k possible codewords. A bi-orthogonal code is a block code that includes an orthogonal code and its corresponding complementary set. The length of this code remains the same but the input alphabet size is increased by one, reducing the code rate to $(k+1)/2^k$. In the rest of this paper, we will refer to these codewords as ‘signals’.

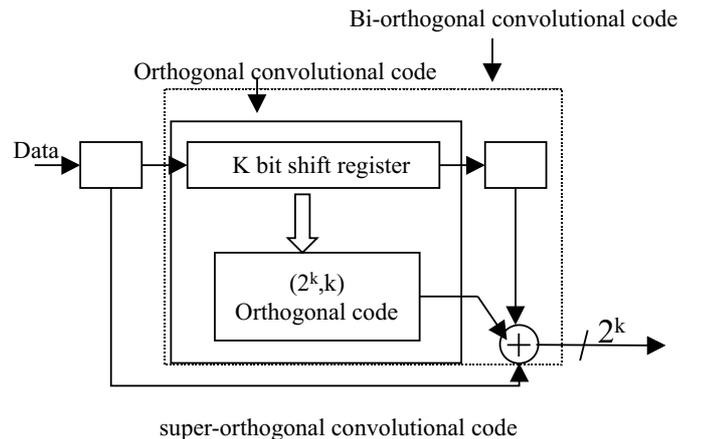


Fig. 1. Encoder structure for Orthogonal, Bi-orthogonal and Super-orthogonal codes

Orthogonal convolutional codes (OCC) of rate $1/2^k$ are de-

signed such that for any given state, the output signals on all the outgoing and incoming branches are pairwise orthogonal. Since these signals are selected from an orthogonal block code of size 2^k , the input alphabet size should be k . A block diagram of this encoder is given in Figure 1. As the rate of the convolutional code is $1/2^k$, only one data bit will be fed to the finite state machine (FSM) for each length 2^k output. Therefore, the memory of the finite state machine should be large enough ($k - 1$ to be precise) to access all the signals. In the trellis structure of the bi-orthogonal convolutional code (BOCC), the signals on the branches that merge into a state are orthogonal whereas those that originate from a state are antipodal or vice versa. Since the code size is 2^{k+1} for rate $1/2^k$, the memory of the finite state machine should be at least k to access all these signals. Figure 2 shows a section of trellis transitions of these codes. The dotted line corresponds to an input '1' and the solid line corresponds to an input '0'.

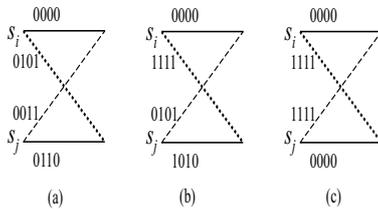


Fig. 2. A section of trellis transition of (a) Orthogonal code, (b) Bi-orthogonal code, (c) Super-orthogonal code

Super-orthogonal convolutional codes (SOCC) are a clever engineering construction of binary trellis codes in such a way that the signals on the trellis transitions from and to any given state are pairwise antipodal (as shown in Figure 2 (c)) [13]. In order to accomplish this kind of trellis structure, the number of states should be at least $2^{(k+1)}$. The total number of possible signals (which is twice that of the code rate) that can be used in a trellis step is same as that of the total number of states. So, when the code rate is decreased, the code size increases and hence the total number of states increases. For this class of codes to be considered for very low rate applications (rate $< 1/16$), it is imperative that this dependence is removed without disturbing the trellis structure.

In this work, we present an efficient way of removing the dependency of the state complexity to the code rate. This results in a class of 'simple' super-orthogonal codes.

III. CODE STRUCTURE

In order to construct very low rate codes, we start with a standard, N state, rate $1/2$ recursive systematic convolutional code and apply a combination of repetition and bit flipping. The algorithm used in this work can be described as follows.

- Divide the state space into m equal, even subsets ($Z_i, i = 1, 2, \dots, m$) such that the cardinality of $\bigcup Z_i = N$.
- For each Z_i , assign a one-to-one mapping with a combination of repetition and flipping of the corresponding output code symbol.

- Depending on the membership of the current state $s_j, j = 0, 1, 2, \dots, N - 1$, apply the mapping to the output.

The mapping that is used can be described recursively. The following example illustrates the construction algorithm. Let the number of states N be 16 and divide this space into four ($m = 4$) subsets. Let the required rate r be $1/8$. Let c denote the two coded output bits of the parent code and the corresponding current state be $s_j, j = 0, 1, \dots, 15$. Apply the following one-to-one mapping to obtain the required rate:

- If $s_j \in Z_1$, then the coded output is ccc .
- If $s_j \in Z_2$, then the coded output is $c\bar{c}\bar{c}$.
- If $s_j \in Z_3$, then the coded output is $c\bar{c}c$.
- If $s_j \in Z_4$, then the coded output is $c\bar{c}\bar{c}$.

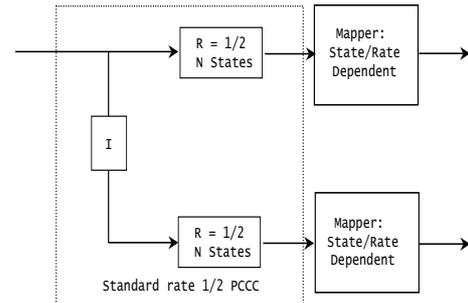


Fig. 3. Parallel concatenation of the proposed low-rate encoder structure

The schematic of the resulting encoder is given in Figure 3 where it is shown applied to each of the two constituent codes of the turbo code. Figure 4 shows the change in the trellis structure by the application of our algorithm to a four state rate $1/2$ parent code in order to obtain a rate $1/4$ code. In this case, the state space is divided into two subsets of two states each. Z_1 corresponds to states $\{s_0, s_1\}$ and Z_2 corresponds to states $\{s_2, s_3\}$. Let c be the coded output of the parent rate $1/2$ code at a given time instant. Then, if the current state belongs to Z_1 , the new coded output is cc whereas if the current state belongs to Z_2 , the output is $c\bar{c}$. Since we start with constituent systematic recursive codes of rate $1/2$, code rates (for the turbo codes) such as $1/7, 1/8, 1/15, 1/16, 1/31, 1/32$, etc. can be obtained directly with other rates achieved by puncturing.

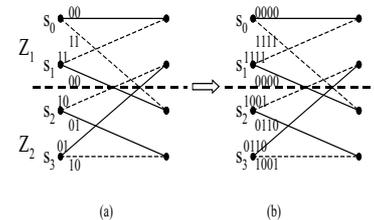


Fig. 4. Change in the trellis structure with application of our algorithm (a) parent code (b) resulting lower rate code

For any given state in the trellis, signals on all the incoming and outgoing branches are pairwise antipodal. In other words, the resulting code will always be super-orthogonal in

nature (and thus have good distance properties) for inverse of the code rates that are multiples of 4. For a code rate of $1/2^k$, at any trellis step, this code uses only a subset of the complete bi-orthogonal signal set. Figure 5 shows the trellis structure of the traditional super-orthogonal code for rate $1/4$ along with that of the ‘simple’ super-orthogonal code that results from our construction procedure. Notice that the traditional SOCC uses all the 8 possible signals while the ‘simple’ code uses only 4 signals.

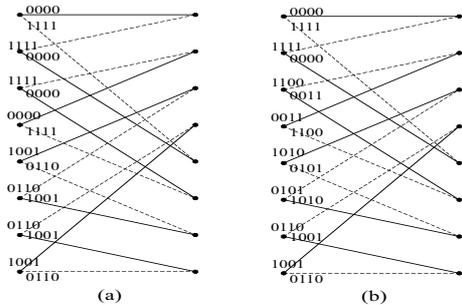


Fig. 5. Trellis diagram of (a) ‘simple’ SOCC (b) traditional SOCC

Because of the super-orthogonal nature, after an initial increase, the size of the signal set becomes a constant for a given state complexity irrespective of the change in the code rate (r). If we represent this signal set as a matrix C , then, for a specific number of states N , as the code rate decreases, C becomes a rectangular matrix of a maximum size of $N \times n$ where n is equal to $1/r$. The rank of this matrix is $\frac{N}{2}$ since there are $\frac{N}{2}$ linearly independent rows and their corresponding complementary sequences. Usually $n - N$ columns of C will be a linear combination of the rest N columns. However, by construction, as $r \rightarrow 0$ with a fixed number of states, several columns of C will be the same. So, it may be possible to puncture one or more of these columns to obtain any arbitrary rate with minimal coding loss.

The decoder used in this work is a standard turbo decoder [14,3] (with log domain sum-product algorithm). This can be implemented as $\min^*(\cdot)$ operation which is defined as

$$\begin{aligned} \min^*(x, y) &= -\ln(e^x + e^y) \\ &= \min(x, y) - \ln(1 + e^{|x-y|}). \end{aligned} \quad (1)$$

IV. NUMERICAL RESULTS

The parent code that we apply our algorithm to is a rate $1/2$ recursive systematic convolutional code. The feed forward and feedback polynomials of this code are 37 and 23 in octal, respectively. Two such codes are concatenated parallelly through an interleaver of size 1024 bits. The state space of each constituent code is divided into four subsets ($m = 4$) of four states each. The mapping technique explained in the example in section III is applied. Additive white Gaussian noise (AWGN) channel is considered. All the results that are shown here are obtained after 15 iterations unless specified otherwise.

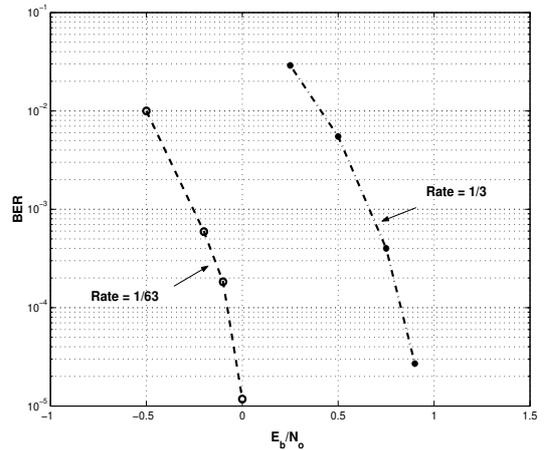


Fig. 6. Rate 1/3 parent code vs Rate 1/63 code

Figure 6 shows the coding gain that is achievable in using this technique in order to reduce the code rate from $1/3$ to rate $1/63$. Since both the parent codes are systematic, one systematic output can be punctured and hence we obtain a rate $1/3$ to begin with. It can be seen that for a 1024 bit interleaver, this simple scheme can achieve approximately 0.9 dB in additional coding gain.

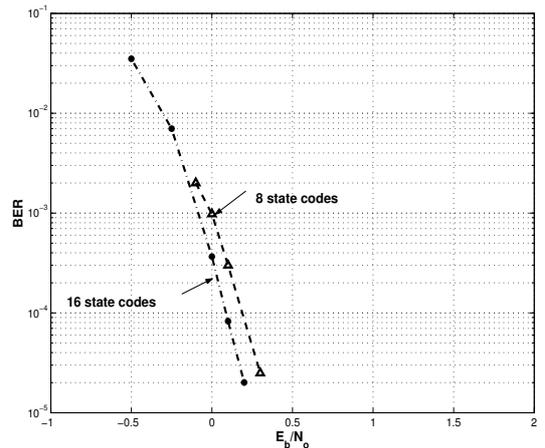


Fig. 7. Performance of the proposed encoding scheme as a function of state complexity (rate $1/63$)

In Figure 7 the performance of the proposed construction method as a function of number of states in each component code is shown. As explained before, a maximum of only 8 signals will be used for a 8-state code. So, to isolate the influence of the number of states on the performance, we have plotted the performance of a 16-state code that uses only 8 signals¹. Though not shown here, based on our simulations, we realize that component encoders with two and four states are not strong enough to handle such low rates. Since one of the main ob-

¹In order to make this 16-state code to use only 8 signals, we divide the state space into two subsets of 8 states each and apply the algorithm as shown in Figure 4.

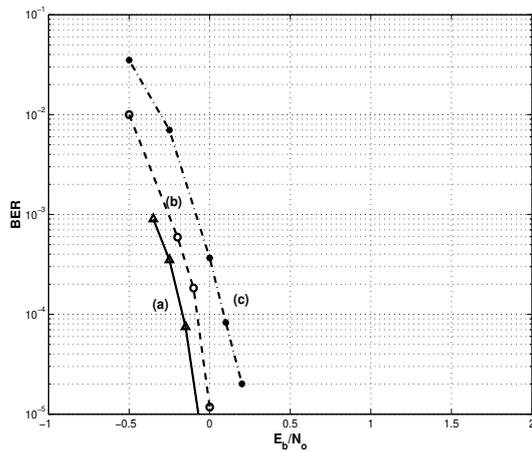


Fig. 8. Performance of various PCCC schemes with rate 1/32 constituent codes. (a) Traditional SOTC (b) Code constructed using our algorithm (c) Code constructed using our algorithm (with less number of code words)

jectives of this work is to reduce the state complexity, we have not simulated codes with more than 16 states each. From the results, we can see that a marginal gain of about 0.15 dB is possible in doubling the number of states.

The performance degradation of this code with respect to the corresponding traditional SOTC is evaluated and the result is shown in Figure 8. As can be noticed, the degradation is less than 0.1 dB at BER of 10^{-5} . However, our code structure results in about 75% complexity reduction at the decoder compared to the traditional SOTC.

The complexity is calculated in terms of total number of operations needed to decode one information bit. Since the code rate and number of iterations are same for all the code structures, the number of states is the only parameter that decides the complexity for a given interleaver size. Since the traditional SOTC for a rate 1/63 requires a total of 128 states compared to our scheme which require 32 states in total, a complexity reduction of about 75% is achieved.

In Figure 9, we have plotted the bit and frame error rate of a low rate code constructed by the application of our algorithm to the 8-state rate 1/2 turbo code used in the 3G standard. Since this standard includes wide band applications, it is worth evaluating the gain that is achievable by the application of our algorithm to this specific code. The results shown here are after performing 10 iterations. A performance gain of about 1.2 dB in terms of bit error rate or frame error rate is obtained.

V. CONCLUDING REMARKS

We have proposed a simple method to generate a class of very low rate turbo codes from rate 1/2 constituent codes that can be used for combined coding and spreading in spread spectrum systems. Apart from being rate compatible for various rates, these codes allow for low complexity encoder and decoder implementation.

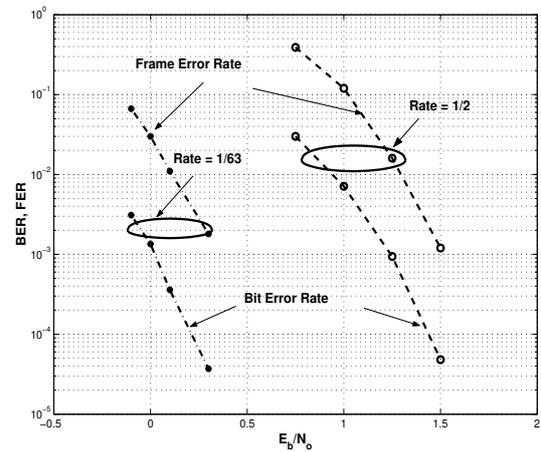


Fig. 9. Performance gain in BER and FER by the application of our algorithm to the 8-state rate 1/2 turbo code used in 3G standard

REFERENCES

- [1] S. Dolinar, D. Divsalar, and F. Pollara, "Code Performance as a function of Block Size," TMO Progress Report 42-133.
- [2] L. Ping, W. K. Leung and K. Y. Wu "Low rate Turbo-Hadamard Codes," in *Proc. ISIT 2001*.
- [3] P. Komulainen and K. Pehkonen, "Performance evaluation of superorthogonal turbo codes in AWGN and flat Rayleigh fading channels," *IEEE J. Selected Areas in Communications*, vol.16, no.2, pp.196-205, Feb. 1998.
- [4] T. J. Richardson, M. A. Shokrollahi and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 619-637, Feb. 2001.
- [5] P. D. Papadimitiou and C. N. Georghiades, "On Asymptotically optimum rate 1/n convolutional codes for a given constraint length," *IEEE Comm. Letters*, Vol. 5, NO. 1, pp. 25-27, Jan. 2001.
- [6] C. F. Leanderson, J. Hokfet, O. Edfors and T. Maseng, "On the design of low rate turbo codes," in *Proc. VTC 2000*.
- [7] R. Herzog, J. Hagenauer and A. Schmidbauer, "Iterative Decoding and Despreading improves CDMA-Systems using M-ary Orthogonal Modulation and FEC," in *Proc. of the IEEE International Conf. on Comm. (ICC)*, Montreal, Canada, pp. 909-913, June 1997.
- [8] J. Hagenauer, E. Offer and L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 429-445, Mar. 1996.
- [9] A. Risteski, P. Popovski and B. Spasenovski, "Joint Despreading and Iterative Decoding in DS-SS-CDMA System with Serially Concatenated Turbo Code and Walsh Code," in *Proc. ICPWC'2000*, pp.378-382.
- [10] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes," in *Proc. ICC'93*, Geneva, Switzerland, May 1993, pp. 1064-1070.
- [11] S. Benedetto and G. Montorsi "Unveiling turbo codes: some results on parallel concatenated coding schemes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 408-428, March 1996.
- [12] L. -N. Lee, A. R. Hammons Jr, F. -W. Sun and M. Eroz, "Application and standardization of turbo codes in third-generation high-speed wireless data services," *IEEE Trans. on Vehicular Technology*, Vol. 49, no. 6, pp. 2198-2207, Nov. 2000.
- [13] A. J. Viterbi, "Very low rate codes for maximum theoretical performance of spread spectrum multiple-access channels," *IEEE J. Select. Areas Comm.*, vol. 8, pp. 641-649, May 1990.
- [14] K. M. Chugg, A. Anastasopoulos, and X. Chen, *Iterative Detection: Adaptivity, Complexity Reduction, and Applications*, Kluwer Academic Publishers, MA, 2001.
- [15] J. P. Chaib and H. Leib, "Very low rate Trellis/Reed-Muller (TRM) Codes," *IEEE Trans. on Comm.*, vol. 47, No. 10, pp. 1476-1487, Oct. 1999.
- [16] A. Viterbi, *CDMA: Principles of Spread Spectrum Communication*, Addison-Wesley, Reading, USA, 1995.
- [17] P. D. Shaft, "Low rate convolutional code applications in spread spectrum

communications," *IEEE Trans. on Comm.*, vol-COM 25, No. 8, pp. 1476-1487, Aug. 1977.