A simple low rate turbo-like code design for spread spectrum systems

Durai Thirupathi and Keith M. Chugg
Communication Sciences Institute
Dept. of Electrical Engineering
University of Southern California, Los Angeles 90089-2565
{thirupat,chugg}@usc.edu

Abstract

In this paper, we introduce a simple method to construct very low rate recursive systematic convolutional codes from a standard rate-$1/2$ convolutional code. These codes in turn are used in parallel concatenation to obtain very low rate turbo like codes. The resulting codes are rate compatible and require low complexity decoders. Simulation results show that an additional coding gain of 0.9 dB in additive white Gaussian noise (AWGN) channel and 2 dB in Rayleigh fading channel is possible compared to rate-1/3 turbo code. The low rate coding scheme increases the capacity by more than 25% when applied to multiple access environments such as Code Division Multiple Access (CDMA).

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 of a given constraint length, it may be difficult to achieve additional coding gain [6]. A considerable amount of research has been done in this area in the past few decades [7, 9, 11]. Such 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 [2], very low rate turbo-like codes such as super-orthogonal turbo codes (SOTC) [4], turbo Hadamard codes (THC) [3], etc. have been introduced. Both these codes use a binary bi-orthogonal block code characterized by $(2^k, k+1, 2^{k-1})$ where $(k+1)$ is the input information size, 2^k is the output code word length and 2^{k-1} is the minimum distance of the block code. SOTC is a parallel concatenation of two super-orthogonal recursive convolutional codes. In the case of THC, k information bits are fed into a recursive single parity check encoder, the output of which is mapped onto a bi-orthogonal signal set. Therefore, in order for a one-to-one mapping it is enough to have a 2-state finite state machine. The disadvantage of the SOTC is that the decoding complexity increases dramatically with decrease in code rate. On the other hand, THC, which is constructed using weaker constituent codes has slower rate of convergence which might not be a desired property for practical applications. So, in designing very low rate turbo-like codes for practical applications, one of the critical issues is to design strong enough constituent convolutional codes with decoding complexity of the overall code being almost independent of the code rate.

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 constructed from rate-$1/2$ parent codes, the complexity of the resulting encoder and decoder is very low.

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. Application to spread spectrum systems is considered in Section IV. Numerical results and conclusions are given in Section V and VI, 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'.

One way of generating orthogonal (hence bi-orthogonal) signal set is by using Hadamard matrices. A simple method to generate a Hadamard matrix of size $2^k \times 2^k$ is through recursion i.e., a Hadamard matrix of size $2^k \times 2^k$ can be generated recursively from a Hadamard matrix of size 1×1 denoted by H_1, where $\mathrm{H}_1 = [1]$. The recursion is given by

$$\mathrm{H}_k \;=\; \begin{bmatrix} \mathrm{H}_{k-1} & \mathrm{H}_{k-1} \\ \mathrm{H}_{k-1} & \bar{\mathrm{H}}_{k-1} \end{bmatrix}, \qquad (1)$$

This work was supported in part by Army Research Office and National Science Foundation under the grants DAAD19-01-1-0477 and ANI-9730556, respectively.

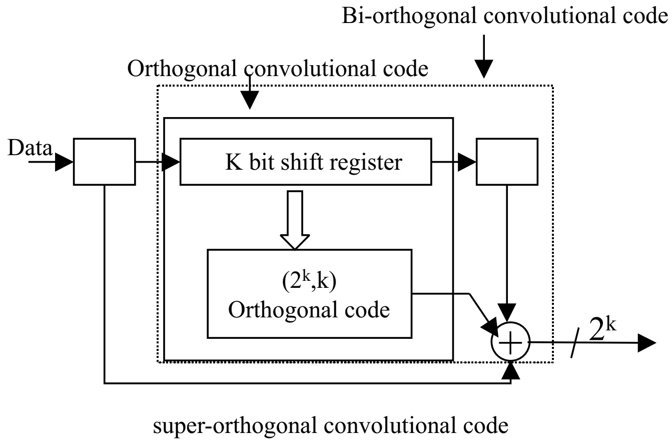
Bi-orthogonal convolutional code
Orthogonal convolutional code
Data
K bit shift register
(2^k,k)
Orthogonal code
2^k
super-orthogonal convolutional code

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

where $\bar{\text{H}}$ represent the complement of H. The Hadamard matrix of size $2^k \times 2^k$ and its corresponding complementary matrix form the required bi-orthogonal signal set.

Orthogonal convolutional codes (OCC) of rate $1/2^k$ are designed 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 $'0'$.

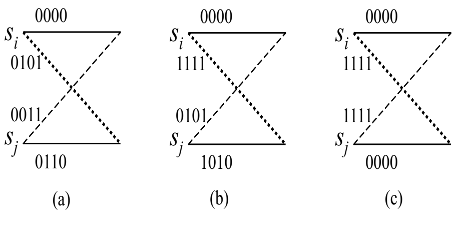

Fig. 2. A section of trellis transition of (a) Orthogonal code, (b) Bi-orthogonal code, (c) Super-orthogonal code

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)). 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. This, in turn increases the complexity of the decoder. Hence, for this class of codes to be considered for very low rate applications (rate $< 1/16$), it is imperative that this dependence be removed without disturbing the trellis structure.

III. Code Construction

In order to construct very low rate codes, we start with a 'good', N-state, rate-$1/2$ recursive systematic convolutional code and apply a combination of repetition and bit flipping. By good code, we mean that the signals emerging from and converging to any given state differ in maximum possible bit locations. Our algorithm can be described as follows:

- Divide the state space into m equal and even subsets ($Z_i, i = 1, 2, ..., m$) 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 $s_j, j = 0, 1, 2, ..., N-1$, 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. Let this space be divided 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 $s_j, j = 0, 1, ..., 15$. Apply the following one-to-one mapping to obtain the required rate:

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

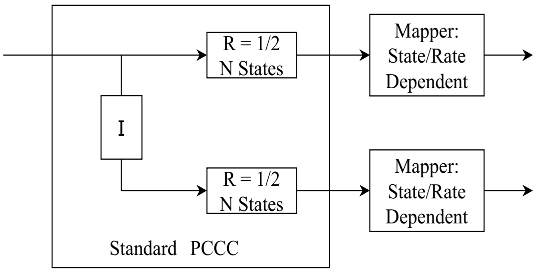

Fig. 3. Parallel concatenation of the proposed low-rate encoder structure

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 d 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 dd whereas if the current state belongs to Z_2, the output is $d\bar{d}$. Since this construction methodology is based on rate-1/2 codes with arbitrary number of states, the following observations can be made.

- code rates (turbo codes) such as 1/7,1/8,1/15,1/16,1/31,1/32, etc. can be obtained directly while other rates can be achieved by puncturing.
- code rate is independent of the number of states which translates to the code complexity being a design parameter [5].

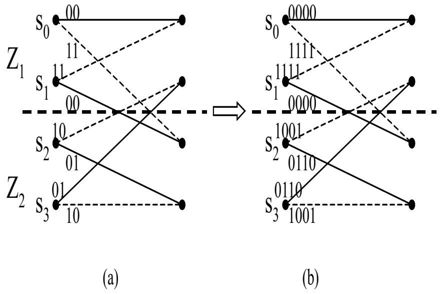

Fig. 4. Change in the trellis structure with application of our algorithm (a) parent code (b) resulting lower rate code

In addition, note that for any given state in the trellis, signals on all the incoming and outgoing branches are pairwise antipodal. In other words, for rates that are reciprocal of multiples of 4, codes constructed by this method will always be super-orthogonal [10] in nature. 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. In other words, the proposed code uses only a subset of the complete bi-orthogonal signal set which in turn leads to reduction in code complexity.

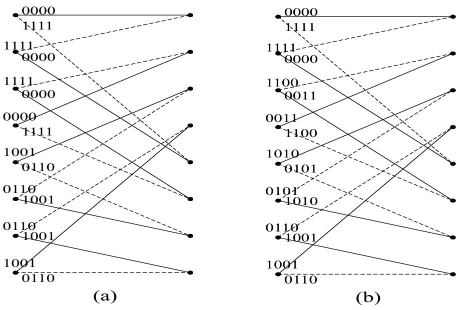

Fig. 5. Trellis diagram of (a) 'simple' SOCC (b) traditional SOCC

Let the code rate be $1/2^k$. Let B be the matrix representation of the complete bi-orthogonal signal set. Then B is of size $2^{(k+1)} \times 2^k$. Let $\hat{B}$ represent a subset of the bi-orthogonal signal set such that the size of $\hat{B}$ is $2^n \times 2^k$. For large enough k, we assume $n < k$. That is, for low encoding rate, our encoder selects signals from a relatively smaller subset of the complete bi-orthogonal signal set. By construction, as $k \to \infty$, with n fixed, several columns of $\hat{B}$ can be made to be 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 is a standard turbo decoder [3] (with log domain sum-product algorithm). The decoding algorithm can be implemented as min*(.) operation which is defined as

$$\begin{aligned} \min{}^*(x,y) &= -\ln(e^x + e^y), \\ &= \min(x,y) - \ln(1 + e^{|x-y|}). \end{aligned} \qquad (2)$$

IV. Application to Spread Spectrum Systems

In this section, the application of the proposed low rate turbo-like code to spread spectrum systems, especially to code division multiple access (CDMA), is considered. The schematics of the code-spread system using low rate codes and conventionally coded (rate-1/3 turbo code) and spread system is given in Figure 6. Three different channels, namely AWGN channel, block fading channel and multi-user channel with AWGN are considered.

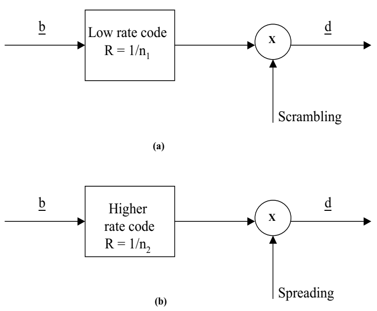

Fig. 6. (a) Low rate coded and scrambled system (b) Conventionally coded and spread system

A simple semi-analytic expression to estimate the degradation in signal to noise ratio due to the presence of multiple users in a low rate coded system can be derived. The matched filter output that corresponds to a coded symbol on a trellis transition can be written as

$$y_k = \sqrt{E_s} d_k^i + \sqrt{E_s} \sum_{j=1, j \neq i}^{K} d_k^j + n_k \qquad (3)$$

under perfect power control and synchronization. Here, n_k is zero mean Gaussian random variable with variance $N_o/2$, K is the total number of users and E_s is the coded symbol energy. Under the assumption that the multiple access interference can be modelled as Gaussian, it is straight forward to show that the additional information bit-to-noise ratio needed to achieve the single user performance can be approximated by

$$\left(\frac{E_b}{N_o}\right)_{req} \approx \frac{(\frac{E_b}{N_o})_1}{1 - R(\frac{E_b}{N_o})_1 (K-1)}, \qquad (4)$$

where $\left(\frac{E_b}{N_o}\right)_1$ is the bit-to-noise ratio needed for single user performance for a specific bit error rate (usually 10^{-5}). Equation (4) can be generalized to

$$\left(\frac{E_b}{N_o}\right)_{req} \approx \frac{(\frac{E_b}{N_o})_1}{1 - R\gamma^{-1}(\frac{E_b}{N_o})_1(K-1)}, \qquad (5)$$

where γ represents additional spreading involved.

V. Numerical Results

The algorithm is applied to 16-state rate-$1/2$ recursive systematic convolutional parent code used in [2]. The feed forward and feedback polynomials of this code are 37 and 23 in octal, respectively. Two such codes are used to build the turbo code. The interleaver size is 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 II is applied. Results shown are after 15 iterations in AWGN channel and 10 iterations in fading channel.

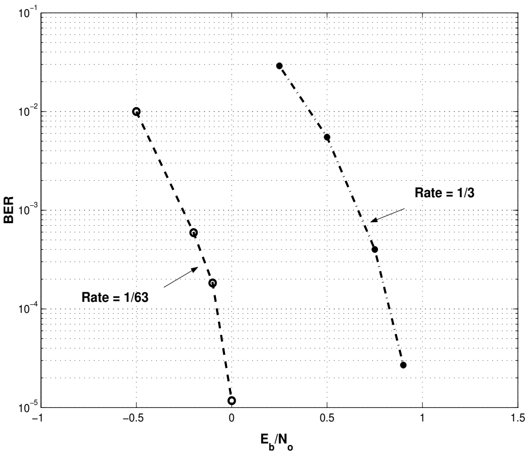

Fig. 7. Performance of Rate-1/3 parent (turbo) code vs Rate-1/63 code in AWGN channel

Figure 3 shows the coding gain achievable using the proposed technique to reduce the code rate from $1/3$ to rate $1/63$ in AWGN channel. Note that an additional coding gain of 0.9 dB is possible. The performance degradation of this code with respect to the corresponding traditional SOTC is evaluated and the results are shown in Figure 8. The degradation is less than 0.1 dB at BER of 10^{-5}. For completeness, we have also included the performance of the best known low rate turbo-like code (THC, code rate 7/370). The result for THC is obtained after performing 50 iterations. Note that there is a degradation of only 0.2 dB. However, the proposed code structure results in about 75% complexity reduction at the decoder compared to the traditional SOTC. In addition, the computational complexity of the proposed scheme per iteration is much lesser than that of THC. Furthermore, the convergence rate of THC is approximately three times slower than that of the proposed scheme. This is an undesired property for hardware implementation.

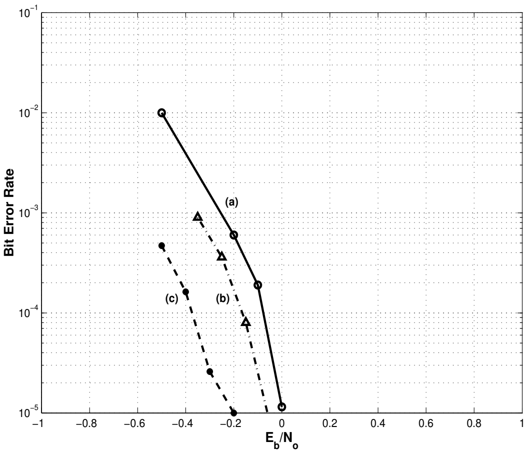

Fig. 8. Performance of various PCCC schemes with rate 1/32 constituent codes. (a) Code constructed using the new algorithm (b) Traditional SOTC (c) Turbo-Hadamard code

No. of signals used	4	8	16	Conventional SOTC
Code rate = 1/16	16	24	-	48
Code rate = 1/32	32	40	-	128
Code rate = 1/64	72	80	96	320

TABLE I

Comparison of number of additions required to compute the branch metric in traditional SOTC and simple SOTC

Here, we compare the complexity of the proposed scheme with that of the traditional SOTC in detail[1]. 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 both 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 the new scheme which require 32 states in total, a complexity reduction of about 75% is achieved. To be precise, in the presence of AWGN, both hard and soft decoding of convolutional codes that use bi-orthogonal signal set involves the calculation of correlation between the received signal with the bi-orthogonal signal set. The complexity of the proposed scheme and the traditional SOTC scheme is compared based on the total number of operations required to calculate this correlation since it has direct influence on the total number of operations required to decode an information bit. One advantage of using Hadamard matrix (of size $2^k \times 2^k$) to generate the bi-orthogonal signal set is that it has a nice property that the correlation can be calculated in $2^k \times \log_2(2^k) = k \times 2^k$ operations. The total number of additions involved (neglecting negations) in order to calculate the branch metric is given in Table 1 for both the traditional SOTC and the proposed code. For a given code rate r, the table gives the number of computations needed by the proposed coding

[1] The complexity analysis of THC is not considered in this work.

scheme if n signals are used from the complete bi-orthogonal signal set. The last column corresponds to using all the signals in the signal set i.e., the computational complexity of the traditional SOTC. Note that a tremendous amount of complexity reduction is achievable with the new coding scheme.

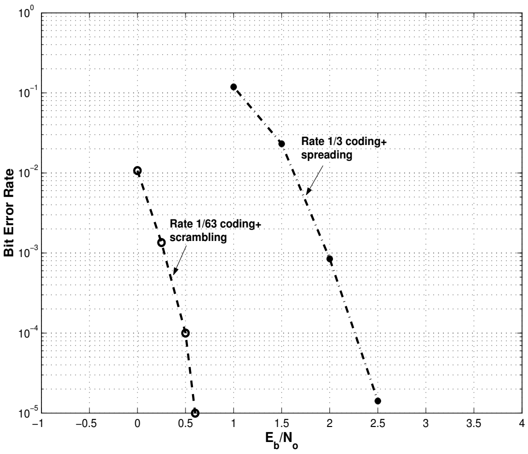

Fig. 9. Performance of Rate-1/3 parent code vs Rate-1/63 code in flat Rayleigh fading channel.

Simulation results for Rayleigh fading channel are shown in Figure 4. In this case, an additional channel interleaver is also included. The fading amplitude is kept constant over a block of 63 coded symbols and independent among blocks. Simulation results show that an additional coding gain of 2 dB can be achieved with the application of low rate coding. Note from Figure 5 that in a heavily loaded system, with single user detectors, a significant reduction in E_b/N_o required to maintain single user performance is possible by using the low rate coded scheme[2] instead of conventionally coded (rate 1/3 turbo code) and spread scheme. Multiple access interference is modelled as Gaussian. Due to complexity, simulations were carried out only for a maximum of 8 users.

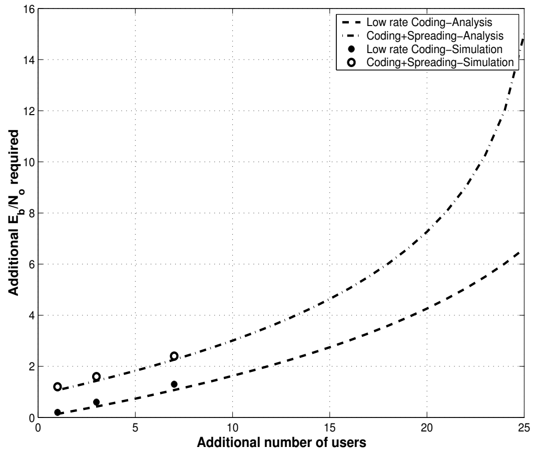

Fig. 10. Additional E_b/N_o (in dB) required to maintain single user performance in multiple access channel

VI. 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 multiple access systems. Apart from being rate compatible for various rates, these codes allow for low complexity encoder and decoder implementation.

References

[1] S. Dolinar, D. Divsalar, and F. Pollara, " Code Performance as a function of Block Size," TMO Progress Report 42-133.

[2] 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.

[3] L. Ping , W. K. Leung and K. Y. Wu "Low rate Turbo-Hadamard Codes," in *Proc. ISIT 2001*.

[4] P. Komulainen and K. Pehkonen, "Performance evaluation of superorthogonal turbo codes in AWGN and flat Rayleigh fading channels," *IEEE JSAC*, vol.16, no.2 ,pp.196-205, Feb. 1998.

[5] D. Thirupathi and K. M. Chugg, "A simple construction of low rate convolutional codes with application to low rate turbo-like code design," accepted for publication in *Globecom* 2002.

[6] P. D. Papadimititiou 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.

[7] A. J. Viterbi, "Very low rate codes for maximum theoretical performance of spread spectrum multiple-access channels," *IEEE J. Select. Areas Commn.*, vol. 8, pp. 641-649, May 1990.

[8] K. M. Chugg, A. Anastasopoulos, and X. Chen, *Iterative Detection: Adaptivity, Complexity Reduction, and Applications*, Kluwer Academic Publishers, MA, 2001.

[9] 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.

[10] A. Viterbi, *CDMA: Principles of Spread Spectrum Communication*, Addison-Wesley, Reading, USA, 1995.

[11] 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.

[2] In this case, users are separated by scrambling sequences