Construction of coset-based low rate convolutional codes and their 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
{thirupat,chugg}@usc.edu

***Abstract* — In this paper, we propose a novel method to construct low rate recursive convolutional codes. The 'overall' convolutional code has a block code and a simple recursive convolutional code as building blocks. The novelty of this type of convolutional code is that it uses coset leaders in order to distinguish the signals that originate from different states. Several of these codes are then used to construct low rate turbo-like codes. Apart from being bandwidth efficient, simulation results show that these codes outperform the best known low rate turbo-like codes, the turbo-Hadamard codes (THC), in additive white Gaussian noise (AWGN) channel for interleaver sizes of practical interest.**

I. Introduction

A considerable amount of research has been done in the area of low rate error correcting codes [6,7,9,10,15]. Such codes are of special interest among researchers working in wide band communications such as ultra-wide band systems, that are extremely power limited. One of the main reasons is that the application of very low rate error control codes in such systems results in added coding gain with no additional penalty [14]. The additional coding gain can translate into added capacity in multi-user spread spectrum schemes.

After the discovery of turbo-codes [1], there has been enormous research effort in designing concatenated coding schemes. This research can be divided into two major categories. The first category is the design of concatenated codes for practical applications where fast convergence for smaller input block sizes are required [8]. Strong constituent codes are used in this design strategy. Design of good concatenated codes for maximum asymptotic coding gain forms the second category [4]. The latter design methodology usually involves weaker constituent codes, slower convergence rate and larger input block sizes.

In the recent past, very low rate turbo-like codes such as super-orthogonal turbo codes (SOTC) [2], turbo-Hadamard codes (THC) [3] etc., were introduced. SOTC belongs to the former category whereas THC belongs to the latter one. In other words, SOTC uses two strong constituent codes and has a fast convergence while THC uses several 'weaker' constituent codes and converges slowly. However, THC was shown to outperform SOTC by about 0.4 dB *asymptotically*, making it the best known low rate concatenated code so far.

In this work, a novel technique to construct low rate turbo-like codes that is aimed at achieving maximum asymptotic gain for a given input block size is introduced. The turbo-like code consists of several weak constituent codes and has slow convergence rate. The new coding scheme thus falls under the second category and it is fair to compare its performance with that of THC. Simulation results show that, in AWGN channel, the proposed coding scheme outperforms THC by about 0.2 dB for interleaver sizes of practical interest.

The rest of the paper is organized as follows. A brief review on low rate concatenated coding schemes is given in Section II. Section III illustrates the proposed code construction methodology. Numerical results and discussion are given in Sections IV and V, respectively.

II. Brief review on low rate concatenated coding schemes

Low rate parallel concatenated coding schemes known so far can be divided into two major classes: parallel concatenation of strong constituent codes (usually two codes) and parallel concatenation of relatively weaker codes (usually more than two codes). The SOTC and THC belong to the first and second categories, respectively. 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. Code words of the block code are used as output signals along the trellis transitions.

SOTC is a parallel concatenation of two super-orthogonal recursive convolutional codes. Super-orthogonal convolutional codes are characterized by the trellis structure such that signals on the trellis transitions from and to any given state are pairwise antipodal. This type of encoder takes in one information bit at a time to give an output of 2^k coded bits. Due to the special structure of the constituent codes[1], rest of the $(k+1)$ bits needed to access the 2^{k+1} bi-orthogonal signal set have to come from the state information, requiring the constituent encoder to have at least 2^{k+1} states. Thus, the number of states increases exponentially with decrease in code rate which is one of the main draw backs of SOTC.

The schematic of the constituent encoder of the THC is shown in Figure 1. Multiple information bits, say k bits are fed into each encoder of THC. Therefore, in order for a one-to-one

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.

[1] Reference [5] and the references therein gives a brief review on super-orthogonal convolutional codes

mapping on to the bi-orthogonal signal set, it is enough to have a 2-state finite state machine. The convolutional encoder used is typically a 2 or 4-state recursive encoder. Due to this fact, the resultant trellis structure has multiple transitions between any two states. More than two such encoders are connected in parallel to form the THC.

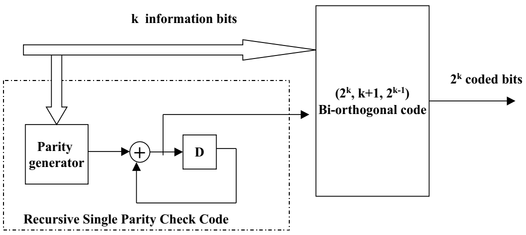

Fig. 1. Schematic of the constituent code of the turbo-Hadamard code

In this work, we construct a new class of low rate codes based on the structure shown in Figure 1. The key contribution of our work is as follows: Note that the input size of the binary block code is $(k+1)$ bits. However, the constituent codes of the THC use only k information bits while the $(k+1)$-th input to the block code comes from a recursive encoder which encodes the parity bit of the other k inputs to the block code. Here, we show that all the $(k+1)$ inputs to the block code can be taken from the input information directly and still an overall recursive convolutional code can be built. This operation increases the code rate for a fixed output length, i.e., instead of rate $k/2^k$ as in THC, the new code rate would be $(k+1)/2^k$. We also show that such codes, when used in parallel concatenation, gives the best known performance (measured both in bit error rate and frame error rate) in AWGN channel.

III. Code construction

A. Constituent Codes

First, we will elaborate on the construction of the coset-based recursive convolutional codes (CB-RCC). The schematic of the structure is given in Figure 2. The input information is divided into blocks of $(k+1)$ data bits. Each block is encoded using a $(2^k, k+1, 2^{k-1})$ bi-orthogonal block code. At the same time, a parity bit is generated for each block. The parity bit is encoded by a simple 2-state, rate-1, recursive convolutional code[2]. The recursive encoding of the single parity check bit ensures that the transitions between even states correspond to inputs with even weight and those between odd states correspond to inputs with odd weight. The current state of the finite state machine is used to select a coset leader of length 2^k (a 1-1 mapping between coset leaders and the state). An element-by-element multiplication is performed on the output of the block code with the elements of the coset leader before being transmitted.

The reason for using coset leaders is straightforward. The total number of branches leaving any state in the overall code is 2^{k+1}. However, there are only 2^{k+1} distinct signals in the $(2^k, k+1, 2^{k-1})$ bi-orthogonal signal set. So, in order to distinguish the signals that leave any state from that of any other state, it is imperative to use some sort of coset leader, the selection of which can be made dependent on the state information. Such construction results in an overall recursive convolutional code with 2^k parallel branches between any two states. The parallel branches in the overall code are the result of the block code. A section of the parent trellis and the trellis of the overall code are given in Figure 3. The label on the branches of the parent trellis represents the input information bit and the corresponding coded output bit.

[2]This can be extended to convolutional codes of arbitrary rate/states.

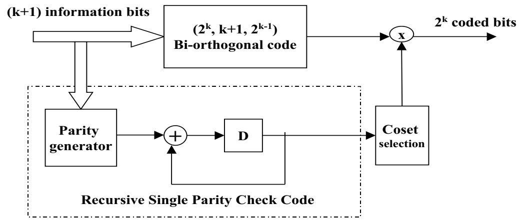

Fig. 2. Schematic of the coset based overall recursive convolutional code

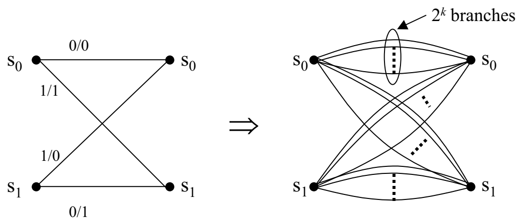

Fig. 3. A section of the trellis of the parent recursive convolutional code and the overall recursive convolutional code

B. Coset Leaders

One of the important issues in designing any coset based code is the selection of coset leaders. In this section, we present an argument on how to choose coset leaders when bi-orthogonal convolutional codes are used. The argument is based on maximum likelihood sequence detection (MLSD) applied to the trellis that corresponds to the overall convolutional code. Though similar arguments can be found scattered in the literature in various forms [10], [18], [19], we have included our own interpretation in presenting the argument for this work to be self contained.

Let the number of states in the convolutional code be N. The received signal corresponding to a single transition of the trellis in AWGN channel can be written as

$$\underline{r} = \underline{c} \odot \underline{\lambda}_m + \underline{n}, \qquad (1)$$

where $\underline{c}$ is a code word from the bi-orthogonal signal set, $\underline{\lambda}_m$ is the coset-leader which is a function of the current state s^m and $\underline{n}$ is the noise vector whose elements are iid, zero mean Gaussian random variables with variance σ^2. The code words are assumed to be modulated using binary phase shift keying (BPSK). Therefore, both $c(j)$ and $\lambda_m(j) \in \{+1, -1\}, j =$

$0, 1, \ldots, 2^{k-1}$ and $m = 0, 1, \ldots, N-1$. The notation $\odot$ is used to represent element-by-element multiplication. The received vector $\underline{r}$ is also a Gaussian random vector of length 2^k, conditioned on the code word $\underline{c}$ and the coset leader $\underline{\lambda}_m$. Figure 4 shows a section of the trellis on which the MLSD is applied.

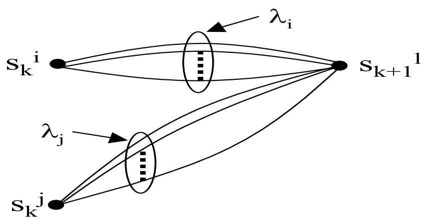

Fig. 4. A section of the trellis on which the add-compare-select procedure is applied

Let $M_0^{k-1}[s_k^i]$ and $M_0^{k-1}[s_k^j]$ be the state metrics associated with s_k^i and s_k^j, respectively, at time instant k. Dropping the superscript on the state variable, the add-compare-select operation that is performed in order to compute the state metric, $M_0^k[s_{k+1}]$, at time $k+1$ can be written as

$$M_0^k[s_{k+1}] = \min_{t_k : s_{k+1}} \left[M_0^{k-1}[s_k] + M_k[t_k] \right], \tag{2}$$

where $M_k[t_k]$ represents the transition or the branch metric. For simplicity let $M_0^{k-1}[s_k]$ be equal for all states at time k. Then

$$M_0^k[s_{k+1}] = \min_{t_k : s_{k+1}} M_k[t_k]. \tag{3}$$

Since all the codewords are equal energy signals, the negative log domain equivalent of the branch metric is simply the negative of the correlation between the received signal with all possible codewords. However, due to the presence of the coset leaders, the received signal has to be pre-multiplied with the coset leader corresponding to the specific state before being correlated. The whole operation can be represented compactly by

$$\begin{aligned} \mathrm{M} &= -\mathrm{H}^T(\underline{\lambda}_n \odot \underline{r}), \\ &= -\mathrm{H}^T(\underline{\lambda}_n \odot \underline{\lambda}_m \odot \underline{c}) - \mathrm{H}^T(\underline{\lambda}_n \odot \underline{n}), \end{aligned} \tag{4}$$

where H is the Hadamard matrix of size $2^k \times 2^k$. Both the operations mentioned above are linear. Therefore, the resulting vector is still a Gaussian random vector whose mean is given by

$$\mathrm{E}\{\mathrm{M}\} = \begin{cases} [0, 0, ..., \mp 2^k, 0, ...0] & \text{if } n = m \\ -\mathrm{H}^T(\underline{\lambda}_n \odot \underline{\lambda}_m \odot \underline{c}) & \text{if } n \neq m. \end{cases} \tag{5}$$

On an average, the branch metric of an incorrect path that corresponds to the same coset to which the transmitted codeword belongs is zero. However, the average branch metric of incorrect paths that corresponds to a different coset is $-\mathrm{H}^T(\underline{\lambda}_n \odot \underline{\lambda}_m \odot \underline{c})$. Let $\underline{y}$ be defined such that

$$\underline{y} = -\mathrm{H}^T(\underline{\lambda}_n \odot \underline{\lambda}_m \odot \underline{c}), \tag{6}$$

where $y(i), i = 0, 1, \ldots, 2^{k-1}$ corresponds to the negative of the correlation between the i-th Walsh-Hadamard (WH) code word and the quantity $\underline{\lambda}_n \odot \underline{\lambda}_m \odot \underline{c}$. Since bi-orthogonal signal set is used, it is straightforward to show that (3) reduces to minimizing the maximum of $|y(i)|$ that can be written as a summation given by,

$$|y(i)| = |\sum_{j=0}^{2^k-1} h_i(j)c(j)\lambda_n(j)\lambda_m(j)|, \tag{7}$$

where $i = 0, 1, ..., 2^{k-1}$. However, bit-by-bit modulo-2 addition of any two WH code words results in another WH code word[3] [18]. Therefore, $h_i(j)c(j)$ can be written as $h_l(j)$ which is the j-th component of another WH code word, i.e.,

$$|y(i)| = |\sum_{j=0}^{2^k-1} h_l(j)\lambda_n(j)\lambda_m(j)|, \tag{8}$$

where $i, l = 0, 1, ..., 2^{k-1}$. Hence, the cost function reduces to

$$\begin{aligned} C &= \arg \min_{\underline{\lambda}_n \odot \underline{\lambda}_m} \{\max_i |y(i)|\}, \\ &= \arg \min_{\underline{\lambda}_n \odot \underline{\lambda}_m} \{\max_i |\underline{h}_i^t(\underline{\lambda}_n \odot \underline{\lambda}_m)|\}. \end{aligned} \tag{9}$$

In other words, the magnitude of the maximum correlation between the product of any two distinct coset leaders and any code word should be minimized. However, since the rows of H are pairwise orthogonal, from Parseval's equality, we have,

$$\begin{aligned} ||\mathrm{H}^T(\underline{\lambda}_n \odot \underline{\lambda}_m)|| &= ||\underline{\lambda}_n \odot \underline{\lambda}_m|| \\ &= 2^k. \end{aligned} \tag{10}$$

The second equality follows from the fact that the elements of both $\underline{\lambda}_n$ and $\underline{\lambda}_m$ are $\pm$ 1. Therefore, (9) is minimized only when the elements of $\mathrm{H}^T(\underline{\lambda}_n \odot \underline{\lambda}_m)$ are ± 1. This property defines a bent sequence [16, pp. 426-428]. In order to maximize the average metric of the incorrect code words, the coset leaders need to be chosen such that the product of any two distinct coset leaders is a bent sequence. It is possible to select the coset leaders to be bent sequences such that their product is also a bent sequence [11].

C. Bent sequences and their construction

Bent sequences are sequences that have constant magnitude spectrum in the Hadamard domain [17]. In this work, we concentrate only on binary bent sequences of length $4^k, k$ an integer. A set of bent sequence vectors of length 4 and the corresponding normalized Hadamard transform coefficients are given in Table 1. We refer to this set as the fundamental set since they can be used to construct bent sequences of other sizes.

[3]Bit-by-bit modulo-2 addition in binary mode is equivalent to bit-by-bit multiplication in bipolar mode

x_1	x_2	x_3	x_4	$\hat{x}_1$	$\hat{x}_2$	$\hat{x}_3$	$\hat{x}_4$
1	1	1	-1	1	1	1	1
1	1	-1	1	1	-1	1	-1
1	-1	1	1	1	1	-1	-1
-1	1	1	1	-1	1	1	-1

Table 1. Length-4 bent sequence vectors and their corresponding normalized Hadamard transform coefficients

There are several methods to construct bent sequence vectors, the details of which can be found in [12] and the references therein. Here, we illustrate the method we have used to construct bent sequence vectors. The method is called bent sequence generation by Kronecker product. Let A and B be two matrices of sizes $u \times v$ and $r \times t$, respectively. Let them be denoted by

$$\mathrm{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1v} \\ a_{21} & a_{22} & \dots & a_{2v} \\ \vdots & \vdots & \ddots & \vdots \\ a_{u1} & a_{u2} & \dots & a_{uv} \end{bmatrix} \qquad (11)$$

and

$$\mathrm{B} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1t} \\ b_{21} & b_{22} & \dots & b_{2t} \\ \vdots & \vdots & \ddots & \vdots \\ b_{r1} & b_{r2} & \dots & b_{rt} \end{bmatrix}. \qquad (12)$$

The Kronecker product of these matrices is given by

$$\mathrm{A} \otimes \mathrm{B} = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1v}B \\ a_{21}B & a_{22}B & \dots & a_{2v}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{u1}B & a_{u2}B & \dots & a_{uv}B \end{bmatrix}, \qquad (13)$$

which is an $(ur) \times (vt)$ matrix. In order to construct a bent sequence of length $4^{i+1}, i$ an integer, the Kronecker product can be applied to any combination of bent sequences in the fundamental set given in Table 1. For example, let the required length of the bent sequence vector z be 64. Then, z can be generated as

$$z = x_i \otimes x_j \otimes x_k, \qquad i, j, k \in \{1, 2, 3, 4\}. \qquad (14)$$

D. Turbo-like code

The codes constructed by the method described in Section III-A are used as constituent codes in parallel concatenation to form a turbo-like encoder. The encoder structure is given in Figure 5. The number of constituent codes is restricted to three due to practical constraints such as decoding delay and complexity. The final codeword consists of the data (systematic bits) and the parity bits from the constituent encoders.

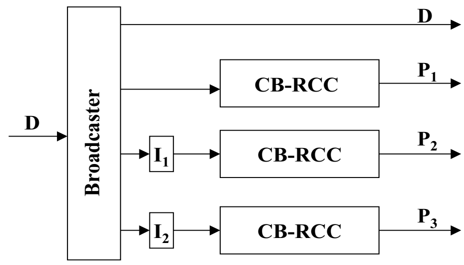

Fig. 5. Turbo-like encoder with coset-based recursive convolutional codes as constituent codes

E. Decoder

A block diagram of the decoder structure that corresponds to the proposed encoder is shown in Figure 6. The soft-in soft-out (SISO) module includes the soft inverse of both the Hadamard code and the 'fundamental' recursive convolutional code. After being compensated for puncturing, the received signal is demultiplexed and fed into the respective SISO modules. In every section of the trellis, the received signal is pre-multiplied by the coset leader that corresponds to the given state before the branch metric is calculated for those branches that leave that particular state. The rest of the decoding process is as given in [3,13].

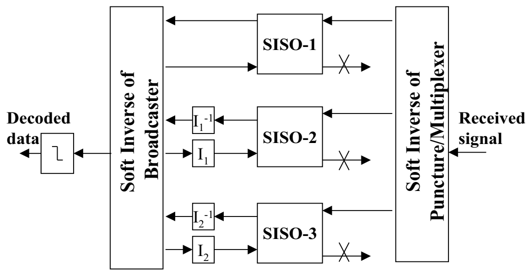

Fig. 6. Decoder structure of the proposed coset based turbo encoder

IV. Numerical Results

Simulation results for the proposed code in additive white Gaussian noise channel are shown in Figures 7 and 8. The over-all code rate is fixed to be 7/180. A-posteriori probability based decoding algorithm is used and the results are presented after performing 50 iterations. In order to keep the encoding and decoding complexity minimal, only 2-state constituent encoders are used. So, only two distinct coset leaders need to be constructed. The generating polynomial of the parent code is $1/(1 + D)$. The performance of the proposed coding scheme is compared against that of rate $6/180$ THC since it is the closest possible code rate attainable in THC. Two different interleaver sizes that are of practical interest are considered. The sizes are chosen such that the same interleavers can be used for both the coding schemes in order to make a fair comparison. Figure 7 shows the bit and frame error rate of the proposed scheme for an interleaver size of 210 bits. Note that the coset based turbo code outperforms the THC both in bit error rate (BER) and frame

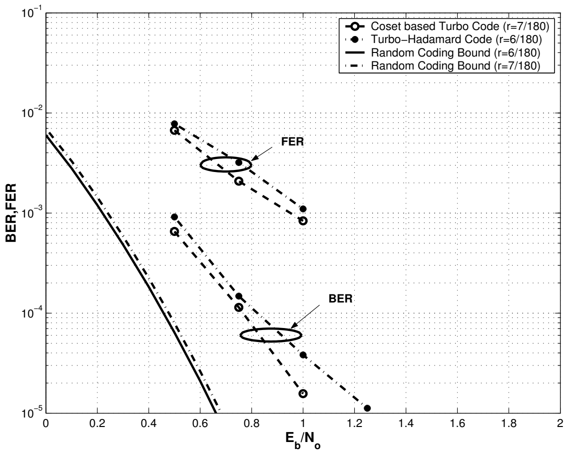

Fig. 7. Performance of coset-based turbo codes and turbo-Hadamard codes for 210 bit interleaver size

error rate (FER). The actual additional coding gain is approximately 0.2 dB in BER and 0.1 dB in FER, respectively. Simulation results for 1050 bit interleaver are shown in Figure 8. In this case, an additional coding gain of about 0.2 dB in both BER and FER is achievable by the coset based encoder. For comparison, the performance of rate 7/370 THC is also plotted. Note that the performance of rate 7/180 coset based code is as good as the rate 7/370 THC. This suggests that with the coset based turbo codes, the throughput can be increased by a factor of two with respect to that of the THC without altering the performance. In addition, random coding bounds for different code rates and interleaver sizes are also included in the corresponding figures.

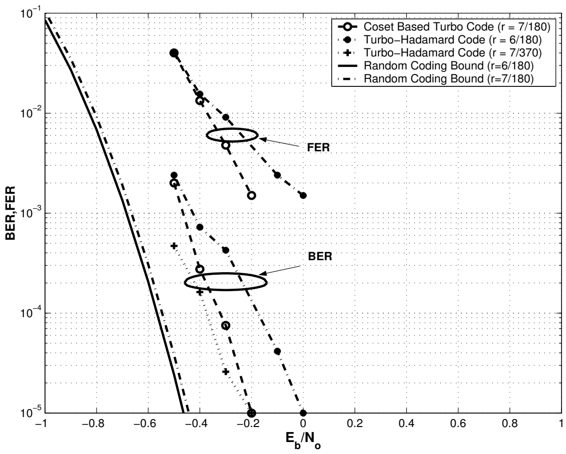

Fig. 8. Performance of coset-based turbo codes and turbo-Hadamard codes for 1050 bit interleaver size

V. Conclusions and Caveats

We have proposed a novel method to construct low rate recursive convolutional codes based on coset encoding with bent sequences as coset leaders. These codes in turn are used in designing low rate turbo-like codes. The turbo-codes constructed using this methodology outperform the best known low rate turbo-like codes by about 0.2 dB in AWGN channel. An important point to note is that the selection of coset leaders for the constituent codes is based on MLSD applied on the trellis that corresponds to the constituent codes and not on that of the concatenated code. Future work along this direction would be to derive coset leaders for the constituent codes based on the application of ML decoding on the hyper-trellis that corresponds to the concatenated code. Also, the coding scheme described in this work is an attempt to design the 'best possible' code that achieves maximum coding gain and not a 'practical' code that has fast convergence. For applications that require practical low rate concatenated codes that have faster convergence, codes in [2] and [5] seem to be suitable candidates.

References

[1] 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.

[2] 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.

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

[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] D. Thirupathi and K. M. Chugg, "A simple construction of low rate convolutional codes with application to low rate turbo-like code design," in *Proc. 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] C. F. Leanderson, J. Hokfet, O. Edfors and T. Maseng, "On the design of low rate turbo codes," in *Proc. VTC 2000.*

[8] 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.

[9] 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.

[10] 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.

[11] G. G. Bottomley, "Signature sequence selection in a CDMA system with orthogonal coding, " *IEEE Trans. on Veh. Tech.*, vol. 42, no. 1, Feb. 1993.

[12] R. Yarlagadda and J. J. Hershey, "Analysis and syntehsis of bent sequences, " *IEE Proceedings*, vol. 136, Pt. E, no. 2, Mar. 1989.

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

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

[15] 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.

[16] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes*, Amsterdam, The Netherlands: North Holland Mathematical Library, 1977.

[17] O. S. Rothaus, "On bent functions," *J. Combn. Theory*, 20, pp. 300-305, 1976.

[18] K. G. Beauchamp, *Applications of Walsh and Related Functions*, Academic Press, New York, 1984.

[19] H. E. Rowe, "Bounds on the number of signals with restricted cross correlation, " *IEEE Trans. on Comm.*, vol-COM 30, no. 5, pp. 966-974, May 1982.