ISSN 0735-2727, Radioelectronics and Communications Systems, 2012, Vol. 55, No. 8, pp. 349–359. © Allerton Press, Inc., 2012.
Original Russian Text © K.H. Prashantha, U.K. Vineeth, U. Sripati, Sh.K. Rajesh, 2012, published in Izv. Vyssh. Uchebn. Zaved., Radioelektron., 2012, Vol. 55, No. 8, pp. 15–29.

Performance Analysis of Stack Decoding on Block Coded Modulation Schemes Using Tree Diagram

K. H. Prashantha, U. K. Vineeth, U. Sripati, and Sh. K. Rajesh

National Institute of Technology Karnataka, Surathkal, India

Received in final form November 25, 2011

Abstract—The channel encoder adds redundancy in a structured way to provide error control capability. Modulator converts the symbol sequences from the channel encoder into waveforms which are then transmitted over the channel. Usually channel coder and modulator are implemented independently one after the other. But in a band limited channel better coding gains without sacrificing signal power are achieved when coding is combined with modulation. Block Coded Modulation (BCM) is such a scheme that results from the combination of linear block codes and modulation. In this paper we are proposing a stack decoding of rate 2/3 and rate 1/2 BCM schemes using tree structure and performance is compared with the Viterbi decoding that uses trellis representation. Simulation result shows that at reasonable bit error rate stack decoder performance is just 0.2 to 0.5 dB inferior to that of Viterbi decoding. Since stack decoding is a near optimum decoding scheme and whose decoding procedure is adaptable to noise level, we can consider this method in place of Viterbi decoding which is optimum and its decoding complexity grows exponentially with large code lengths.

DOI: 10.3103/S073527271208002X

INTRODUCTION

When conventional coding techniques are introduced in a transmission system, the bandwidth of the coded signal after modulation is wider than that of the uncoded signal for the same information rate and the same modulation scheme. In fact, the encoding process requires a bandwidth expansion that is inversely proportional to the code rate (k / n), being traded for a coding gain. This is the reason why, conventional coding schemes have been very popular on power-limited channels (where bandwidth is not a constraint) but not on bandwidth-limited channels [1].

There is another class of encoder that treats convolutional and block codes as their component codes and combines modulation and coding as a single entity. These modulation schemes accommodate the redundancy of a code on an expanded signal set, and achieve excellent coding gain over uncoded modulation with no bandwidth expansion. It is well known fact that power and bandwidth can be traded against each other. But with this class of encoders we can significantly decrease the power, without having to increase the bandwidth for the same performance. There are two classes of encoders—Trellis Coded Modulation (TCM) and Block Coded Modulation (BCM). There is hardly any difference between the two schemes in terms of performance but the former is well known in the community.

ENCODING

There are many encoding schemes available for generating BCM. Multilevel block codes are one of the ways of generating compact codes. A general block diagram of a binary BCM encoder is shown in Fig. 1 [2].

The encoding process takes m parallel messages $\mathbf{u}^{(1)}$, $\mathbf{u}^{(2)}$, …, $\mathbf{u}^{(m)}$ of lengths $k_1, k_2, \ldots, k_m$ bits and each message is encoded by one of m parallel block encoders, resulting in m-codewords all of equal length n bits with minimum Hamming distances $d_1, d_2, \ldots, d_m$. This arrangement of block encoders is called multi-level block coding and was proposed by Imai and Hirakawa [3]. Finally, each bit from the m parallel encoders is mapped onto a 2^m-ary constellation. The overall code rate R of the multi-level block code is:

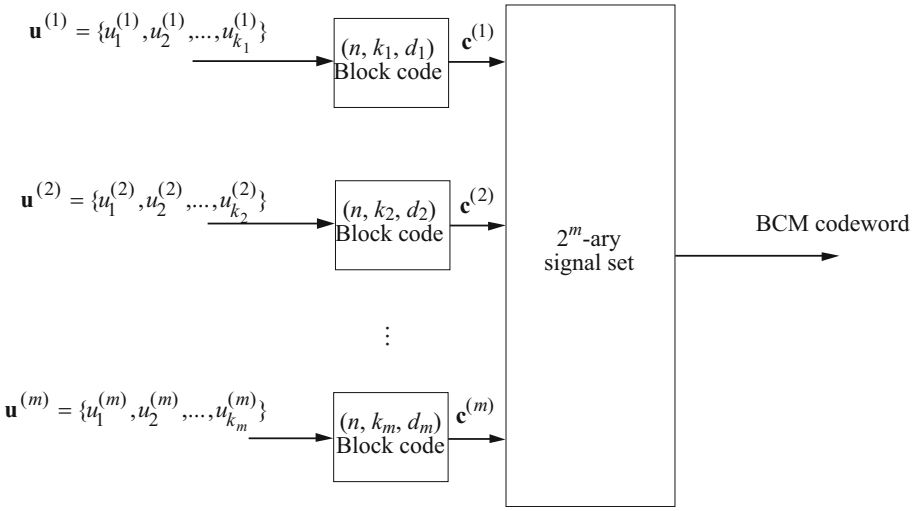

Fig. 1.

$$R = \frac{\sum_{i=1}^{m} k_i}{mn}. \tag{1}$$

An example of a BCM code using $m = 3$ block encoders which are mapped onto a $2^3 = 8$-PSK constellation is given in Fig. 2.

The top block encoder is an (8,1,8) repetition code, which takes a message of $k_1 = 1$ bit and repeats that value seven times, for example $0 \rightarrow (00000000)$ and $1 \rightarrow (11111111)$. The second block code is the (8,7,2) even parity check code. It takes a seven bit message and appends a parity check bit, which is either zero if the number of '1's in the message is even or one if the number of '1's in the message is odd. Finally, the last one is the universal code that consists of all the 8-tuples over F_2. From Fig. 2 it is evident that the code rate of this BCM code is $R = \frac{1+7+8}{3\times 8} = 2/3$. All that remains is to map each bit from the three encoders to the 8-PSK constellation.

SIGNAL SET PARTITIONING

The output of BCM encoder is a constrained sequence of constellation symbols. This mapping should ensure the maximization of the free distance d_{free} or minimum Euclidean distance Δ between the BCM code symbols. To achieve this constraint, Ungerboeck [4, 5] proposed the set partitioning of the constellation, where the points in the constellation are divided recursively into subsets, with the Euclidean distance between neighboring points in each subset being increased. Also he postulated that the signals in the bottom level of the partition tree are assigned parallel transitions and these parallel transitions must appear in subsets separated by the largest Euclidean distance. Figure 3 shows set partitioning of the 8-PSK constellation.

In [6] it is shown that the squared free Euclidean distance between nearest sequences of the BCM code signal is

$$d^2_{\text{free,coded}} = \min(\Delta_1^2 d_1, \Delta_2^2 d_2, \Delta_3^2 d_3, \ldots, \Delta_m^2 d_m), \tag{2}$$

where $\Delta_i^2, i = 0,1,2,\ldots,m$ is the squared Euclidean distance between neighboring points in a subset in the ith level of partition chain and d_i is the minimum Hamming distance of the ith block code of the BCM code. The quantity γ called asymptotic coding gain (ACG) is used to measure the performance of BCM code over an uncoded constellation [7]. It is defined as

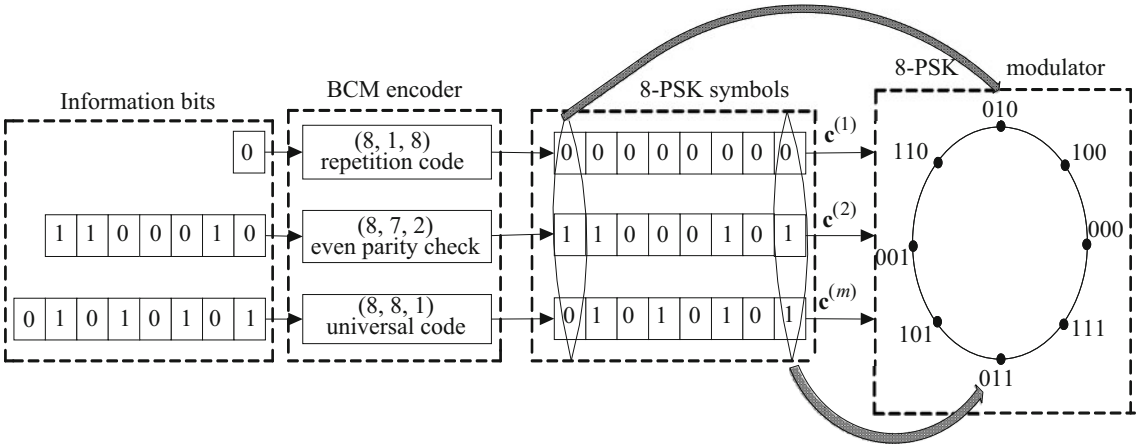

Fig. 2.

$$\gamma = \left(\frac{d^2_{\text{free,coded}}}{d^2_{\text{free,uncoded}}} \right) \left(\frac{\varepsilon'}{\varepsilon} \right), \tag{3}$$

where $d_{\text{free,uncoded}}$ is the minimum distance between points in the original signal constellation, ε is the energy of the coded constellation and ε' is the energy of the uncoded constellation. For the BCM scheme in Fig. 2 the minimum Hamming distances are $d_1 = 8$, $d_2 = 2$, and $d_3 = 1$. The respective squared Euclidean distances from the subsets of 8-PSK in Fig. 3 are $\Delta_1^2 = 0.585\varepsilon$, $\Delta_2^2 = 2\varepsilon$, $\Delta_3^2 = 4\varepsilon$. So the squared free distance of the BCM code from (2) is

$$d^2_{\text{free,coded}} = \min\{4.68\varepsilon, 4\varepsilon, 4\varepsilon\} = 4\varepsilon. \tag{4}$$

This increase in free distance is responsible for providing the much needed coding gain. The uncoded QPSK modulation and BCM coded 8-PSK modulation have the same average energy, i.e. $\varepsilon = \varepsilon'$. The minimum squared Euclidean distance between neighboring points of QPSK constellation is $d^2_{\text{free,uncoded}} = 2\varepsilon'$, giving us asymptotic coding gain of $\gamma = \frac{4\varepsilon}{2\varepsilon'} = 2$ or 3 dB.

TRELLIS AND TREE REPRESENTATION OF BCM SCHEMES

Significant coding gain is achieved over uncoded modulation by imposing suitable tree or trellis structure to the block coded modulation scheme and decoding using stack or Viterbi algorithm. It is possible to draw a trellis diagram for block codes [8]. For this generator/parity check matrix has to be reduced to trellis oriented form. Cartesian product of the individual block code trellises results in trellis of the over all BCM scheme [6, 9].

The state transitions in the individual block code's trellis is determined by the state transition equation given as

$$\sigma_{t+1} = \sigma_t + \alpha h_{t+1}, \ \forall \alpha \in F_2, \tag{5}$$

where σ_t is the present state, σ_{t+1} is the next state of the trellis and h_{t+1} is the $(t+1)$th column of parity check matrix H. This particular method of trellis construction is called syndrome trellis and syndrome trellis of a linear code is minimal [10]. The parity check matrices of the (8,1,8) repetition code H_{rep} and the (8,7,2) even parity check code H_{par} are given below

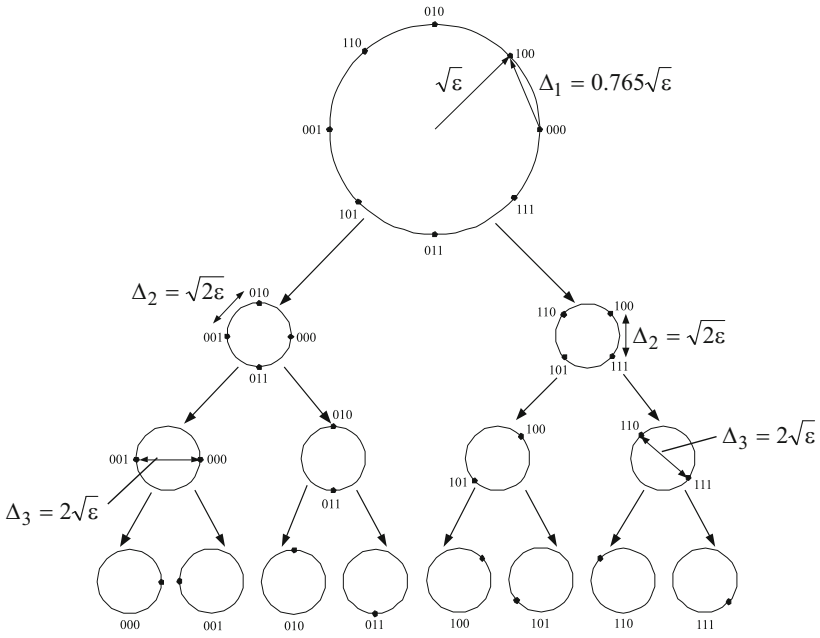

Fig. 3.

$$H_{\text{rep}} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$H_{\text{par}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Figures 4–6 show the minimal trellis diagram for the (8,1,8) repetition code, (8,7,2) even parity check code and (8,8,1) universal code. It is readily observed that when the construction is complete all trellis paths that do not terminate at all zero state are removed as they do not represent valid code sequences.

The overall trellis for the BCM scheme is constructed by using the individual trellises of every component code. Taking the Cartesian product $c^{(1)} * c^{(2)} * c^{(3)}$ of these three minimal trellises gives us eight-section Cartesian product trellis for the (8,1,8), (8,7,2) and (8,8,1) BCM scheme i.e. the trellis of the interleaved code as shown in Fig. 7 [6].

From trellis structure we can derive the tree diagram. The code tree can be treated as an expanded version of the trellis, where every path is totally distinct from every other path. Tree structure of BCM scheme facilitates the stack decoding which is near optimal decoding algorithm as compared with maximum likelihood (ML) or Viterbi decoding which makes use of trellis diagram. Tree representation of above BCM code trellis is shown in Fig. 8. Since there are a large number of legitimate (or possible) code sequences, only few paths are shown in the tree diagram.

We have derived trellis structure for second BCM scheme using (8,4,4) Reed–Muller (RM) code, (8,1,8) repetition code and (8,7,2) even parity check code. This selection of first order Reed–Muller code instead of universal code increases the free distance from 4 to 4.686 and code rate reduces from 2/3 to 1/2. First order RM code has error correcting capability of one bit.

We have taken 8-section trellis diagram for the (8,4) RM code with state labeling by the state-defining information set from [6]. This is shown in Fig. 9. This trellis is derived from code's generator matrix which is

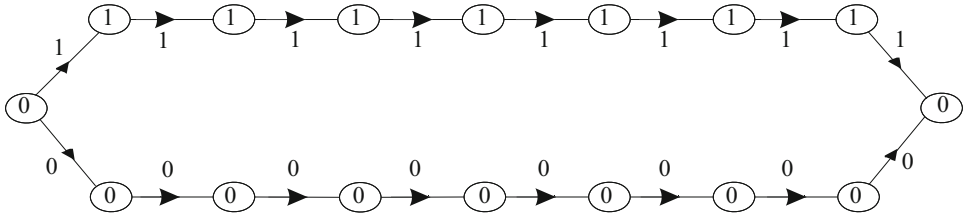

Fig. 4.

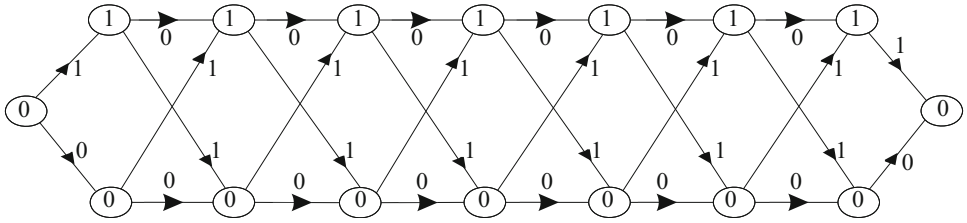

Fig. 5.

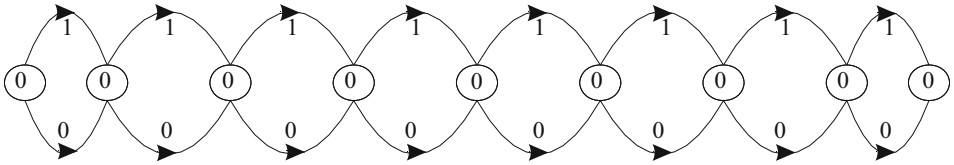

Fig. 6.

arranged in the trellis oriented generator matrix (TOGM) form. Any generator matrix of the code can be brought into trellis oriented form by performing the Gaussian elimination or elementary row operations over original matrix. Trellis oriented generator matrix for (8,4) first order RM code is given below

$$G_{\text{TOGM}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

This is a time variant trellis. All valid sixteen codewords can be traced from initial all zero state to final all zero state. Figure 10 shows the trellis resulting after Cartesian product of first order RM code with the (8,1,8) repetition code.

Taking Cartesian product of intermediate trellis shown in Fig. 10 with the trellis of (8,7,2) even parity check code resulting in final BCM trellis as shown in Fig. 11.

Unwrapping the above trellis gives us the tree diagram shown in Fig. 12 and this can be used for decoding of received codeword adopting stack algorithm. We have listed a few legitimate paths (valid codeword sequences) only.

PERFORMANCE ANALYSIS USING VITERBI AND STACK DECODING

BCM schemes can be represented by both a trellis and tree representation. Thus decoding involves either ML based Viterbi algorithm using trellis representation or near optimal decoding based on stack algorithm adopting tree diagram. The first decoding technique even though optimum is unable to achieve low probability of error at data rates close to channel capacity. This is because the decoding effort (number of computations) is fixed for a particular code no matter what the received sequence is. This is particularly a

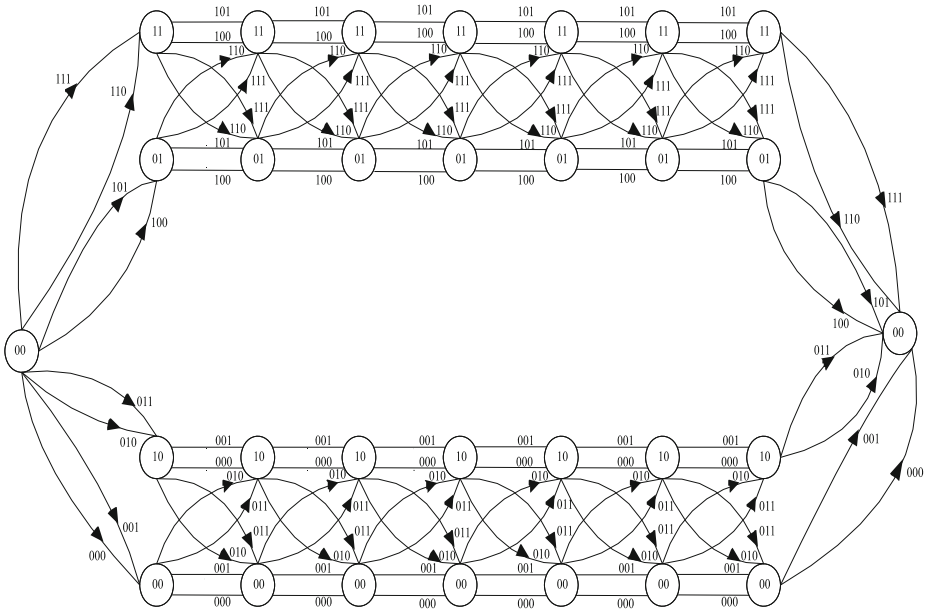

Fig. 7.

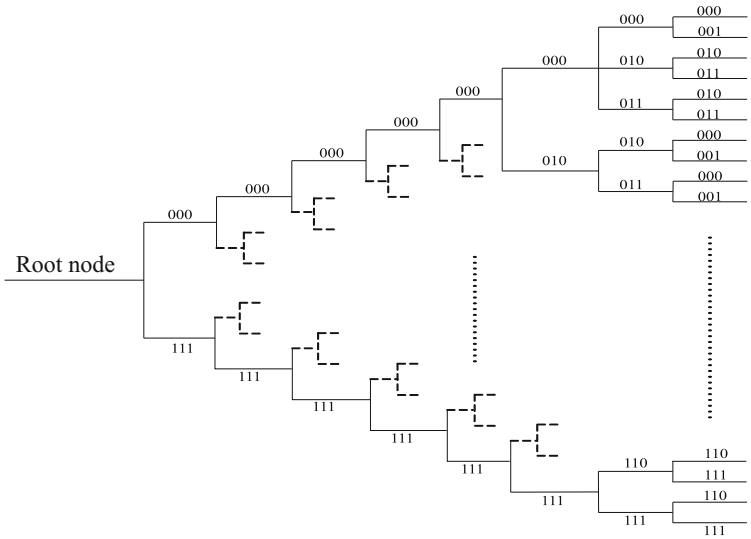

Fig. 8.

disadvantage when the noise is light (or high SNR). Also the complexity of decoding grows exponentially with large code lengths [11].

It is under these circumstances that sequential decoding is of use, a decoding procedure whose effort is adaptable to the noise level. The purpose of stack algorithm is to search through the nodes of the code tree (an expansion of the trellis), that is, without having to examine too many nodes, in an attempt to find the maximum likelihood path. An ordered list or stack of previously examined paths with different lengths is kept in storage. Each stack entry contains a path along with its metric (the metric is a measure of the "closeness" of a path to the received sequence), the path with the largest metric is placed on top, and the others are listed in order of decreasing metric. Each decoding steps consists of extending the top path in the stack by computing the Fano metrics of its succeeding branches and then adding these to the metric of the top path to form new paths, called the successors of the top path. The top path is then deleted from the stack, its successors are inserted and the stack is rearranged in order of decreasing metric values. When the top path in the stack has highest Fano metric and also it is the end of tree, the algorithm terminates [12].

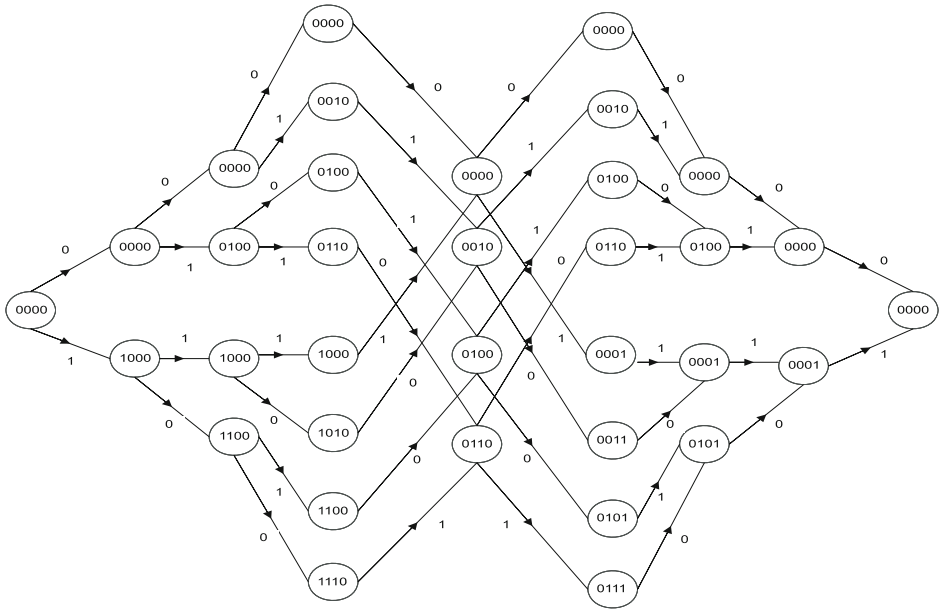

Fig. 9.

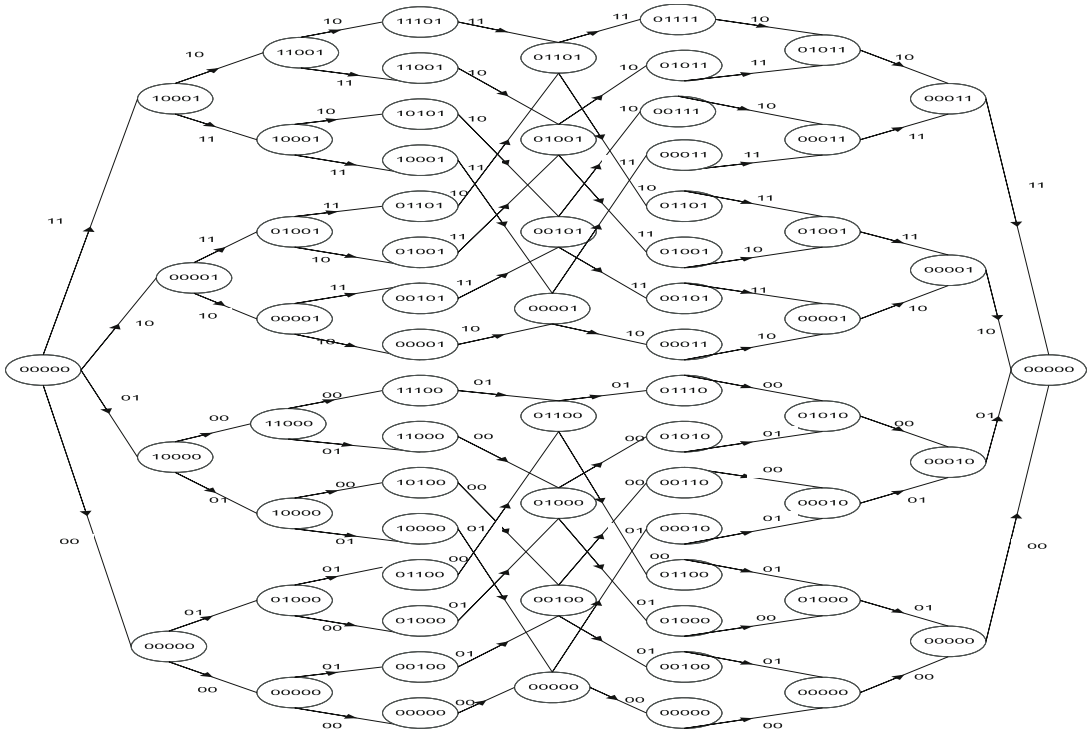

Fig. 10.

The Viterbi algorithm for each received vector **r** starts with an initial state which is commonly the all-zero state. If we arbitrarily choose a node in the trellis diagram of a BCM code and look at all of the paths going into it, we can always observe a path that has a smaller distance between the received sequence and the code sequence than all other paths. Viterbi noticed that the paths that are not optimal now can never be optimal in the future. Thus, Viterbi algorithm chooses a branch that belongs to a path with the smallest distance. This retained path is called the survivor path. All other paths at each state with higher distances are

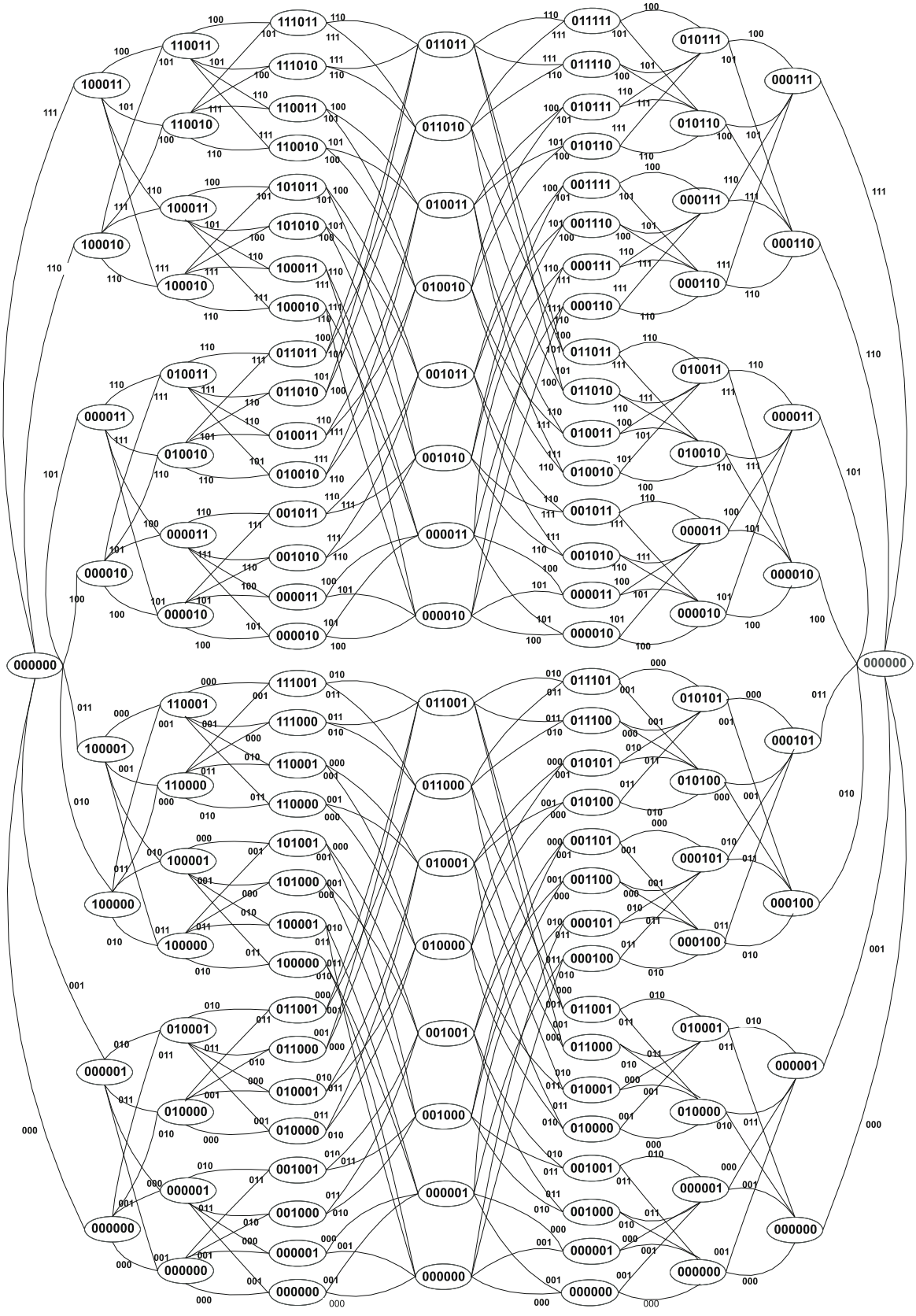
000000
111011
110011
100011
110010
111010
110011
110010
101011
100011
101010
100011
100010
100010
011011
010011
000011
010010
010010
001011
000011
001010
000010
000011
000010
000010
011111
010111
011110
010111
010110
010110
001111
000111
001110
000111
000110
000110
011011
010011
011010
010011
010010
010010
001011
000011
001010
000011
000010
000010
111001
110001
111000
110001
110000
110000
101001
100001
101000
100001
100000
100000
011001
010001
011000
010001
010000
010000
001001
000001
001000
000001
000000
000000
011101
010101
011100
010101
010100
010100
001101
000101
001100
000101
000100
000100
011001
010001
011000
010001
010000
010000
001001
000001
001000
000001
000000
000000
000000

Fig. 11.

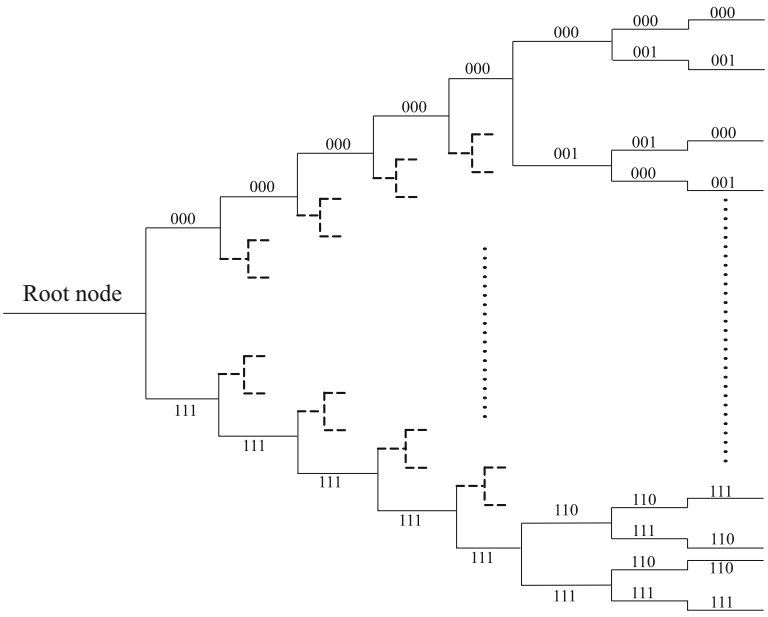
Root node
000
001
111
110

Fig. 12.

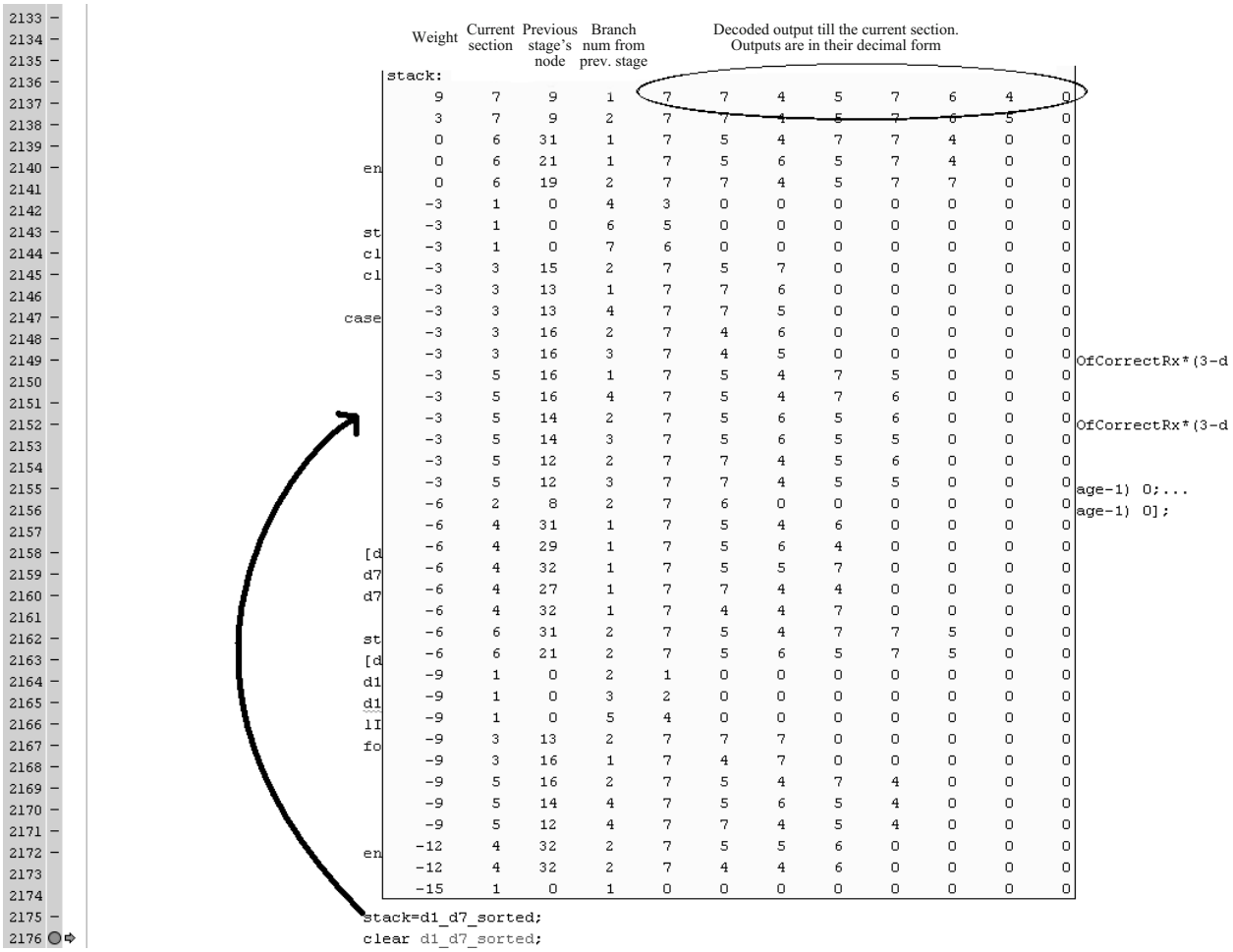
Weight
Current section
Previous stage's node
Branch num from prev. stage
Decoded output till the current section. Outputs are in their decimal form
stack:
stack=d1_d7_sorted;
clear d1_d7_sorted;

Fig. 13.

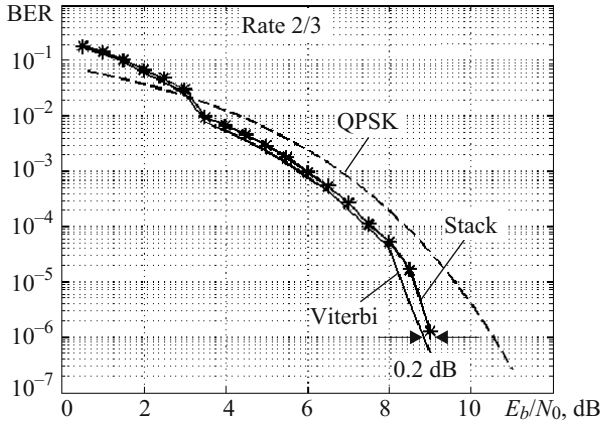

Fig. 14.

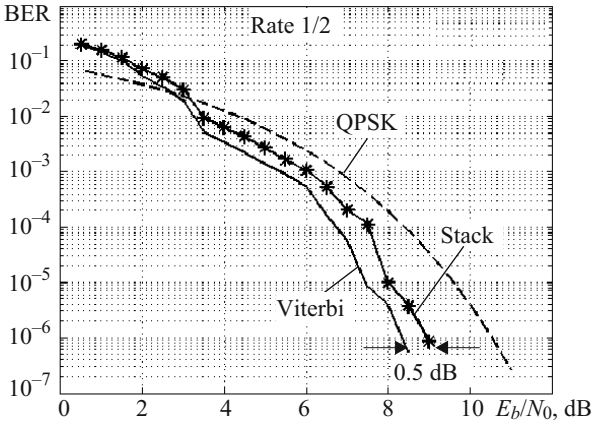

Fig. 15.

discarded. In the case of a tie, a survivor path might be chosen randomly. Updating the path metric in each trellis node involves add, compare (to find the minimum path metric), and select (choose the path with minimum metric as the survivor). The operations form the so called add-compare-select (ACS). At the end of the decoding window, the survivor path with the smallest distance is selected and the associated code vector is chosen as the transmitted codeword [13].

The stack algorithm using associated BCM code tree is summarized below:

1. Initialize the stack with the root of the code tree and set its path metric to zero.

2. Extend the path at the top of stack with highest metric (i.e., the best path) to its successor branches.

3. Store all of the paths and their path metrics in the stack from top to bottom in the order of decreasing metric values.

4. If the top path has reached the end of the code tree, stop the process and output it as the decoded sequence. Otherwise, repeat the procedure from step 2.

The amount of computation involved here depends on the channel condition. If the channel SNR is high, tree searching finishes quickly. Otherwise, more computation is expected.

Figure 13 shows the MATLAB simulated screen shot of stack contents during the decoding process.

We can note that the weights in the first column are in descending order. The second column indicates the label of current tree section. The third column indicates the node numbers of the last compared sections. The index of branch with minimum weight from this node is stored in the fourth column. Remaining eight columns depict the decoded symbols for the current section represented in decimal form.

We have done simulation to analyze the performance of two BCM schemes by using Viterbi and stack algorithm and compared it with uncoded QPSK system. In the first BCM scheme whose rate is 2/3,Viterbi decoding performs 1.8 dB better than uncoded QPSK modulation at bit error rate (BER) of10^{-6}. This coding gain is achieved without bandwidth expansion. Stack decoding performance is just 0.2 dB away from Viterbi decoding for this BER. In our second BCM scheme whose rate is 1/2, Viterbi decoding performs 2.3 dB better than uncoded QPSK modulation at BER of 10^{-6}. This significant improvement in coding gain compare to first scheme is due to greater redundancy and the corresponding increase in free distance. In this second scheme, the performance of stack decoding is inferior to Viterbi decoding by 0.5 dB. The performance of both the schemes is summarized in Figs.14, 15, respectively.

CONCLUSIONS

It is well known fact that coded modulation schemes provide significant coding gain over equivalent uncoded systems over band-limited channels. This is achieved by an intelligent mapping of symbol sequences to constellation points.

Decoding of block coded modulation schemes for getting good bit error rate performance has always been achieved through the use of optimum decoders like Viterbi and BCJR (Bahl–Coke–Jelinek–Raviv) along with exploiting the trellis structure of these schemes. But when the code length becomes large, the use

of ML decoding turns out to be computationally cumbersome because computational complexity increases exponentially with code length in these schemes. This increases the delay in decoding process.

Suboptimal methods such as stack decoding extend the most likely path in their code tree and give fast output if the channel noise is light. Since the number of computations is not fixed like that of Viterbi or BCJR, they give near optimum performance for much less computation. This has the benefit of decreasing the latency.

This stack decoding method can be extended to reasonably long length BCM codes to get near optimal performance at higher SNRs. Thus, BCM schemes with stack decoding can be advantageously used in any application where the channel is characterized by having moderately high SNR but limited bandwidth.

REFERENCES

1. A. G. Burr, "Block versus trellis: An introduction to coded modulation," J. Electronics and Commun. Eng. **5**, No. 4, 240 (Aug. 1993).
2. R. A. Carrasco and M. Johnston, *Non-Binary Error Control Coding for Wireless Communication and Data Storage*, 1st ed. (Wiley, 2009).
3. I. Hideki and Sh. Hirakawa, "A new multilevel coding method using error correcting codes," IEEE Trans. Inf. Theory **23**, No. 3, 371 (May 1977).
4. G. Ungerboeck, "Trellis Coded Modulation with Multilevel/Phase Signals," IEEE Trans. Inf. Theory **IT-28**, 55 (Jan. 1982).
5. G. Ungerboeck, "Trellis Coded Modulation with Redundant Signal Sets. Part I: Introduction," IEEE Commun. Magazine **25**, 5 (Feb. 1987).
6. Sh. Lin and D. Costello, *Error Control Coding*, 2nd ed. (Pearson Education, New Jersey, 2004).
7. T. K. Moon, *Error Correction Coding* (Wiley-Interscience, New York, 2006).
8. J. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," IEEE Trans. Inf. Theory **24**, No. 1, 76 (Jan. 2003).
9. A. Vardy, *Trellis Structure of Codes*, *Handbook of Coding Theory* (Elsevier Science, Amsterdam, 1998) [ed. by Vera Pless, W. Huffman, and R. A. Brualdi].
10. Sh. Lin, Kasami, Fujiwara, and Fossorier, *Trellises and Trellis Based Decoding Algorithms for Linear Block Codes*, 1st ed. (Kluwer Academic, 1998).
11. A. B. Kiely, S. J. Dolinar, R. J. McEliece, L. L. Ekroot, and W. Lin, "Trellis decoding complexity of linear block codes," IEEE Trans. Inf. Theory **42**, No. 6 (Nov. 1996).
12. R. Johannesson and K. Sh. Zigangirov, *Fundamentals of Convolutional Coding* (Universities Press, 2001).
13. A. J. Viterbi and Omura, *Principles of Digital Communication and Coding*, Int. ed. (McGraw-Hill, New York, 1979).