

## Concatenated Tree Codes and Related Schemes<sup>†</sup>

Li Ping, B. M. Bai and W. K. Leung

Department of Electronic Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong  
e-mail: eeliping@cityu.edu.hk

**Abstract** — We report the recent progress in concatenated tree (CT) codes and related schemes, including low rate codes for CDMA applications and bandwidth efficient coded modulation schemes. It is shown that CT codes provide efficient solutions for various situations. Near capacity performances (within about 0.5 dB) can be achieved at significantly reduced decoding costs.

### I. INTRODUCTION

The advent of turbo codes [1] and the re-discovery of LDPC codes [2] have changed the traditional view of the connection between coding performance and decoding complexity.

Concatenated tree (CT) codes [3] represent a further effort towards low complexity coding schemes without compromising performance. This paper aims to report the recent progress in CT codes and related schemes, including low rate codes for CDMA applications and bandwidth efficient coded modulation schemes. We will show that CT codes provide efficient solutions for various applications. Near capacity performances (within about 0.5 dB) can be achieved at significantly reduced decoding costs.

### II. PSEUDO-RANDOM CODES

We start with a heuristic discussion of the basic rationale behind CT codes. For convenience, we will only consider linear codes, although the principles can actually be applied to non-linear codes (such as coded modulations [4]). A common property of turbo codes and LDPC codes is the return to the “random code” concept of Shannon. Consider a generalized turbo-type code consisting of  $M$  component codes concatenated in parallel. Let  $w$  be the total parity weight obtained after concatenating. Then,

$$w = \text{the sum of the parity weights of } M \text{ component codes} \quad (1)$$

Assume that the right hand side of (1) involves (approximately) independent and identically distributed random variables (the i.i.d. assumption). Applying the Law of Large Numbers, for sufficiently large  $M$ ,  $w$  approaches Gaussian. This indicates a close resemblance between such a concatenation scheme and a “random code” [5]. The latter is a well-known “good code”.

The i.i.d. assumption above is conditional. The information and parity weights can be strongly correlated. For instance, a small information weight will produce a small parity weight in a conventional non-recursive convolutional code. This problem has been treated through the use of recursive component codes and random interleaving in turbo codes.

Based on (1), we can establish some basic principles that may potentially lead to good codes.

- (i) Interleavers should be adopted to ensure independence between  $M$  component codes.
- (ii) Recursive component codes should be used.  
(A code is said to be recursive if an indefinite parity weight can be generated from a finite information weight. This property serves to break the correlation between small input weights and small parity weights.)
- (iii) Multiple component codes should be used.  
(The number of component codes need not be too large, as the Law of Large Numbers can be applied to the sum of about four or more random variables.)

Note that, in the above, the encoding method of component codes is only related to the recursive requirement. This can be achieved using some simple codes. It turns out that some good codes, such as those discussed below, can be constructed based on principles (i)-(iii).

### III. CT CODES AND RELATED SCHEMES

The CT code introduced in [3] is the parallel concatenation of multiple component tree codes. A tree code can be represented by a loop-free Bayesian network or a loop-free Tanner graph. The former is convenient for encoding and the latter for decoding. In this paper, we will also consider the generalization of CT-type codes such as low-rate CT codes [6] and CT-TCM codes [4].

#### A. Binary Tree Codes

The Bayesian network representation of a tree code is given in Fig. 1(a). Here, every node represents a bit in the code. An information symbol in the  $k$ th section consists of  $n$  information bits, denoted by  $\mathbf{d}_k = (d_{k,0}, d_{k,1}, \dots, d_{k,n-1})$ . In Fig. 1,  $\mathbf{d}_k$  contains two parts,  $\{\mathbf{d}'_k\}$  and  $\{\mathbf{d}''_k\}$ . A state node at time  $k$ , denoted by  $q_k$ , is given by

$$q_k = q_{k-1} + \sum_{b \in \mathbf{d}'_k} b \quad (2)$$

which is the binary sum (i.e., parity check) of  $(\mathbf{d}'_k, q_{k-1})$ . We call  $\mathbf{c}_k = (\mathbf{d}_k, q_k)$  a coded symbol. The sets  $\{\mathbf{d}'_k\}$  and  $\{\mathbf{d}''_k\}$  are referred to as the recursive and non-recursive part, respectively. Consider a weight-1 input word. If the only nonzero bit falls in the recursive part, it leads to an indefinite number of nonzero parity bits. (Hence the name recursive part.) This is not the case if the only nonzero bit falls in the non-recursive part. The advantage of adopting both recursive and non-recursive parts was discussed in [4].

<sup>†</sup> This work was supported by a CERG grant #9040518.

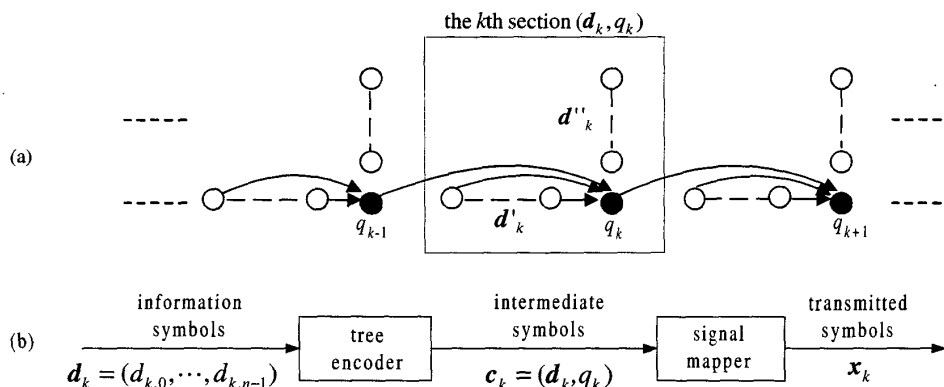


Fig.1. (a) The Bayesian network representation of the underlying tree code. (b) A generalized tree-code-type scheme.

### B. Two-State Trellis Representation of Tree Codes

Tree codes can also be represented by two-state trellises. Since an input symbol comprises  $n$  bits, the trellis diagram may contain a large number of parallel branches. Parallel branches are usually avoided in conventional coding schemes for performance concerns. However, the use of multiple component codes in CT-based schemes can effectively resolve this issue [4][6][7].

### C. Decoding Complexity

The global decoder of a CT code was discussed in [3]. The local decoder can be constructed based on the belief propagation decoding or the BCJR algorithm. These two methods are actually equivalent and result in similar decoding costs.

Here is a brief comparison between the decoding costs of a two-state trellis code and a more general  $S$ -state trellis code using the BCJR algorithm. For a two-state trellis, one of the two state-variables in each trellis section can be normalized to unity. The multiplication operation involving a normalized state variable can be avoided [4]. This technique also applies to a general  $S$ -state trellis, but the relative benefit reduces as the number of states increases. For instance, using the above technique, the decoding complexity of a two-state trellis code is about 16 times (instead of 8 times) lower than that of a 16-state one.

### D. Generalized CT-Code-Based Schemes

We now outline several CT-code-based schemes for different applications. Referring to Fig. 1(b), an underlying tree encoder is used in all situations, producing an intermediate sequence  $\{c_k\}$  that is mapped to the transmitted sequence by a signal mapper detailed below.

#### Moderate to high rate ( $1/4 < \text{rate} < 1$ )

In this case, BPSK or QPSK transmission is assumed. Since the code is systematic, information bits need to be transmitted for only once. The overall rate of the code is thus  $R=n/(n+M)$ , where  $M$  is the number of component codes and

$n$  is the number of bits in  $d_k$ . In the following, we consider two situations depending on whether  $\{d_k''\}$  is empty or not.

When  $\{d_k''\}$  is empty, we can simply let  $x_k=c_k$ . This leads to the so-called zigzag code [8]. Due to the simple structure involved, union bound can be easily calculated for a zigzag code. It has been shown that a zigzag code with  $M>4$  has better error floor performance than a 16-state turbo code. Increasing  $M$  appears a more efficient method to suppress error floor, as opposed to increasing the state number  $S$  of component codes. Decoding cost *increases only linearly* with  $M$  while it *increases exponentially* with  $S$ . As an example, increasing  $M$  from 4 to 5 only raises decoding complexity by a factor of 5/4, but it can reduce error floor significantly [8].

When  $\{d_k''\}$  is not empty, an alternative technique is to let  $x_k=(d_k, p_k)$ , where  $p_k$  is the parity check of  $(d_k'', q_k)$ . (In this way  $q_k$  is punctured.) This leads to the CT codes discussed in [3]. This technique results in improved performance in the waterfall region. In Fig.2, the performance curves of a rate 1/2 CT code are given, which nearly overlap with those of a punctured (37, 23) turbo codes.

#### Low rate (rate $\ll 1$ )

One approach to low rate design is to employ more component codes. However, we observed that this leads to relatively poorer performance compared with low rate turbo codes [9]. An intuitive explanation of this observation is that the local CT decoder based on parity checks, which represent a relatively weak relationship, becomes very unreliable for a very noisy channel. (Recall that, for a fixed energy per information bit, energy per coded-bit is proportional to the coding rate.) We expect the same principle applies to low rate LDPC codes and low rate RA codes [10] for which decoding is also based on local parity checks. It appears that for low rate concatenated codes, it is necessary to resort to "stronger" component codes (e.g. Hadamard codes) to provide more reliable soft information.

Accordingly, we can adopt  $\{x_k\} = \{h\}$ , where  $\{h\}$  is the codeword set in a Hadamard code. The resultant scheme is referred to as a turbo-Hadamard code [6]. A very efficient decoding technique for a turbo-Hadamard code has been

introduced in [6]. The core part is a fast APP decoding algorithm for the Hadamard code. The performance of a rate 1/53 turbo-Hadamard code is shown in Fig. 2, which is within 0.4dB of the ultimate Shannon limit of -1.6dB.

A potential application of such low rate codes is in joint coding and spreading. Traditionally, coding and spreading are separated in CDMA systems (e.g. IS-95). It has been demonstrated that performance can be significantly improved by treating the coding and spreading functions in IS-95 as a concatenated code and applying an iterative decoding [11]. This implies that a properly designed low rate code can further enhance the performance of a CDMA system.

#### Multi-ary OAM, ASK and PSK

For higher spectral efficiency, we can adopt  $\{x_k\}$  as a multi-ary modulation constellation. A joint design strategy of all component tree codes is established [4]. This leads to the so-called "asymmetrical and time-varying" structures with good distance distributions. Compared with the existing turbo-TCM schemes [12][13], the codes in [4] have significantly reduced decoding complexities and still demonstrate improved performances, as seen from Fig.2.

#### IV. CONCLUSIONS

We have demonstrated that two-state CT codes provide versatile and efficient solutions to a wide range of coding applications with near capacity performances.

#### REFERENCES

- [1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," *Proc. IEEE ICC'93*, vol. 2, pp. 1064-1070, 1993.
- [2] D. J. C. McKay and R. M. Neal, "Near Shannon limit performance of low-density parity check codes," *Electron. Lett.*, vol. 33, no. 6, pp. 457-458, Mar. 1997.
- [3] Li Ping and K. Y. Wu, "Concatenated tree codes: A low complexity, high performance approach," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 791-799, Feb. 2001.
- [4] Li Ping, B. M. Bai and X. M. Wang, "Low complexity concatenated two-state TCM schemes with near capacity performance", submitted to *IEEE Trans. Inform. Theory* and available at <http://www.ee.cityu.edu.hk/~liping/research>.
- [5] S. Benedetto and G. Montorsi, "Unveiling turbo codes: some results on parallel concatenated decoding schemes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 409-428, Mar. 1996.
- [6] Li Ping, W. K. Leung and K. Y. Wu, "Low-rate turbo-Hadamard codes", submitted to *IEEE Trans. Inform. Theory* and available at <http://www.ee.cityu.edu.hk/~liping/research>.
- [7] Li Ping, W. K. Leung and Nam Phamdo, "Low density parity check codes with semi-random parity check matrix," *IEE Electron. Lett.*, vol. 35, no. 1, pp. 38-39, Jan. 1999.
- [8] Li Ping, X. L. Huang, and N. Phamdo, "Zigzag codes and concatenated zigzag codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 800-807, Feb. 2001.
- [9] P. Komulainen and K. Pehkonen, "Performance evaluation of superorthogonal turbo codes in AWGN and flat Rayleigh fading channels," *IEEE J. Select. Areas Commun.*, vol. 16, no. 2, pp. 196-205, Feb. 1998.
- [10] D. Divsalar, H. Jin and R. McEliece, "Coding theorems for 'turbo-like' codes," *Proc. Allerton Conference*, pp. 201-210, 1998.
- [11] R. Herzog, J. Hagenauer and A. Schmidbauer, "Soft-in/soft-out Hadamard despreader for iterative decoding in the IS-95(A) system," *Proc. IEEE VTC'97*, pp. 1219-1222, 1997.
- [12] P. Robertson and T. Woz, "Bandwidth-efficient turbo trellis-coded modulation using punctured component codes," *IEEE J. Select. Areas Commun.*, vol. 16, no. 2, pp. 206-218, Feb. 1998.
- [13] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Parallel concatenated trellis coded modulation," *Proc. IEEE ICC'96*, pp. 974-978, 1996.

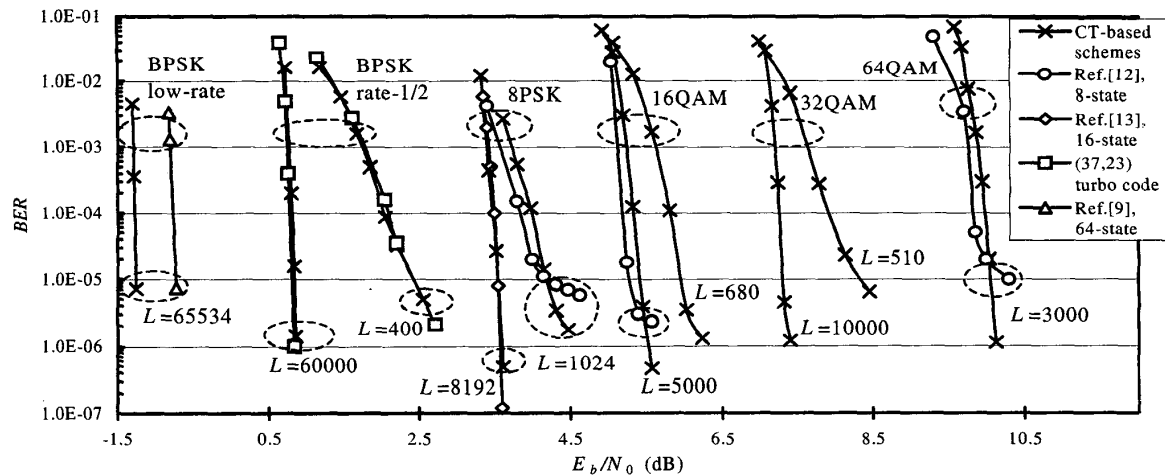


Fig. 2. Performance comparison of various CT-based schemes [3][4][6] and the existing schemes [9][12][13]. "L" represents the interleaver size (in bits for BPSK and symbols for other cases). The capacity limits are (modulation system, bits/symbol,  $E_b/N_0$ ): (BPSK, very low rate, -1.6dB), (BPSK, 1/2, 0dB), (8-PSK, 2, 2.9dB), (16-QAM, 3, 4.5dB), (32-QAM, 4, 6.8dB) and (64-QAM, 5, 9.2dB). The CT-based schemes have about 6-16 times lower decoding costs than their counterparts compared. For large interleaver sizes, the performance curves are about 0.5 dB from their respective capacities.