Performance Analysis of Turbo-SPC Codes

Keying Wu, *Student Member, IEEE*, Li Ping, *Member, IEEE*, Xiaoling Huang, and Nam Phamdo, *Senior Member, IEEE*

***Abstract*—This correspondence concerns the performance analysis of turbo-single-parity-check (SPC) codes based on the union bound of bit-error rate (BER). A treatment of the special interleavers used in turbo-SPC codes is discussed. It is shown that simple two- or four-state turbo-SPC codes with multiple component codes can perform comparably as (or even better than) the 16-state standard turbo codes. Using more complex trellis codes (with state number more than 4) appears unnecessary for such codes. Instead, performance improvement can be achieved by increasing the number of component codes, which maintains the low decoding complexity property of turbo-SPC codes.**

***Index Terms*—Free distance, parallel concatenated codes, single-parity-check (SPC) codes, turbo codes, union bound.**

I. Introduction

Turbo codes are powerful error-correcting codes [1]. Recently, it has been shown that turbo-single-parity-check (SPC) codes [2] provide a low-complexity alternative to the standard punctured turbo codes [1] for adjustable coding rates. The decoding cost of a turbo-SPC code is usually much lower (5–10 times) than that of a standard turbo code with comparable performance, which is important from the implementation point of view. Previous work [2] on turbo-SPC codes is mainly based on simulation for the bit-error-rate (BER) range above 10^{-6}. For high-quality systems, such as data communication and storage systems, a BER of 10^{-8} or even lower may be required but then simulation becomes excessively time consuming. In this case, the bound technique is a more efficient approach for performance assessment [3]–[9].

This correspondence concerns the performance analysis of turbo-SPC codes in the error floor range. The union bound [3] is adopted for this purpose that provides an upper bound on the optimal maximum-likelihood (ML) decoding. However, the decoding of turbo-SPC codes is accomplished by a suboptimal iterative process. Thus, the union bound is actually "an upper bound of the lower bound" on the BER performance of turbo-SPC codes. Nevertheless, this approach is still useful since it has been repeatedly observed that simulation results of turbo codes agree well with the union bound results in the error floor range.

Based on the union bound results, we will show that increasing the number of component codes is an efficient approach to improve the performance of turbo-SPC codes without seriously increasing the decoding cost [2]. It will be shown that some simple two- or four-state turbo-SPC codes can achieve performance comparable to (or even better than) much more complex 16-state standard turbo codes, and the use of more complex trellis codes (with more than four states) seems unnecessary. These findings serve as a helpful guideline for the design of turbo-SPC codes.

Manuscript received February 13, 2001; revised June 26, 2004. This work was supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project CityU 1182/02E).

K. Wu and L. Ping are with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong (e-mail: kewu@ee.cityu.edu.hk; eeliping@cityu.edu.hk).

X. Huang is with the Department of Electrical and Computer Engineering, State University of New York at Stony Brook, Stony Brook, NY 11794-2350 USA (e-mail: xhuang@ece.sunysb.edu).

N. Phamdo is with the Applied Physics Laboratory, The Johns Hopkins University, Laurel, MD 20723 USA (e-mail: nam.phamdo@jhuapl.edu).

Communicated by R. Urbanke, Associate Editor for Source Coding.

Digital Object Identifier 10.1109/TIT.2004.834850

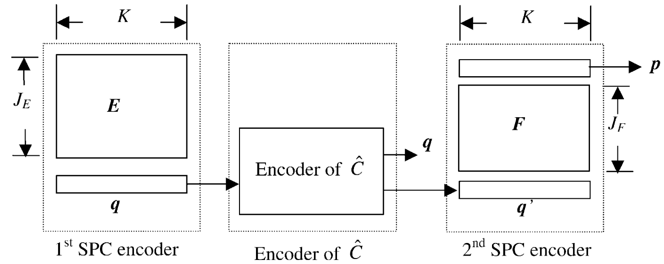

Fig. 1. The encoder of a convolutional-SPC code. The codeword is formed by $\{\boldsymbol{E}, \boldsymbol{F}, \boldsymbol{p}\}$.

It can be shown that some concatenated tree codes studied in [10] are special cases of turbo-SPC codes. Thus, the discussion in this correspondence also provides a means for the performance analysis of such codes in the error floor range, where simulation is difficult.

II. Turbo-SPC Codes

In this section, we briefly review the structure of turbo-SPC codes and propose a two-interleaver technique to solve the error floor problem of such codes.

A. Convolutional-SPC Code

Convolutional-SPC codes [2] are component codes of turbo-SPC codes, in which an SPC precoding technique is adopted for rate adjustment. Compared with other rate adjustment techniques based on puncturing, the SPC precoding technique leads to greatly reduced decoding complexity [2].

A convolutional-SPC encoder [2] is illustrated in Fig. 1. The information sequence is divided into two arrays $\boldsymbol{E}$ and $\boldsymbol{F}$ with J_E and J_F rows, respectively, and K columns. The encoding procedure can be summarized as follows.

- For each k $(k = 1, \ldots, K)$, calculate the parity check q_k of the kth column of $\boldsymbol{E}$.
- Use $\boldsymbol{q} = \{q_k\}$ to drive a systematic trellis code $\hat{C}$, producing a parity vector $\boldsymbol{q}' = \{q'_k\}$.
- For each k $(k = 1, \ldots, K)$, calculate the parity check p_k of the kth column of $\begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{q}' \end{bmatrix}$.
- The final output is $\{\boldsymbol{E}, \boldsymbol{F}, \boldsymbol{p}\}$, where $\boldsymbol{p} = \{p_k\}$.

(Note: The intermediate vectors $\boldsymbol{q}$ and $\boldsymbol{q}'$ do not appear in the final output.) The trellis representation of a convolutional-SPC code can be derived directly from that of $\hat{C}$ [2], [10]. Based on its trellis representation, the input–output weight distribution of a convolutional-SPC code can be obtained using the technique introduced in [11], which is useful in calculating the union bound [3] for the overall turbo-SPC code (as will be discussed in Section II-C).

B. Recursive and Nonrecursive Parts

The recursive linear code has the property that a weight-1 information sequence will always produce a codeword with infinite weight (assuming an infinite code length). For convolutional-SPC codes introduced in Section II-A, it is easy to verify that if $\hat{C}$ is recursive, a weight-1 information sequence will always produce an infinite-weight codeword if the information one is in $\boldsymbol{E}$. Otherwise, if the information one is in $\boldsymbol{F}$, it will produce only one parity one. For this reason, we call $\boldsymbol{E}$ the recursive part and $\boldsymbol{F}$ the nonrecursive part, respectively.

C. Turbo-SPC Codes

A turbo-SPC code C is constructed by concatenating M component convolutional-SPC codes in parallel via interleavers [2]. The overall codeword of a turbo-SPC code is formed by $\{\boldsymbol{D}, \boldsymbol{p}^{(0)}, \ldots, \boldsymbol{p}^{(M-1)}\}$, where $\boldsymbol{D} = \{\boldsymbol{E}, \boldsymbol{F}\}$ and $\boldsymbol{p}^{(m)}$ is the parity sequence generated by the mth component encoder.

With uniform interleavers [3], the input–output weight distribution of a turbo-SPC code can be obtained from those of its component codes [3], based on which the union bound on BER can be generated as [3]

$$P_b \leq \frac{1}{2} \sum_d \sum_{\substack{w,j \\ w+j=d}} \frac{w}{L} A_{w,j} \operatorname{erfc}\left(\sqrt{d \frac{E_b R}{N_0}}\right) \tag{1}$$

where L is the interleaver length, R the coding rate, and $A_{w,j}$ the number of codewords with information weight w and parity weight j.

D. Error Floor Problem of Turbo-SPC Codes With Uniform Interleavers

The concept of uniform interleaver [3] greatly simplifies the analysis of turbo-type codes. However, we will show that the use of uniform interleavers leads to a serious error floor problem for turbo-SPC codes. A cyclic interleaver will be proposed to solve this problem.

The conditional redundancy weight enumerating function (CRWEF) of a code C is [3]

$$A_w^C(Z) = \sum\nolimits_j A_{w,j} Z^j \tag{2}$$

where $A_{w,j}$ has the same definition as in (1). With uniform interleavers, the CRWEF of a turbo-SPC code C can be derived from those of its component codes $C^{(m)} (m = 0, \ldots, M-1)$ as

$$A_w^C(Z) = \left(\prod_{m=0}^{M-1} A_w^{C^{(m)}}(Z)\right) \bigg/ \binom{L}{w}^{M-1} \tag{3}$$

where $A_w^{C^{(m)}}(Z)$ is the CRWEF of $C^{(m)}$. (Note: The weight distribution obtained in this way is an average over the ensemble of all possible interleavers.) Assume $J_F > 0$ and let $J = J_E + J_F$. For $w = 1$ we have

$$\begin{aligned} A_1^C(Z) &= \left(\prod_{m=0}^{M-1} A_1^{C^{(m)}}(Z)\right) \bigg/ \binom{KJ}{1}^{M-1} \\ &= \left(\prod_{m=0}^{M-1} A_1^{C^{(m)}}(Z)\right) \bigg/ (KJ)^{M-1}. \end{aligned} \tag{4}$$

Let $A_1^{C^{(m)}}(Z) = N_1^{C^{(m)}}(Z) +$ other terms, where $N_1^{C^{(m)}}(Z)$ consists of the terms generated by those weight-1 information sequences whose only information one appears in $\boldsymbol{F}$ for the mth component code. It is easy to see that $N_1^{C^{(m)}}(Z) = KJ_F Z$ with KJ_F the size of the nonrecursive part. Then (4) becomes

$$A_1^C(Z) = KJ_F \left(J_F/J\right)^{M-1} Z^M + \text{other terms}. \tag{5}$$

The first term in the right-hand side of (5) represents codewords with a relatively small weight of $M+1$ (parity weight $= M$ and information weight $= 1$), whose multiplicity grows linearly with the interleaver length. Such codewords lead to a high error floor, which has been confirmed by simulation results.

E. Two-Interleaver Technique to Overcome the Error Floor Problem

In the following, we propose a two-interleaver technique to treat the error floor problem mentioned above. Partition the information array $\boldsymbol{D}$ into M nonoverlapping subsets $\boldsymbol{D}^{(0)}, \boldsymbol{D}(1), \ldots, \boldsymbol{D}^{(M-1)}$ with equal size $J \times K/M$. Let $a' \equiv J_E M/J$ and $a'' \equiv J_F M/J$. We assume that a' and a'' are both integers.[1] Define

$$\langle x \rangle \equiv x \text{ modulo } M.$$

For the mth component code, $\boldsymbol{E}$ is obtained by applying a random interleaver to the elements in $\boldsymbol{D}^{(\langle m \rangle)}, \boldsymbol{D}^{(\langle m+1 \rangle)}, \ldots, \boldsymbol{D}^{(\langle m+a'-1 \rangle)}$ and $\boldsymbol{F}$ is obtained by applying an independent random interleaver to the elements in the complementary of $\boldsymbol{E}$ in $\boldsymbol{D}$. In this way, every information bit will be in $\boldsymbol{E}$ and $\boldsymbol{F}$ for exactly a' and a'' times, respectively, over M component codes. With $J_E > 0$, the above two-interleaver technique eliminates the error events related to weight-1 information sequences (assuming an infinite interleaver length), since the information one will appear in $\boldsymbol{E}$ at least once and generate an infinite parity weight if $\hat{C}$ is recursive.

Recall that (3) is based on the assumption that one uniform interleaver is used for each component code [3]. This assumption is no longer valid if the two-interleaver technique is used. A solution to this problem is given in the Appendix.

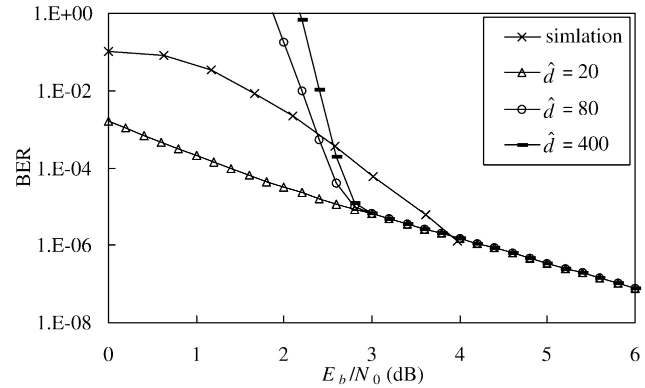

Fig. 2. Union bounds on BER for a two-state turbo-SPC code with $L = 200$, $M = 4$, $(J_E, J_F) = (4, 0)$, generator $= 1/(1+x)$, and different $\hat{d}$ values. Simulated result is included for reference. Notice that the simulated result is above the bound results for 2.3 dB $< E_b/N_0 <$4 dB. Similar phenomenon has been observed in other literature [8], which may be due to the suboptimal iterative decoding.

III. Performance Comparisons

In this section, we present a comparative study between turbo-SPC codes and turbo codes. We will focus on the performance in the error floor range where the union bound is relatively accurate. Simulated results will be used as supplements in the waterfall range where the union bound is quite loose. Let L be the interleaver length, M the number of component codes, S the state number of the reference convolutional code $\hat{C}$, and R the code rate. Throughout the following discussions, we will use a standard 16-state $1/2$-rate punctured $(37, 23)$ turbo code (with generator $(1+x+x^2+x^3+x^4)/(1+x^3+x^4)$) as reference, which is known for its low error floor [12].

A. A Numerical Consideration

Denote by $d_{\max}$ the maximum value of d in (1). With large $d_{\max}$, the generation of the whole $\{A_{w,j}\}$ is very time consuming. One solution is to choose a number $\hat{d} < d_{\max}$ and approximate the summation in (1) by a smaller one containing only those terms with $d \leq \hat{d}$. Fig. 2 reports the impact of $\hat{d}$ on the union bound of a two-state turbo-SPC code. It is seen that all curves merge around $E_b/N_0 = 3$ dB and stay together

[1] This may not be satisfied for every $(\boldsymbol{R}, \boldsymbol{M})$ pair, but usually we can find suitable values to meet the requirement.

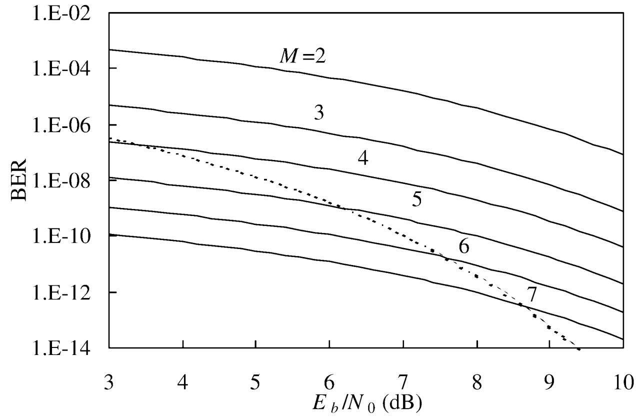

Fig. 3. The comparison of the free-distance asymptotes for turbo-SPC codes and the reference turbo code with $L = 200$ and $R = 1/2$. The turbo-SPC codes have $S = 2$, $(J_E, J_F) = (M, 0)$, and $M = 2 \sim 7$. All the codes are terminated using approximately eight extra bits. The solid lines are for turbo-SPC codes and the dashed line for the turbo code.

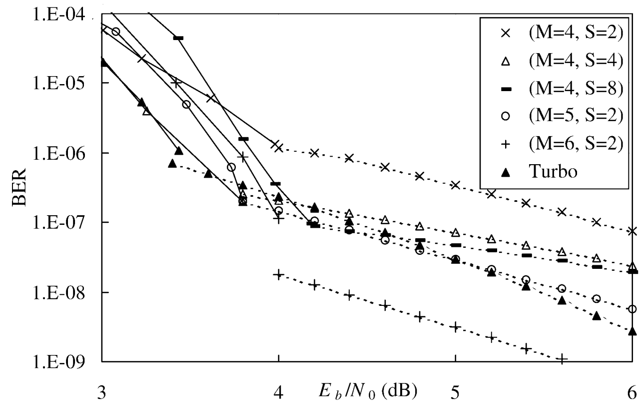

Fig. 4. Performance comparison among turbo-SPC codes with $L = 200$, $(J_E, J_F) = (M, 0)$, $M = 4, 5, 6$, and $S = 2, 4, 8$. The solids lines are for simulated results and the dashed lines for union bounds.

from there on. Therefore, $\hat{d} = 20$ appears sufficient for analyzing the error floor behavior, and will be used throughout the later discussion.

B. Asymptotic Behavior of Turbo-SPC Codes With Only the Recursive Part

For high signal-to-noise ratio (SNR) values, (1) is dominated by the free-distance term [13]

$$P_b \approx \frac{\tilde{w}_{\text{free}} \cdot N_{\text{free}}}{L} \times \frac{1}{2}\text{erfc}\left(\sqrt{d_{\text{free}} \cdot \frac{E_b R}{N_0}}\right) \tag{6}$$

where $\tilde{w}_{\text{free}}$ is the average weight of the information sequences generating free-distance codewords, and N_{free} is the multiplicity, or thetotal number, of free-distance codewords. Equation (6) is sometimes referred to as the free-distance asymptote of a code [13].

We first consider the turbo-SPC code with only the recursive part, i.e., $J_F = 0$. In this case, the two-interleaver technique simplifies to generating one random interleaver, so (3) is applicable assuming one uniform interleaver. This helps to gain some insights into the code structure.

With $J_F = 0$ and $J_E \geq 2$, the free-distance codewords correspond to the situation of weight-2 information sequences with both information ones appearing in the same column in every component code. Thus, $d_{\text{free}} = 2$ and [14]

$$\begin{aligned} N_{\text{free}} = A_{2,0}^{C} &= \binom{L}{2}\left(K\binom{J}{2}\Big/\binom{L}{2}\right)^{M} \\ &\approx \frac{1}{2}(J-1)^{M}\left(\frac{1}{L}\right)^{M-2}. \end{aligned} \tag{7}$$

Clearly, increasing either L or M leads to lower N_{free}, i.e., lower error floor. Fig. 3 illustrates the free-distance asymptotes for turbo-SPC codes with different M. The free-distance asymptote of the reference turbo code is also included, which has $d_{\text{free}} = 4$ and $N_{\text{free}} \approx 2.22 \times 10^{-6}$. The turbo code has better asymptotic performance at very high SNRs due to its larger d_{free}, but in the range of practical interest (e.g., BER $> 10^{-8}$), turbo-SPC codes demonstrate better potential.

C. Impact of the Numbers of States and Component Codes

Based on the above discussion, for turbo-SPC codes with $J_F = 0$ and $J_E \geq 2$, their d_{free} and N_{free} are independent of the convolutional code $\hat{C}$ used. However, the use of more complex $\hat{C}$ can reduce the multiplicities of other small-weight codewords and still lead to better performance.

The effects of increasing M and/or S are compared in Fig. 4. Turbo-SPC codes with two, four, and eight states are based on generators $1/(1+x)$, $(1+x)/(1+x+x^2)$, and $(1+x^2+x^3)/(1+x+x^3)$, respectively. It can be observed that the advantage of increasing S to more than four is marginal, and increasing M can suppress the error floor more efficiently.

We make a brief comparison for the decoding costs of the codes in Fig. 4. Use the ($M = 4$, $S = 2$) code as a reference. Increasing M from 4 to 5, the decoding complexity increases only by a factor of $5/4$ [2], while increasing S from 2 to 4, the decoding complexity at least doubles. (It is shown in [2] that based on a normalizing technique, the decoding cost of a two-state trellis code is only about $1/3$ of that of a four-state one.) Therefore, increasing M appears a better choice if only the error floor performance is concerned. Overall, we have the following observations.

- For BER above 2×10^{-7}, the ($M = 4$, $S = 4$) turbo-SPC code seems the best choice. Although the turbo code has comparable performance in this range, its decoding cost is about five times higher.
- For BER below 2×10^{-7}, the ($M = 6$, $S = 2$) turbo-SPC code is a favorable choice.

Notice that increasing M to more than 5 leads to performance degradation in the waterfall range. We are still investigating this issue.

D. Turbo-SPC Codes With Hybrid Recursive and Nonrecursive Parts

We now consider the impact of the relative sizes of the recursive and nonrecursive parts on the performance of turbo-SPC codes. We use turbo-SPC codes with $R = 1/2$, $M = 4$, $S = 2$, generator $= 1/(1+x)$, $(J_E, J_F) = (4, 0)$ and $(3, 1)$ as examples. The two-interleaver technique discussed in Section II-E is employed for the $(3, 1)$ code.

Compare d_{free} and N_{free} for these two codes. As explained in Section III-B, for the $(4, 0)$ code we have

$$d_{\text{free}} = 2 \text{ and } N_{\text{free}} = A_{2,0}^{C} \approx \frac{1}{2}(J-1)^{M}\left(\frac{1}{L}\right)^{M-2} = \frac{81}{2} \cdot \frac{1}{L^2}. \tag{8a}$$

The free distance codeword of the $(3, 1)$ code corresponds to the event of weight-2 information sequences with both information ones appearing in the same column in $\boldsymbol{E}$ in three component codes (and

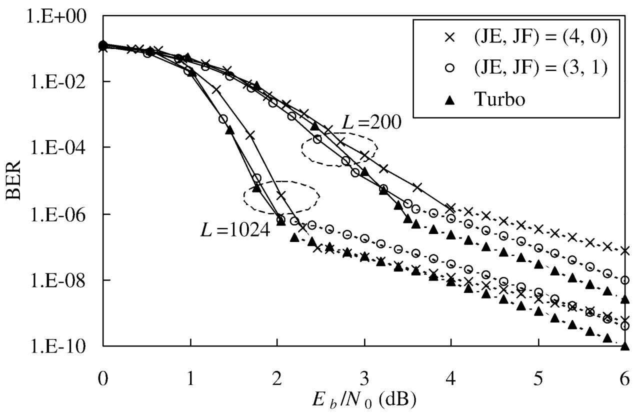

Fig. 5. Performance comparison for turbo-SPC codes with $(J_E, J_F) = (4,0)$ and $(3,1)$. $M = 4$, $S = 2$, generator $= 1/(1+x)$, $L = 200$ and 1024. The solids lines are for simulated results and the dashed lines for union bounds.

they generate two nonzero parity bits when they are in $\boldsymbol{F}$ in the rest component code). Thus, for the $(3,1)$ code, we have

$$\begin{aligned} d_{\text{free}} &= 4 \text{ and } N_{\text{free}} = A^{C}_{2,2} \\ &= J \binom{K}{2} \left(K \binom{J_E}{2} \Big/ \binom{J_E K}{2} \right)^{M-1} \\ &\approx \frac{1}{2} L K \left(\frac{J_E - 1}{J_E K} \right)^{M-1} = \frac{64}{27} \cdot \frac{1}{L}. \end{aligned} \tag{8b}$$

Comparing (8a) and (8b), N_{free} decreases faster as L increases for the $(4,0)$ code. This effect can be seen from Fig. 5, where comparison is made for $L = 200$ and 1024. It is seen that the $(3,1)$ code is always superior to the $(4,0)$ code for $L = 200$. However, for $L = 1024$, the later outperforms the former in 2 dB $< E_b/N_0 <$ 5 dB. This is because, as L increases, N_{free} decreases faster for the $(4,0)$ code than for the $(3,1)$ code. For very high $E_b/N_0 (>$ 5.5 dB), the $(3,1)$ code (with $L = 1024$) still has lower error floor due to its larger free distance. Another advantage of the $(3,1)$ code resides in its waterfall behavior, which is always better than that of the $(4,0)$ code. The performance improvement due to the increase of L agrees well with the results for standard turbo codes [3], [5].

IV. Discussion and Conclusion

We have applied the union bound technique to assess the error floor behavior of turbo-SPC codes. Based on the bound results, we have shown that simple turbo-SPC codes with two or four states can achieve similar (or better) performance compared with the standard 16-state turbo code.

From Figs. 3–5, it is seen that for a specific required BER value, we can always find some two-state turbo-SPC codes to achieve comparable performance as the standard turbo code at much lower decoding costs. If the BER requirement is specified over a wide range, then four-state turbo-SPC codes seem to be good choices. Using more complex component codes has only marginal benefit. Instead, performance improvement in the error floor range can be achieved more efficiently by increasing the number of component codes, which is an attractive approach since it maintains the low-cost property of turbo-SPC codes.

Appendix
Weight Distribution Function of Turbo-SPC Codes With Two-Interleaver Technique

In this appendix, we outline a technique to obtain the CRWEF of turbo-SPC codes using the two-interleaver technique introduced in Section II-E. The key problem is to separate the input weights involved in two interleavers for every component code. We use two dummy variables U and V to indicate the weights of $\boldsymbol{E}$ and $\boldsymbol{F}$, respectively. Thus, the definitions of CRWEF in (2) should be modified to

$$\tilde{A}^{C}_{u,v}(Z) = \sum_j \tilde{A}_{u,v,j} Z^j \tag{9}$$

where $\tilde{A}_{u,v,j}$ denotes the number of codewords with u ones in $\boldsymbol{E}$, v ones in $\boldsymbol{F}$ and j ones in $\boldsymbol{p}$.

Consider a turbo-SPC code C consisting of M component codes $C^{(0)}, C^{(1)}, \ldots, C^{(M-1)}$. Let w be the information weight and

$$\boldsymbol{w} \equiv (w^{(0)}, w^{(1)}, \ldots, w^{(M-1)})$$

the numbers of ones in $\boldsymbol{D}^{(0)}, \boldsymbol{D}^{(1)}, \ldots, \boldsymbol{D}^{(M-1)}$, with

$$\sum_{x=0}^{M-1} w^{(x)} = w.$$

Considering all the possible distributions of w ones in $\boldsymbol{D}^{(0)}$, $\boldsymbol{D}^{(1)}, \ldots, \boldsymbol{D}^{(M-1)}$, the probability of a particular event $\boldsymbol{w}$ is

$$\alpha = \prod_{x=0}^{M-1} \binom{aK}{w^{(x)}} \Big/ \binom{J \cdot K}{w} \tag{10}$$

where a is the number of rows in each subset $\boldsymbol{D}^{(x)}$ and aK the size of each subset $\boldsymbol{D}^{(x)} (0 \le x \le M-1)$.

Using the two-interleaver technique in Section II-E, the $\boldsymbol{E}$ part of the mth component code is composed of the elements in $\boldsymbol{D}^{(\langle m \rangle)}, \ldots, \boldsymbol{D}^{(\langle m+a'-1 \rangle)}$. Thus, for $C^{(m)}$, the number of ones in $\boldsymbol{E}$ is

$$u(m) = \sum_{x=m}^{m+a'-1} w^{(\langle x \rangle)}, \qquad 0 \le m \le M-1 \tag{11a}$$

and the number of ones in $\boldsymbol{F}$ is

$$v(m) = w - u(m), \qquad 0 \le m \le M-1 \tag{11b}$$

where a' and $\langle \cdot \rangle$ are defined in Section II-E. Therefore, $\tilde{A}^{C^{(m)}}_{u(m),v(m)}(Z)$ represents the conditional weight distribution of $C^{(m)}$ for a particular $\boldsymbol{w}$. Then, two independent uniform interleavers are applied to $\boldsymbol{E}$ and $\boldsymbol{F}$, respectively. Considering all the possibilities, the event of a specific interleaver pair has the probability of

$$\beta_m = \frac{1}{\binom{J_E K}{u(m)} \binom{J_F K}{v(m)}}. \tag{12}$$

Based on the work of [3], the parallel concatenation of M such component codes has the following conditional weight distribution for a particular $\boldsymbol{w}$:

$$\beta \prod_{m=0}^{M-1} \tilde{A}^{C^{(m)}}_{u(m),v(m)}(Z) \tag{13}$$

where

$$\beta \equiv \prod_{m=1}^{M-1} \beta_m. \tag{14}$$

Considering all possible $\boldsymbol{w}$, we have the CRWEF of C

$$A_w^C(Z) = \sum_{\boldsymbol{w}} \left(\alpha\beta \prod_{m=0}^{M-1} \tilde{A}^{C^{(m)}}_{u(m),v(m)}(Z) \right). \tag{15}$$

Keep in mind that $u(m)$, $v(m)$, α, and β are all functions of $\boldsymbol{w}$ as defined in (11), (10), and (14).

The complexity of the above technique is considerably higher than that of the standard method for generating CRWEF in (3) [3]. Therefore, it is most useful when $\hat{d}$ is relatively small, say, $\hat{d} < 30$. Such $\hat{d}$ is sufficient to provide reasonably accurate assessment for the error floor behavior, see Fig. 2, but it is not sufficient for the waterfall behavior, for which we still rely on simulation. For the waterfall behavior, the tangent sphere bound [7]–[9] is much tighter than the union bound but it requires the whole weight distribution $\{A_{w,j}\}$ for a turbo-type code [7]–[9]. This constitutes a difficulty for the two-dummy-variable technique introduced above due to complexity.

REFERENCES

[1] C. Berrou and A. Glavieux, "Near Shannon limit error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.

[2] L. Ping, "Turbo-SPC codes," *IEEE Trans. Commun.*, vol. 49, pp. 754–759, May 2001.

[3] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 409–428, Mar. 1996.

[4] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Analysis, design, and iterative decoding of double serially concatenated codes with interleavers," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 231–244, Feb. 1998.

[5] A. G. Burr and G. P. White, "Comparison of iterative decoder performance with union bounds for short frame turbo codes," *Ann. Telecommun.*, vol. 54, no. 3–4, pp. 201–207, Mar.–Apr. 1999.

[6] E. Telatar and R. Urbanke, "On the ensemble performance of turbo codes," in *Proc. IEEE int. Symp. Information Theory (ISIT'97)*, Ulm, Germany, June 1997, p. 105.

[7] G. Poltyrev, "Bounds on the decoding error probability of binary linear codes via their spectra," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1284–1292, July 1994.

[8] I. Sason and S. Shamai (Shitz), "Improved upper bounds on the ML decoding error probability of parallel and serial concatenated turbo codes via their ensemble distance spectrum," *IEEE Trans. Inform. Theory*, vol. 46, pp. 24–47, Jan. 2000.

[9] ——, "Bounds on the error probability of ML decoding for block and turbo-block codes," *Ann. Telecommun.*, vol. 54, no. 3–4, pp. 183–201, Mar.–Apr. 1999.

[10] L. Ping and K. Y. Wu, "Concatenated tree codes: A low complexity, high performance approach," *IEEE Trans. Inform. Theory*, vol. 47, pp. 791–799, Feb. 2001.

[11] A. J. Viterbi, A. M. Viterbi, J. Nicolas, and N. T. Sindhushayana, "Perspectives on interleaved concatenated codes with iterative soft output decoding," in *Proc. Int. Symp. Turbo Codes and Related Topics*, Brest, France, Sept. 3–5, 1997, pp. 47–54.

[12] S. Benedetto, R. Garello, and G. Montorsi, "A search for good convolutional codes to be used in the construction of turbo codes," *IEEE Trans. Commun.*, vol. 46, pp. 1101–1105, Sept. 1998.

[13] L. C. Perez, J. Seghers, and D. J. Costello Jr, "A distance spectrum interpretation of turbo codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1698–1709, Nov. 1996.

[14] X. L. Huang, N. Phamdo, and L. Ping, "BER bounds on parallel concatenated single parity check arrays and zigzag codes," in *Proc. GLOBECOM'99*, Rio de Janeiro, Brazil, Dec. 1999, pp. 2436–2440.

Burst Erasure Correction Codes With Low Decoding Delay

Emin Martinian, *Member, IEEE,* and
Carl-Erik W. Sundberg, *Fellow, IEEE*

***Abstract*—We present a new class of systematic, time-invariant, convolutional encoders suitable for low delay burst erasure correction. Specifically, we show that the new encoders have the shortest possible decoding delay required to correct all bursts of a given length with a fixed redundancy. By comparing the new encoders to maximum distance separable (MDS) codes, we show that the latter generally require either more redundancy or more delay to correct bursts of a given length. In addition, we show that the new encoders can achieve better performance than MDS codes on a simple two-state Markov erasure channel. Thus, we demonstrate the advantages of using cross packet coding for delay-sensitive applications such as Voice over Internet Protocol, video-conferencing, etc., on bursty packet networks. Finally, we discuss suitable performance measures for encoders designed to correct both burst and random erasures and report the results of a computer search for such hybrid encoders.**

***Index Terms*—Convolutional codes, erasure channel, low-delay coding, maximally short codes.**

I. INTRODUCTION

Recent investigations suggest that, in a variety of networks, packet losses occur in bursts [3]–[7]. Conventional application of error-correcting codes or retransmission (e.g., automatic repeat request (ARQ)) for packet recovery often requires interleaving and long decoder delays. Since long delays are usually unacceptable in real-time multimedia communication applications such as Voice over Internet Protocol, video-conferencing, tele-medicine, etc., erasure correction codes with low decoding delays are desirable.

Specifically, consider a scenario where a source (e.g., audio or video) must be transmitted over a packet channel (e.g., the Internet). Specifically, imagine that every second, one frame of the source consisting of k bits is provided to the encoder. Similarly, each second the transmitter can send a packet of n bits which is either received correctly at the receiver or erased (e.g., due to congestion at an intervening Internet router). Due to delay constraints, the receiver must attempt to reproduce source packet i from the received packet stream with a delay of at most T packets. How should the transmitter encode each source packet into a channel packet such that the system is robust to bursts of packet losses?

Previous researchers have mostly focused on codes designed to maximize burst error-correcting capability without regard to decoding delay (see [8]–[16] and references therein). An important metric in such work is the relationship between rate, correctable burst length, and guard space (the number of correctly received symbols required between bursts). In contrast, we consider codes for burst erasure correction and are primarily concerned with the relationship between rate, correctable burst length, and decoding delay.

Manuscript received October 1, 2000; revised June 14, 2002. The material in this correspondence was presented in part at the International Conference on Communications, May 2002 [1] and in part at the IEEE International Symposium on Information Theory, Laussane, Switzerland, June/July 2002 [2]. This work was initiated while both authors were with Lucent Technologies' Bell Labs, Murray Hill, NJ.

E. Martinian is with Mitsubishi Electric Research Laboratoriess, Cambridge, MA 02139 USA (e-mail: emin@alum.mit.edu).

C.-E. W. Sundberg is with SundComm, Chatham, NJ 07928 USA (e-mail: cews@ieee.org).

Communicated by R. Urbanke, Associate Editor for Coding Techniques.
Digital Object Identifier 10.1109/TIT.2004.834844