A STUDY ON THE PERFORMANCE, POWER CONSUMPTION TRADEOFFS OF SHORT FRAME TURBO DECODER DESIGN

Zhipei Chi, Zhongfeng Wang and Keshab K. Parhi

Department of Electrical and Computer Engineering
University of Minnesota, Minneapolis, MN 55455, USA
E-mail:{*zchi, zwang, parhi*}*@ece.umn.edu*

ABSTRACT

Protecting short frames using turbo coding is a challenging task because of the small interleave size and the need for transmission efficiency. In this paper, we explore possible trade-off between power consumption (estimated by the average number of iterations) and performance of turbo decoders when short frame turbo codes are used. Three encoding/decoding schemes are proposed to improve performance of turbo decoder in terms of frame/bit error rate, and to increase the data transmission efficiency whether ARQ protocols are performed or not. Specifically, turbo decoding metrics aided short CRC codes are applied to terminated trellis codes, tail-biting encoded trellis codes and CRC embedded trellis codes with a two-fold purpose: to stop the iterative decoding processes and to detect decoding errors at the last iteration. We show that significant coding gains can be achieved by actually increasing the coding rate with negligible increase in power consumption. Performance improvement is demonstrated over both AWGN and Rayleigh flat fading channels.

1. INTRODUCTION

Since their introduction by Berrou *et al.* [1], turbo codes have received tremendous attention in the recent years. It has been shown that turbo codes can provide significant coding gains for additive white Gaussian noise (AWGN) channels since a relatively long data frame, a random interleaver and an iterative decoding process are utilized.

Outstanding performance can be achieved by turbo codes when the frame size is relatively large, e.g., 1024 or larger. However, in the 3rd generation wireless communication systems, there are also short frames of less than 50 information bits which need to be transmitted [2]. Protecting these short frames turns out to be a more difficult task compared with the case in which longer frame size is used. Many problems arise in this situation, such as “how to find the best possible coding scheme in terms of both performance and efficiency”, “how the coding should change with the channel SNR and type”, etc. In particular, the insertion of trellis terminating bits and cyclic redundancy check (CRC) bits causes significant rate loss when the number of information bits k is small, where the CRC codes are introduced with a two-fold purpose: to stop the iterative coding processes and to detect decoding errors at the last iteration. In other words, the effective signal to noise ratio per channel bit is decreased. Let L, K and L_{CRC} denote the length of uncoded data frame, constraint length of the recursive systematic convolutional (RSC) encoder and the length of the CRC codes, respectively. The coding rate r and the energy per channel bit E_c are defined as:

$$r = (L - (K-1) - L_{CRC})\,/\,n \quad (1)$$
$$E_c = r\,E_b \quad (2)$$

If we use frame length 120, constraint length 5 turbo codes and 8-bit CRC codes as recommended by [2], then according to (1) and (2), a $0.45\,dB$ SNR loss occurs by turbo encoding. To combat the rate loss (therefore, SNR loss), we propose using turbo decoding metrics (TDMs) aided short CRC codes to carry out the error detection operation during the turbo decoding process. The performance of proposed algorithm has been studied under three different encoding/decoding schemes, namely,

1. The original terminated trellis turbo codes;
2. A novel tail-biting coding/decoding scheme;
3. A novel CRC code embedded trellis coding/decoding scheme.

The rest of the paper is organized as follows. In section 2, the error detection algorithm which uses turbo decoding metrics aided CRC codes is described. Then in section 3, the performance of proposed algorithm is studied under both AWGN and Rayleigh flat fading channels. In section 4 and section 5, we demonstrate the functionality of proposed detection method under an efficient low decoding delay tail-biting encoding/decoding scenario and a CRC code embedded encoding/decoding scheme, respectively. Section 6 is devoted to the issue on how the algorithm should be adapted to accommodate ARQ protocols.

2. ERROR DETECTION USING TURBO DECODING METRICS AIDED CRC CODES

2.1. Turbo Decoding Metrics

Turbo decoder consists of two soft-output decoders, capable of accepting and generating soft outputs. Either maximum *a posteriori* (MAP) or soft output Viterbi algorithm (SOVA) can be adopted by the decoders. Each decoder computes the log likelihood ratio (LLR) of information bits and passes part of the LLR to its counterpart as extrinsic information. Specifically, we have

$$L_i(u_k) = L_c y_k^s + L_{ji}^e(u_k) + L_{ij}^e(u_k) \quad (3)$$

where $i,\, j \in \{1,\, 2\}$. In (3) u_k is the information bit, $L_i(u_k)$ is its log likelihood ratio and $L_{ij}^e(u_k)$ is the log extrinsic information. In [3], a set of turbo decoding metrics (TDMs) obtained during

This research was supported by DARPA under contract number DA/DABT 63 -96-C-0050.

the turbo decoding process have been defined. TDMs provide us extra information which can be utilized to design low energy, high throughput turbo decoders. In particular, we use three of them in the proposed error detection method, namely,

1. The number of non-matching bits. We call bit u_k a "non-matching bit (NMB)" if the signs of $L_{ij}^e(u_k)$ and $L_i(u_k)$ are different. Then by accumulating the number of NMBs in one data frame, we get the first decoding metric.
2. The minimum absolute value of the extrinsic information $L_{ij}^e(u_k)$ in one date frame, denoted as LE_{min}.
3. The minimum absolute value of the log likelihood ratio $L_i(u_k)$ in one date frame, denoted as LLR_{min}.

The statistical properties of the above three TDMs have been shown in [4] where justification on why they can serve to carry out error detection is given.

2.2. TDMs-CRC Error Detection

Similar to the algorithm proposed in [4], the following procedure can be implemented to distinguish if a decoded data frame is free of errors or not:

• • •

```
Carry out error detection using 1 even/odd parity check bit
or multiple CRC bits;
if (CRC error free & number of NMBs < threshold_1 &
    LLRmin > threshold_2 & LEmin > threshold_3) {
    stop iteration;
    send ACK signal to the sender;
}
else {
    if (# of iterations < pre-set value)
        go on to the next iteration;
    else
        send NACK signal to the sender for re-transmission;
    end;
end;
```

We'll refer the above algorithm as TDMs-CRC error detection. Note that in [4], the performance of the above algorithm was demonstrated only for the last iteration of turbo decoding. In this paper, we show that it can also be used to stop the iterative decoding process to avoid unnecessary iterations. However, the pre-set thresholds are tighter than those set in the last iteration to ensure extremely small performance degradation.

3. TDMS-CRC ERROR DETECTION FOR TERMINATED TRELLIS CODES

In this section, the performance of TDMs-CRC error detection method is demonstrated through computer simulations over AWGN and Rayleigh fading channels. The length of uncoded data frame L is set to 49. Rather than using 8-bit CRC code, a 4-bit CRC is appended after the 43 information bits. 2 terminating bits are added as $K = 3$. We show the frame error rate (FER) and bit error rate (BER) improvement over AWGN channels in Fig. 1. The following conclusions can be drawn from Fig. 1 and Fig. 2. Using 4-bit CRC code only will introduce significant performance degradation. Using 8-bit CRC to do early stopping also results in extra decoding errors; however, it is negligible. Using 4-bit TDMs-CRC code results in negligible performance degradation as well, but a $0.5\,dB$ coding gain compared with using 8-bit CRC code can be achieved in terms of either FER or BER. Small number of extra iterations are needed to ensure negligible performance loss. Using 0-bit CRC, i.e., TDMs only, achieves maximum coding gain, but requires more power consumption. In addition, TDMs only can not be used to do error detection in the last iteration since it will claim many free-of-error frames as error-corrupted frames. Note that this property of TDMs only lowers the efficiency of early decoding stopping. Same conclusions can be obtained from Fig. 3 and Fig. 4 where Rayleigh flat fading channel is assumed ($f_d T_S = 0.01$).

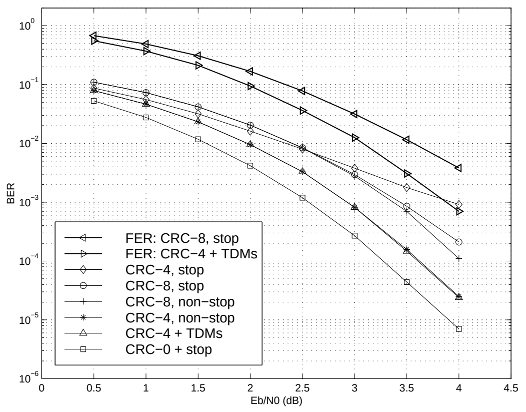

Figure 1. Performance of various early stopping algorithms (terminated trellis, K=3, AWGN channel)

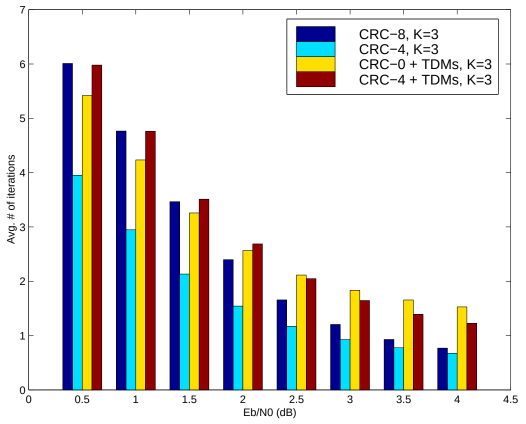

Figure 2. Average number of iterations of various early stopping algorithms (terminated trellis, K=3, AWGN channel)

4. TDMS-CRC ERROR DETECTION FOR TAIL-BITING ENCODED TRELLIS CODES

The algorithm given in [1] requires that the distribution of the starting and ending trellis states be known. This is achieved by terminating the trellis using extra termination bits. An alternative way to do path termination that does not reduce coding rate or degrade the error probability is tail-biting [5]. In [6], to solve the problem of how to initialize the α_0 and β_L vectors used in the BCJR

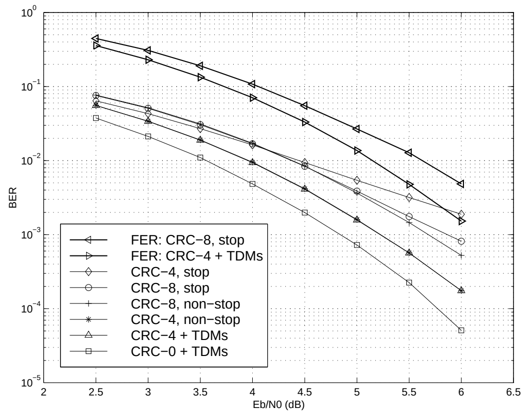

Figure 3. Performance of various early stopping algorithms (terminated trellis, K=3, Rayleigh fading channel)

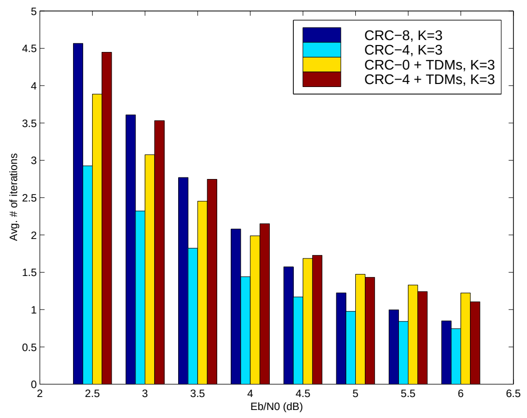

Figure 4. Average number of iterations of various early stopping algorithms (terminated trellis, K=3, Rayleigh fading channel)

algorithm [7], eigenvector of the trellis transition matrix was computed. We had briefly described a novel tail-biting scenario which avoids complicated computation of the eigenvector in [4]. In this paper, we show the performance of TDMs-CRC early stopping criterion on tail-biting encoded trellis. In order to maintain the same BER as the terminated trellis codes, tail-biting must only be applied when a large K is assumed. In particular, we assume the same set of parameters as in section 3 except that $K = 5$. Define the information vector $\mathbf{u} = (\, u_0, \, \ldots \, , u_{L-1} \,)$ and the interleaved information vector as $\mathbf{u}_{int}$. Then the ending state S_L depends on the joint vector $\mathbf{u}_{joint} = (\mathbf{u}, \, \mathbf{u}_{int})$. Following the procedure described in [8], the starting state S_0 can be obtained by solving the following equation:

$$(\mathbf{A}^L + \mathbf{I}_{K-1})S_0 = S_{2L}^{[zs]} \tag{4}$$

where the $S_{2L}^{[zs]}$ is the zero-state response for the joint information vector $\mathbf{u}_{joint}$. Therefore, the initial value of α_0 and β_L of decoder 2 can be found as:

$$\alpha_0^{(2)} = \alpha_L^{(1)}, \qquad \beta_L^{(2)} = \beta_0^{(1)}. \tag{5}$$

The same operation is carried out when decoder 2 passes its information to decoder 1. Similar conclusions as in section 3 can be drawn from Fig. 5 and Fig. 6. The performance of tail-biting encoding/decoding method is close to terminated trellis codes when K is relatively large for a given L. Using 4-bit CRC code only will introduce significant performance degradation, while using 8-bit CRC to perform early stopping results in negligible performance loss. Using 4-bit TDMs-CRC code results in negligible performance degradation as well, but a $0.5\,dB$ coding gain compared with using 8-bit CRC code can be achieved in terms of either FER or BER. Small number of extra iterations are needed to ensure negligible performance loss. Again, using TDMs only achieves maximum coding gain, but requires more power consumption and it is not able to generate accurate enough “ACK” and “NACK” signals.

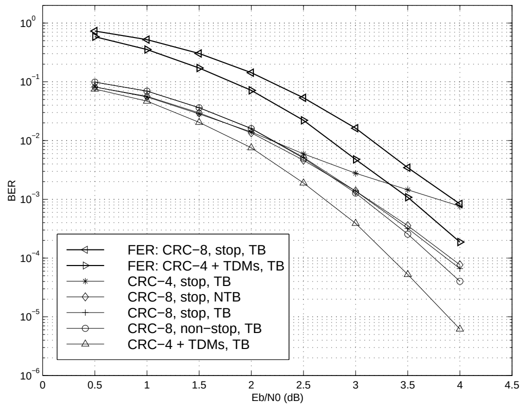

Figure 5. Performance of various early stopping algorithms (tail-biting trellis, K=5, AWGN channel)

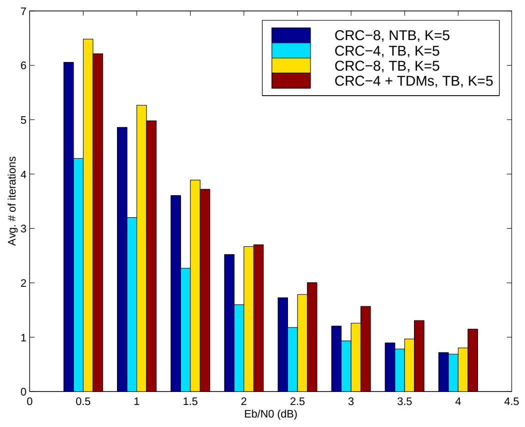

Figure 6. Average number of iterations of various early stopping algorithms (tail-biting trellis, K=5, AWGN channel)

5. TDMS-CRC ERROR DETECTION FOR CRC-EMBEDDED TRELLIS CODES

In this section, we propose a novel encoding/decoding scenario when no ARQ protocols are involved. Specifically, the short CRC code is embedded in the trellis when the turbo encoding is carried out and extracted out during the turbo decoding process. Assuming a L_{CRC}-bit CRC code, the rear $K-1$ bits out of the L_{CRC} ones will be taken as the ending state of the trellis. Denoting the state as S_{CRC}, thus we have

$$S_L = S_L^{[zi]} + S_L^{[zs]} = S_{CRC} \tag{6}$$

where $S_L^{[zi]}$ is the zero-input response $\mathbf{A}^L S_0$. Therefore, the starting state can be computed as

$$S_0 = \mathbf{A}^{-L}(S_{CRC} - S_L^{[zs]}) \tag{7}$$

where $\mathbf{A}^{-L}$ can be pre-computed and all operations are in $GF(2)$. A summary of the encoding and decoding procedure is as follows:
Encoder 1*:* Append the first $L_{CRC} - (K-1)$ CRC bits after the information vector and encode starting from zero state;
Encoder 2*:* Compute the starting state S_0 according to (7) and encode from S_0.
Decoder 1*:* Compute the forward state metrics $\alpha^{(1)}$ starting from the zero state and the backward state metrics $\beta^{(1)}$ starting from $\alpha_L^{(1)}$. Check only the first $L_{CRC} - (K-1)$ bits when carrying out the TDMs-CRC error detection method;
Decoder 2*:* Compute the forward state metrics $\alpha^{(2)}$ starting from an all $1/2^{K-1}$ vector and the backward state metrics $\beta^{(2)}$ starting from $\alpha_L^{(2)}$. Select the ending state which corresponds to the maximum element of $\alpha_L^{(2)}$ and retrieve the $(K-1)$-bit CRC code accordingly. Carry out the TDMs-CRC error detection method.

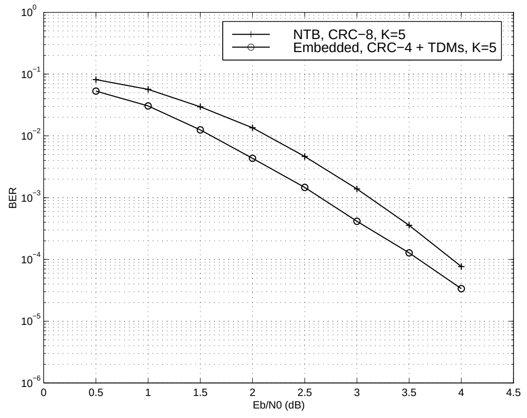

Figure 7. Performance comparison of an 8-bit CRC and embedded 4-bit TDMs-CRC early stopping schemes

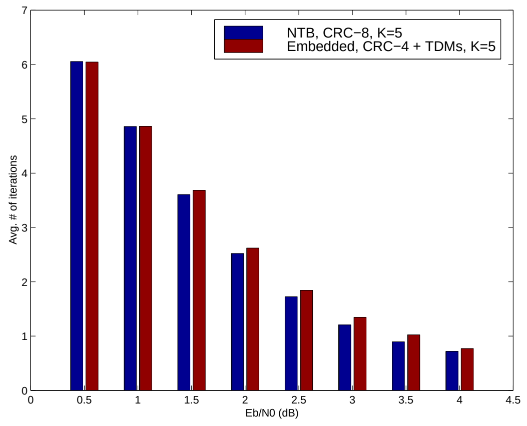

Figure 8. Average number of iterations of an 8-bit CRC and embedded 4-bit TDMs-CRC early stopping schemes

From Fig. 7 and Fig. 8, it is seen that besides a BER improvement, the proposed method introduces only very small amount of extra iterations. However, with the above decoding scheme, we are not able to provide an "ACK/NACK" to the sender. The reason is discussed in the next section.

6. TDMS-CRC ERROR DETECTION FOR ARQ

Besides saving unnecessary decoding iterations, another function of the CRC code is to decide whether the decoded frame is free of error after finishing all the pre-set number of iterations. In this case, the pre-set thresholds of TDMs are different compared with the ones used before the the final half iteration. Note that the thresholds are set in a very tight manner so that we don't mistakenly declare a frame is free of error when it is actually not. Then, for many frames, an extra half or more iterations are carried out and this explains why TDMs-CRC needs more iterations than long CRC codes. However, in the final half iteration, tight bounds will send many wrong "NACK" signals which will lead to a decrease of transmission throughput and will offset the FER improvement. So slightly relaxed thresholds are set under the constraint that wrong error detection probability is less than 0.5% for low SNR range (e.g., $< 2\,dB$) and 0.1% for high SNR range (e.g., $> 2\,dB$). For the CRC embedded scheme, it is hard to keep the miss detection rate under 0.5%. The performance of using TDMs-CRC to perform the final error detection can be found in [4].

7. CONCLUSION

In this paper, we demonstrate performance of TDMs-CRC error detection method over three different encoding/decoding scenario. A summary of the performance comparison (compared with Type (1)) is listed in Table. 1.

Type (#)	FER/BER gain	Iterations saved	Coding Overhead	Support ARQ
8-bit CRC (1)	–	most	25.6%	Yes
4-bit Term'ed (2)	$0.5\,dB$	<type (1)	13.9%	Yes
4-bit TB (3)	$0.5\,dB$	<type (1)	8.8%	Yes
4-bit Emb'ed (4)	$< 0.5\,dB$	≤type (1)	0.0%	No

Table 1. Performance Comparison of TDMs-CRC error detection method on different schemes (L=49, $L/n = 1/3$)

8. REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo codes," in *Proc. of IEEE International Conference on Communications*, pp. 1064–1070, 1993.

[2] "http://www.3gpp.org," tech. rep., 1999.

[3] Z. Wang, H. Suzuki, and K. K. Parhi, "VLSI implementation issues of Turbo decoder design for wireless applications," in *IEEE SiPS'99*, (Taiwan), pp. 503–512, 1999.

[4] Z. Chi, Z. Wang, and K. Parhi, "High-Throughput, Low-Energy FEC/ARQ Technique for Short Frame Turbo Codes," in *Proc. of 2000 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, (Istanbul), pp. 2653–2656, June 2000.

[5] H. H. Ma and J. K. Wolf, "On tail biting convolutional codes," *IEEE Trans. Comm.*, vol. 34, pp. 104–111, Feb. 1986.

[6] J. B. Anderson and S. M. Hladik, "Tailbiting MAP decoders," *IEEE JSAC.*, vol. 16, pp. 297–302, Feb. 1998.

[7] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, pp. 284–287, March 1974.

[8] C. Weiss, C. bettstetter, S. Riedel, and D. J. Costello, "Turbo Decoding with Tail-Biting Trellises," in *URSI Int. sym. on Sig. Sys. and Elec.*, pp. 343–348, 1998.