Reliability-Based Decoding for Convolutional Tail-biting Codes*

Ting-Yi Wu†, Po-Ning Chen†, Hung-Ta Pai‡, Yunghsiang S. Han‡§ and Shin-Lin Shieh¶
†Dept. of Electrical Engineering, National Chiao-Tung University, Hsinchu City, Taiwan, R.O.C.
E-mails: mavericktywu@gmail.com, poning@faculty.nctu.edu.tw
‡Graduate Institute of Communication Engineering, National Taipei University, Taipei, Taiwan, R.O.C.
E-mails: htpai@mail.ntpu.edu.tw, yshan@mail.ntpu.edu.tw.
§Dept. of Computer Science and Information Engineering, National Chi Nan University, Nantou, Taiwan, R. O. C.
¶Sunplus mMobile Inc., Hsinchu Science Park, Taiwan, R.O.C.
E-mail: shinlin.shieh@sunplusmm.com

***Abstract*—In this work, we proposed a reliability-based enhancement for the decoding of convolutional tail-biting codes (CTBC) from the observations that the decoding does not have to start from the beginning of the received vector, and that the reliability of the received vector can be used to determine a good starting position of the decoding process. Simulations show that our reliability-based enhancement can be used together with existing decoding algorithms of the CTBC to improve either their error rate or complexity.**

I. Introduction

Convolutional codes are perhaps the most popular codes in modern digital communications. In their usual encoding, a certain number of zeros is required to be appended at the end of the information sequence in order to clear the contents of encoder shift registers. For sufficiently long information sequence, the loss in code rates owing to these appended zero tail-bits can be neglected. Yet, evident code rate loss will be induced by these extra zero tail-bits when the information sequence becomes short.

In literature, several approaches have been proposed to resolve the code rate loss of convolutional zero-tail codes (CZTC), such as Direct Truncation, Puncturing [9] and Tail-biting [7], [8]. Among them, the convolutional tail-biting codes (CTBC) overcomes the loss on the code rate by directly eliminating the appended zero tail-bits. Different from the code paths of the CZTC, of which the initial and final states are definitely the all-zero state, those of the CTBC only ensure that the initial and final states are the same but change with the status of the input data. Since all possible states can be the initial state for the CTBC, the Viterbi algorithm (VA) when applied in its original form cannot guarantee to obtain the maximum likelihood (ML) solution.

Most of the decoding algorithms for the CTBC [1], [2], [4]–[9] were modified from the VA. Specifically, Wang and Ramesh extended the received vector by repeating a portion of it, and provided a tradeoff between performance and complexity by adjusting the length of the repetitive portion [9]. Shao *et al.* proposed to wrap-around several VAs, and resulted a low complexity algorithm, named wrap-around Viterbi algorithm (WAVA), with near optimal performance [7]. Shankar *et al.* developed a two-phase ML decoding algorithm of practical decoding complexity by first applying the VA, followed by the algorithm A* at the second phase [6]. In all aforementioned algorithms, the decoding always starts from the beginning of the received vector with the same initial metrics for all states.

Since the code trellis of the CTBC is circularly invariant, it is not necessary to start the decoding process at the beginning of the received vector. In [3], Handlery *et al.* proposed to use the BCJR algorithm to obtain the posteriori probability (APP) of each state, base on which the initial decoding position of the received vector is subsequently selected [3]. The extra computational complexity introduced by the BCJR algorithm however may not be compensated by the saving from the adjustment of initial decoding position especially for short information sequence. This observation leads us to the query that “Can one use a simple scheme to determine a good starting position of the received vector for use of CTBC decoding algorithms?” At this motivation, we proposed based on the reliability of the received channel bits to determine the starting position of the received vector in the decoding process. Simulations show that our proposed modification not only lower the computational complexity of the optimal two-phase decoding algorithm in [6], but improve both the error performances and decoding complexities of the suboptimal Wang and Ramesh’s decoding algorithm and WAVA.

The rest of the paper is organized as follows. Section II briefly addresses the existing decoding algorithms for CTBC. Section III introduces the concept of the proposed reliability-based enhancement. Section IV presents the simulation results. Conclusions are made in Section V.

II. CTBC Decoding: A Survey

Let $\mathcal{C}$ be an $(n, 1, m)$ CTBC with L information bits, where n is the number of output bits per information bit, and m is the memory order. The tail-biting trellis corresponding to $\mathcal{C}$ is thus of $L + 1$ levels, indexed from level 0 to level L, and has 2^m

*This work was supported by the *National Science Council* of Taiwan, R.O.C., under the project: NSC 96-2628-E-305-001.

states at each level. The tail-biting paths in this trellis are by definition constrained to have the same initial and final states, and are one-to-one correspondence to the codewords in $\mathscr{C}$. By relaxing such constraint, we denote by $\mathscr{C}_{\rm s}$ the super code of $\mathscr{C}$, of which its codewords correspond to all paths that are allowed to end at a final state different from the initial state.

In 1996, Wang and Ramesh [9] extended the received vector $\boldsymbol{r} = (r_0, r_1, \ldots, r_{N-1})$ to $\boldsymbol{r}_{\rm ext} = (r_0, r_1, \ldots, r_{N-1}, r_0, r_1, \ldots, r_{\lfloor \alpha N \rfloor -1})$, where $N = nL$, and $\alpha \in (0, 1]$ satisfies that $\lfloor \alpha N \rfloor / n$ is an integer. With equal initial metrics for all nodes at level 0, they proposed to apply the Viterbi algorithm onto the extended tail-biting trellis that is of $L + 1 + \lfloor \alpha N \rfloor / n$ levels now. The path (not necessarily a tail-biting one) with the smallest metric is then traced back after the final level is reached, and the middle L information bits are outputted. As a larger α generally implies a higher complexity but a lower WER, a trade-off in the choice of α is resulted.

The wrap-around Viterbi algorithm (WAVA) proposed in [7] also perform the VA by starting from all states at level 0 with equal initial metrics. Upon the completion of the first VA iteration over the (non-extended) tail-biting trellis with $L + 1$ levels, the path with the smallest metric will be outputted as the final decision if it is a tail-biting path. Otherwise, the VA needs to be re-executed with the initial metrics equal to the ones obtained at the end of the previous VA iteration. This process is repeated until either the resultant best path is a tail-biting path or the maximal number of iterations, I, is reached. For most CTBCs simulated, near-optimal performance can be achieved with $I = 2$.

Both algorithms introduced above are sub-optimal in performance. A straightforward optimal decoding algorithm is to separately perform the VA on all of 2^m zero-tail subtrellises. Such approach however is impractical due to its high computational complexity. Very recently, an ML decoding algorithm of practical decoding complexity has been proposed [6]. The proposed scheme has two phases. In the first phase, the VA is applied to the tail-biting trellis of the CTBC to collect the trellis information. The decoding process stops after the completion of the first phase if the minimum-metric survivor path is a tail-biting path; otherwise, based upon the trellis information obtained in the first phase, the algorithm A* is subsequently performed on all 2^m zero-tail subtrellises in the second phase to yield the ML decision. It has been shown that the decoding complexity can be reduced from 2^m VA trials of the straightforward optimal decoding to equivalently 1.3 VA trials.

III. Reliability-Based Decoding

Denote the binary codeword of $\mathscr{C}$ as $\boldsymbol{v} \triangleq (v_0, v_1, \ldots, v_{N-1}) \in \{0,1\}^N$. Define the hard-decision sequence $\boldsymbol{y} = (y_0, y_1, \ldots, y_{N-1})$ corresponding to the received vector $\boldsymbol{r} = (r_0, r_1, \ldots, r_{N-1})$ as

$$y_j \triangleq \begin{cases} 1, & \text{if } \phi_j < 0; \\ 0, & \text{otherwise,} \end{cases}$$

where $\phi_j \triangleq \ln[\Pr(r_j|v_j = 0)/\Pr(r_j|v_j = 1)]$ is the jth log-likelihood ratio. Then, it can be derived that the ML codeword $\boldsymbol{u}$ for received vector $\boldsymbol{r}$ satisfies

$$\sum_{j=0}^{N-1} (u_j \oplus y_j)|\phi_j| \le \sum_{j=0}^{N-1} (v_j \oplus y_j)|\phi_j| \qquad (1)$$

for all $\boldsymbol{v} \in \mathscr{C}$, where "$\oplus$" is the exclusive-or operation. Let the information bits corresponding to codeword $\boldsymbol{u}$ be $\boldsymbol{z} \triangleq (z_0, z_1, \ldots, z_{L-1})$. Then, $\boldsymbol{u}$ can be generated according to $\boldsymbol{z}$ through the L state transitions:

$$S_0 \to S_1 \to \cdots \to S_\ell \to S_{\ell+1} \to \cdots \to S_L,$$

where $S_\ell \in \{0, 1, \ldots, 2^m - 1\}$ and $S_0 = S_L$.

From (1), it is obvious that if $\boldsymbol{r}$ is circularly shifted left by np positions before it is fed into the decoding algorithm, where $0 \le p \le L-1$, the ML decoding output is exactly $\boldsymbol{u}' = (u_{np}, u_{np+1}, \ldots, u_{N-1}, u_0, u_1, \ldots, u_{np-1})$ since (1) can be rewritten as

$$\sum_{j=np}^{N-1} (u_j \oplus y_j)|\phi_j| + \sum_{j=0}^{np-1} (u_j \oplus y_j)|\phi_j| \le \sum_{j=np}^{N-1} (v_j \oplus y_j)|\phi_j| + \sum_{j=0}^{np-1} (v_j \oplus y_j)|\phi_j|$$

for all $\boldsymbol{v} \in \mathscr{C}$. As $\boldsymbol{u}'$ can be generated through the following state transition:

$$S_p \to S_{p+1} \to \cdots \to S_{L-1} \to S_0 \to S_1 \to \cdots \to S_p,$$

the information bits corresponding to $\boldsymbol{u}'$ are $\boldsymbol{z}' = (z_p, z_{p+1}, \ldots, z_{L-1}, z_0, z_1, \ldots, z_{p-1})$. Hence, $\boldsymbol{z}$ can be obtained through circularly right-shifting $\boldsymbol{z}'$ by p positions. This indicates that the ML decision can still be found when $\boldsymbol{r}'$ is used as the input to an ML decoding algorithm. It is thus possible to choose a proper value of p such that the complexity of an ML decoding algorithm can be reduced.

Indeed, the same trick can also be applied to a non-ML decoding algorithm to improve its performance. Recall that the Viterbi decoding for the CZTC starts its decoding from the all-zero state. Branches of the trellis of the CZTC are not merged until the mth level. On the other hand, the circular Viterbi decoding for the CTBC starts from all states with the same zero initial metrics, and branches of the trellis of the CTBC are merged right from the first level. An erroneous path elimination at the early merging stage thus may result in a high error probability. It is therefore possible to reduce the error probability by finding a better starting position.

Question is how to determine the proper starting position p. We observe from (1) that a larger $|\phi_j|$ can differentiate two bits u_j and v_j more significantly. This hints that the larger the $|\phi_j|$ is, the higher the probability that the correct decision is made. It is thus reasonable to adopt the reliability $|\phi_j|$ as a criterion to determine the starting position.

Define

$$R_p \triangleq \sum_{j=np}^{np+nm-1} w_{j-np} \cdot |\phi_j|, \qquad (2)$$

where w_i denotes the ith weight in (2). A simple and straightforward choice of $\boldsymbol{w}$ is the equal weight one, i.e., $\boldsymbol{w} = (w_0, w_1, \ldots, w_{nm-1}) = (1, 1, \ldots, 1)$. Since a larger R_p hints that one can differentiate $(u_{np}, u_{np+1}, \ldots, u_{np+nm-1})$ from $(v_{np}, v_{np+1}, \ldots, v_{np+nm-1})$ more easily, the error probability caused by early merging can possibly be reduced by setting

$$p = \arg \max_{0 \leq \bar{p} \leq L-1} R_{\bar{p}}. \tag{3}$$

By further considering that the code bit being closer to the starting position p should have higher influence on early merging, a non-increasing $\boldsymbol{w}$ (i.e., $w_0 \geq w_1 \geq \ldots \geq w_{nm-1}$) may provide a better choice of p. Along this line, we set another non-equal weight coefficient $w_i = 1/d_{\lfloor i/n \rfloor}$ with $d_0 \leq d_1 \leq \cdots \leq d_{m-1}$, where d_i is the minimum Hamming distance between two path portions, starting from different nodes at level 0, passing through different nodes at level i, but ending at the same node at level $i+1$. The non-equal weight coefficients are chosen based on the intuition that the determined Hamming distance can be generally regarded as the correcting capability for an erroneous early merging at level $i+1$.

All the algorithms introduced in Section II can be modified by adding our proposed reliability-based starting position enhancement. Other than the inclusion of our proposed enhancement, we in particular modify Wang and Ramesh's decoding algorithm in its trace back strategy in order to obtain a better error performance. The resultant algorithm is provided in the following.

Step 1. *Upon the reception of $\boldsymbol{r} = (r_0, r_1, \ldots, r_{N-1})$, determine p according to (3).*

Step 2. *Circularly left-shift $\boldsymbol{r}$ by np positions to get $\boldsymbol{r}' = (r_{np}, r_{np+1}, \ldots, r_{N-1}, r_0, r_1, \ldots, r_{np-1})$.*

Step 3. *Extend $\boldsymbol{r}'$ to $\boldsymbol{r}'_{\text{ext}} = (r'_0, r'_1, \ldots, r'_{N-1}, r'_0, r'_1, \ldots, r'_{\lfloor \alpha N \rfloor -1})$, where $0 < \alpha \leq 1$.*

Step 4. *With zero initial metrics for all nodes at level 0, perform the VA onto the extended tail-biting trellis with levels $L + 1 + \lfloor \alpha N \rfloor / n$.*

Step 5. *When the final level is reached, do the following.*

- *Among the remaining 2^m survivor paths, locate all paths associating with the same states at levels $\lfloor \alpha N \rfloor / n$ and $L + 1 + \lfloor \alpha N \rfloor / n$, if there is any, and trace back the one with the smallest metric to retain its corresponding L information bits.*
- *If however there exists no survivor path that passes through the same states at levels $\lfloor \alpha N \rfloor / n$ and $L + 1 + \lfloor \alpha N \rfloor / n$, trace back the survivor path with the smallest metric to obtain its corresponding L information bits as $(z'_0, z'_1, \ldots, z'_{L-m-1}, z'_{L-m}, \ldots, z'_{L-1})$. Replace the last m bits by the m information bit patterns $z^*_{L-m}, \ldots, z^*_{L-1}$ that make the corresponding code path lead to the same state at level $L+1+\lfloor \alpha N \rfloor / n$ as that at level $\lfloor \alpha N \rfloor / n$, and result in $(z'_0, z'_1, \ldots, z'_{L-m-1}, z^*_{L-m}, \ldots, z^*_{L-1})$.*

Step 6. *Output the information vector by circularly right-shifting the retained information vector in Step 5 by p positions.*

In Step 5, the trace-back strategy of Wang and Ramesh's decoding algorithm is modified in two places. First, our algorithm outputs the *last* L information bits instead of the *middle* L information bits. Secondly, when no survivor pathes that pass through the same states at levels $\lfloor \alpha N \rfloor / n$ and $L+1+\lfloor \alpha N \rfloor / n$ are available, the resultant information vector $(z'_0, z'_1, \ldots, z'_{L-m-1}, z'_{L-m}, \ldots, z'_{L-1})$ should contain incorrect information bits. We thus force the corresponding path to reroute to the ending state that is the same as its initial state at level $\lfloor \alpha N \rfloor / n$. Conceptually, such modification will output a legal code path in $\mathcal{C}$, which with high probability is the correct one under the circumstance.

IV. Performance Evaluation

In this section, we investigate by simulations the computational effort and the world error rate (WER) of the reliability-based decoding algorithm over the additive white Gaussian noise (AWGN) channels. We assume that the codeword is BPSK-modulated. Hence, the received vector is given by

$$r_j = (-1)^{v_j} \sqrt{\mathcal{E}} + \lambda_j,$$

for $0 \leq j \leq N-1$, where $\mathcal{E}$ is the signal energy per channel bit, and $\{\lambda_j\}_{j=0}^{N-1}$ are independent noise samples of a white Gaussian process with single-sided noise power per hertz N_0. The signal-to-noise ratio (SNR) per information bit is therefore

$$\text{SNR}_{\text{b}} = \frac{N\mathcal{E}/L}{N_0} = n\left(\frac{\mathcal{E}}{N_0}\right).$$

The binary convolutional tail-biting code under test is the $(2,1,6)$ code with generator $133, 171$ (octal) and information length $L = 48$. For such a code, the non-equal weight coefficients are given by

$$\boldsymbol{w} = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right).$$

At least 10^6 runs were performed for all simulations.

For ease of referring them, we will abbreviate the original Wang and Ramesh's algorithm and the two-phase optimal algorithm introduced in Section II by Ext and 2Phase, respectively. The Ext decoding with the modified trace-back strategy we proposed will be referred to as Ext′. The wrap-around algorithm proposed in [7] retains its name in the paper, i.e., WAVA. The modified schemes with equal weight coefficients are briefed as ew-Ext′, ew-WAVA and ew-2Phase, respectively, where "ew" stands for "equal weight". In a similar fashion, the modified schemes with nonequal weight coefficients are shortened as new-Ext′, new-WAVA and new-2Phase, respectively.

First, we examine the improvement in computational complexity of our proposed modification on 2Phase. Since 2Phase is an ML decoding algorithm, and since the first phase performs the VA, the WER performance as well as the decoding complexity in the first phase cannot be improved. Table I however shows that its decoding complexity in the second phase can be reduced by our proposed enhancement.

Next, we investigate the WER and complexity improvements of ew-WAVA and new-WAVA over the original WAVA. It is

TABLE I

COMPARISON OF AVERAGE (AVE) AND MAXIMUM (MAX) NUMBERS OF BRANCH METRIC COMPUTATIONS PER INFORMATION BIT IN THE SECOND PHASE.

SNR_b	2 dB		3 dB		4 dB	
	ave†	max	ave†	max	ave†	max
2Phase	394	140537	102	51755	28	25845
ew-2Phase	260	120374	43	46853	7	17803
new-2Phase	259	93441	42	46853	7	17803

† When the decoding process stops after the completion of the first phase, the branch metric computations per information bit in the second phase will be accounted as zero in the computation of ave.

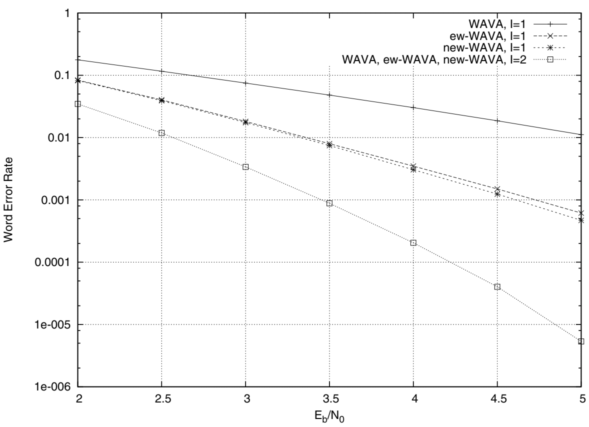

Fig. 1. Word error rates (WERs) of WAVA, ew-WAVA and new-WAVA.

observed from Figure 1 that both ew-WAVA and new-WAVA perform much better than WAVA when the maximal number of iterations is set as $I = 1$. Although no apparent improvement of our proposed modification on WAVA can be sensed when $I = 2$ (as the WAVA has already achieved near-optimal performance), the computational complexity is reduced as shown in Table II.

In the end, the effect of our proposed modification on Ext is illustrated. Note that the main advantage of Ext over near-optimal WAVA (i.e., $I = 2$ version) is that its maximum decoding complexity is $(1-\alpha)$-VA less. Figure 2 then indicates that when $\alpha = 0.75$, the Ext decoding with our proposed trace-back strategy and reliability-based starting-decoding-position adjustment can now have comparable performance to the near-optimal WAVA and ML decoding algorithm.

V. CONCLUSION AND FUTURE WORK

This work proposed a reliability-based decoding enhancement for the convolutional tail-biting code (CTBC). It can co-work with the existing algorithms surveyed to improve either performance or decoding complexity. In our study, we simply use a window size matching the memory order of the code, i.e., m, in the computation of R_p. It would be of interest to examine whether other window size can give further improvement.

TABLE II

PROBABILITIES OF WAVA, EW-WAVA AND NEW-WAVA THAT STOP AT THE FIRST ITERATION UNDER $I = 2$.

SNR_b	2 dB	3 dB	4 dB	5 dB
WAVA	0.685	0.855	0.940	0.977
ew-WAVA	0.723	0.880	0.952	0.982
new-WAVA	0.727	0.883	0.954	0.983

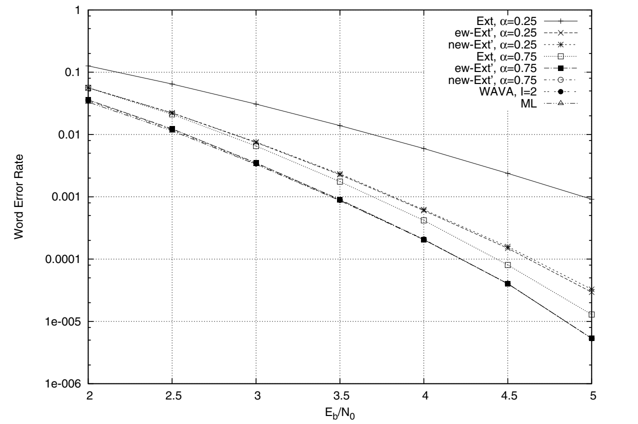

Fig. 2. Word error rates (WERs) of Ext, ew-Ext′, new-Ext′, WAVA and ML algorithm.

REFERENCES

[1] R. V. Cox and C. E. W. Sundberg, "An efficient adaptive circular viterbi algorithm for decoding generalized tailbiting convolutional codes," *IEEE Trans. Veh. Techol*, vol. 43, no. 11, pp. 57–68, February 1994.

[2] Y. S. Han, T.-Y. Wu, H.-T. Pai, P.-N. Chen and S.-L. Shieh, "Priority-first search decoding for convolutional tail-biting codes," in *2008 Information Theory and Its Applications (ISITA 2008)*, Auckland, New Zealand, December 2008.

[3] M. Handlery, R. Johannesson and V. V. Zyablov, "Booting the error performance of suboptimal tailbiting decoders," *IEEE Trans. Comm.*, vol. 51, no. 9, pp. 1485–1491, September 2003.

[4] H. H. Ma and J. K. Wolf, "On tail biting convolutional codes," *IEEE Trans. Commun.*, vol. COM-34, no. 2, pp. 104–111, February 1986.

[5] H.-T. Pai, Y. S. Han, T.-Y. Wu, P.-N. Chen and S.-L. Shieh, "Low-complexity ml decoding for convolutional tail-biting codes," *IEEE Communications Letters*, vol. 12, no. 12, pp. 883–885, December 2008.

[6] P. Shankar, P. N. A. Kumar, K. Sasidharan, B. S. Rajan, and A. S. Madhu, "Efficient convergent maximum likelihood decoding on tail-biting," available at http://arxiv.org/abs/cs.IT/0601023.

[7] R. Y. Shao, S. Lin, and M. P. C. Fossorier, "Two decoding algorithm for tailbiting codes," *IEEE Trans. Commun.*, vol. COM-51, no. 10, pp. 1658–1665, October 2003.

[8] Q. Wang and V. K. Bhargava, "An efficient maximum likelihood decoding algorithm for generalized tail biting," *IEEE Trans. Commun.*, vol. COM-37, no. 8, pp. 875 – 879, 1989.

[9] Y. P. E. Wang and R. Ramesh, "To bite or not to bite - a study of tail bits versus tail-biting," *in Proceeding of IEEE Personal, Indoor and Mobile Radio Communications*, vol. 2, pp. 317–321, October 1996.