On a Reduced-Complexity Inner Decoder for the Davey-MacKay Construction

Xiaopeng Jiao and M.A. Armand

The Davey-MacKay construction is a promising concatenated coding scheme involving an outer 2^k-ary code and an inner code of rate k/n, for insertion-deletion-substitution channels. Recently, a lookup table (LUT)-based inner decoder for this coding scheme was proposed to reduce the computational complexity of the inner decoder, albeit at the expense of a slight degradation in word error rate (WER) performance. In this letter, we show that negligible deterioration in WER performance can be achieved with an LUT as small as $7 \cdot 2^{k+n-1}$, but no smaller, when the probability of receiving less than $n-1$ or greater than $n+1$ bits corresponding to one outer code symbol is at least an order of magnitude smaller than the WER when no LUT is used.

Keywords: Insertion/deletion channel, watermark code, synchronization, LDPC code.

I. Introduction

Channels that introduce synchronization errors (insertions and deletions) have received increased attention in recent years, due in part to their application in emerging technologies such as bit-pattern media [1]. A single uncorrected insertion or deletion can lead to catastrophic consequences for a communication system, as it causes a burst of substitution errors that persists until the system is resynchronized. The Davey-MacKay (DM) construction [2] is a promising concatenated coding scheme for insertion-deletion-substitution (IDS) channels. In the IDS channel model considered therein, when bit t enters the channel, a random bit is inserted with probability P_i and then bit t remains untransmitted, is deleted with probability P_d, or is transmitted with probability $1-P_i-P_d$. In the third case, bit t is flipped with probability P_s.

The outer code in the DM scheme is a non-binary, 2^k-ary low-density parity-check (LDPC) code. The inner code has rate k/n and comprises a sparsifier and a watermark code. Synchronization is achieved by the inner decoder, which employs a BCJR-like, forward-backward algorithm. In [2], bit-level synchronization (BLS) is performed with the assumption that the sparsifier output bits are independent and identically distributed. In [3], Briffa and others dispensed with this assumption with a modified inner decoder that performs symbol-level synchronization (SLS) to yield significant improvements in word error rate (WER) performance.

Like the BCJR algorithm, the computational complexity of both inner decoders is very high. A lookup table (LUT)-based implementation of the inner decoder, as originally proposed in [4], nevertheless addresses this issue, although no thorough study of its computational requirements with and without the LUT was presented to show this. In this letter, we show that under the assumption of identical insertion and deletion rates, negligible deterioration in WER performance can be achieved with an LUT as small as $7 \cdot 2^{k+n-1}$, but no smaller, when the probability of receiving less than $n-1$ or greater than $n+1$ bits corresponding to one outer code symbol is at least an order of magnitude smaller than the WER when no LUT is used. In addition, we give a detailed complexity analysis of the BLS and SLS inner decoders with and without the LUT.

II. LUT Approach

In the DM construction, each element of a codeword of the

Manuscript received Nov. 2, 2011; revised Dec. 29, 2012, accepted Jan. 12, 2012.
This work is supported by the Singapore National Research Foundation under CRP Award No. NRF-CRP 4-2008-06. This work was done when Dr. Jiao was with the Department of Electrical & Computer Engineering, National University of Singapore, Singapore.
Xiaopeng Jiao (phone: +86 136 4923 1460, jiaozi1216@126.com) was with the Department of Electrical & Computer Engineering, National University of Singapore, Singapore, and is now with the School of Computer Science and Technology, Xidian University, Xi'an, China.
M.A. Armand (eleama@nus.edu.sg) is with the Department of Electrical & Computer Engineering, National University of Singapore, Singapore.
http://dx.doi.org/10.4218/etrij.12.0211.0461

outer q-ary LDPC code, $q=2^k$, is represented as an n-bit string by the inner encoder. Based on the received version $\boldsymbol{r}=\{r_i\}$ of a transmitted codeword, the inner decoder computes the likelihood $\Pr(\boldsymbol{r}|d_l)$ for each of the q possible values of the outer code symbol d_l according to

$$\Pr(\boldsymbol{r}\,|\,d_l)=\sum_{x_1,x_2}\begin{bmatrix}\alpha(nl,x_1)\Pr\left\{\boldsymbol{r}\,|_{nl+x_1}^{n(l+1)+x_2}\,|\,d_l\right\}\\ \beta(n(l+1),x_2)\end{bmatrix}, \tag{1}$$

where

$$\alpha(nl,x_1)\triangleq\Pr\left\{\boldsymbol{r}\,|_0^{nl+x_1},\chi_{nl}=x_1\right\}$$

and

$$\beta(n(l+1)x_2)\triangleq\Pr\left\{\boldsymbol{r}\,|_{n(l+1)+x_2}^{\rho},\chi_{n(l+1)}=x_2\right\}$$

are the forward and backward metrics, respectively, χ_i is the difference between the number of transmitted bits and the number of received bits up to the point the i-th code bit enters the channel, and $\boldsymbol{r}\,|_s^l$ denotes the subsequence $\{r_i\}_{i=s}^{l-1}$ of $\boldsymbol{r}$.

Following [4], the middle metric $\Pr\left\{\boldsymbol{r}\,|_{nl+x_1}^{n(l+1)+x_2}\,|\,d_l\right\}$ is to be precomputed for each of the q possible values of d_l and stored in an LUT to reduce the computational complexity of the inner decoder. The sequence $\boldsymbol{r}\,|_{nl+x_1}^{n(l+1)+x_2}$ consists of the received bits corresponding to the transmission of d_l. Its length ranges from 0 to $n+nI$, where I is the maximum number of bits that the channel inserts for each channel use.

The challenge is to select just a small subset of the $q(2^{n+nI+1}-1)$ possible middle metric values to be stored in the LUT, without severely affecting WER performance. In [4], the approach was to impose an upper bound δ $(2n\leq\delta\leq n+nI)$ on the length of the sequence $\boldsymbol{r}\,|_{nl+x_1}^{n(l+1)+x_2}$ for which the corresponding middle metric is to be stored, the choice of δ being dependent on the insertion and deletion error rates of the channel. While this leads to an LUT of size $q(2^{\delta+1}-1)$, which is significantly smaller than $q(2^{n+nI+1}-1)$ when $\delta=2n$ and I is large, an even smaller LUT can be obtained by only storing the middle metrics corresponding to sequences of length $n-L$ to $n+L$ for some $L\geq1$. Note that this assumes that $P_i=P_d$, since if $P_i>P_d$, for example, the LUT size can be further reduced by storing the metrics corresponding to sequence lengths from $n-L'$ to $n+L$ for some $L'<L$.

Determining the probability $P_e(L)$ that the length of the received sequence $\boldsymbol{r}\,|_{nl+x_1}^{n(l+1)+x_2}$ does not lie in the set $\{n-L,\ldots,n+L\}$ is straightforward. To this end, let P_{ie} (resp., P_{de}) denote the probability that the length of this received sequence is larger (smaller) than $n+L$ ($n-L$) so that $P_e=P_{ie}+P_{de}$. One checks that

$$P_{ie}=P_{i,L+1}+P_{i,L+2}+\cdots P_{i,nI},$$

where

$$\begin{aligned}P_{i,L+1}&=(P_i)^{L+1}+(P_i)^{L+2}P_d+\cdots+(P_i)^{L+n+1}(P_d)^n\\&\approx(P_i)^{L+1},\\P_{i,L+2}&=(P_i)^{L+2}+(P_i)^{L+3}P_d+\cdots+(P_i)^{L+n+2}(P_d)^n\\&\approx(P_i)^{L+2},\\\vdots\ &=\ \vdots\\P_{i,nI}&=(P_i)^{nI}.\end{aligned}$$

Thus, $P_{ie}\approx(P_i)^{L+1}+(P_i)^{L+2}+\cdots+(P_i)^{nI}\approx(P_i)^{L+1}$.

Similarly, $P_{de}\approx(P_d)^{L+1}$ and so

$$P_e(L)\approx(P_i)^{L+1}+(P_d)^{L+1}=2(P_i)^{L+1}.$$

For negligible performance degradation, it is intuitive that $P_e(L)$ should be smaller than the WER P_w^0 when no LUT is used. Focusing henceforth on an LUT size of $q(2^{n-1}+2^n+2^{n+1})=7\cdot2^{k+n-1}$ corresponding to the case $L=1$, we will show in section III that when $P_e(1)$ is at least an order of magnitude smaller than P_w^0, the deterioration in error performance will be insignificant.

III. WER Analysis

Consider two instances of the DM coding scheme, Code A and Code B. Code A comprises a (999, 888) LDPC code over GF(2^k=16) and an inner code of rate k/n=4/5. Code B comprises a (999, 888) LDPC code over GF(2^k=64) and an inner code of rate k/n=6/7. Figure 1 plots the WER performance of these two codes when $P_i=P_d$, with BLS and

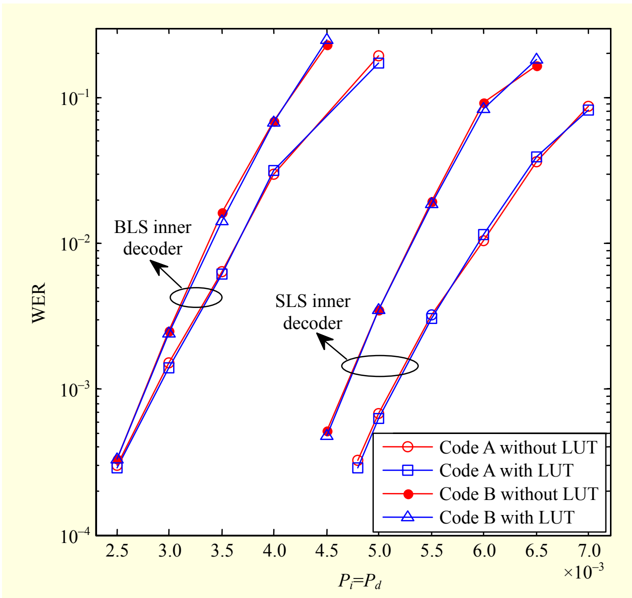

Fig. 1. WER performance of Code A and Code B with BLS and SLS inner decoders with and without LUT.

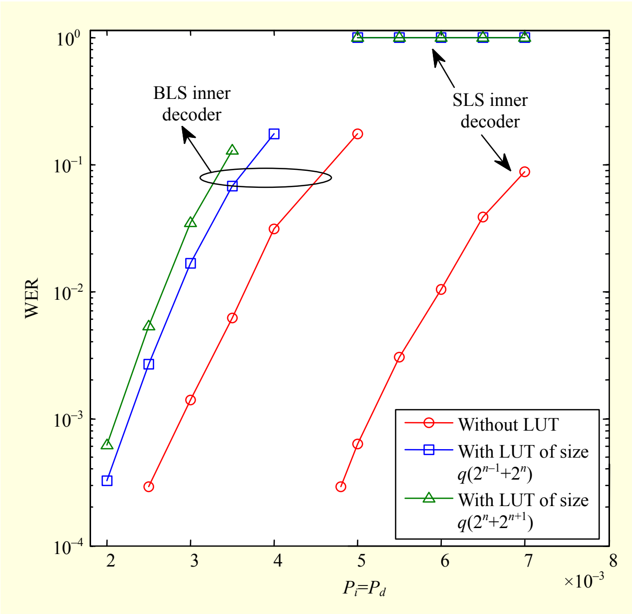

Fig. 2. WER performance of Code A with BLS and SLS inner decoders with LUT smaller than $7{\cdot}2^{k+n-1}$.

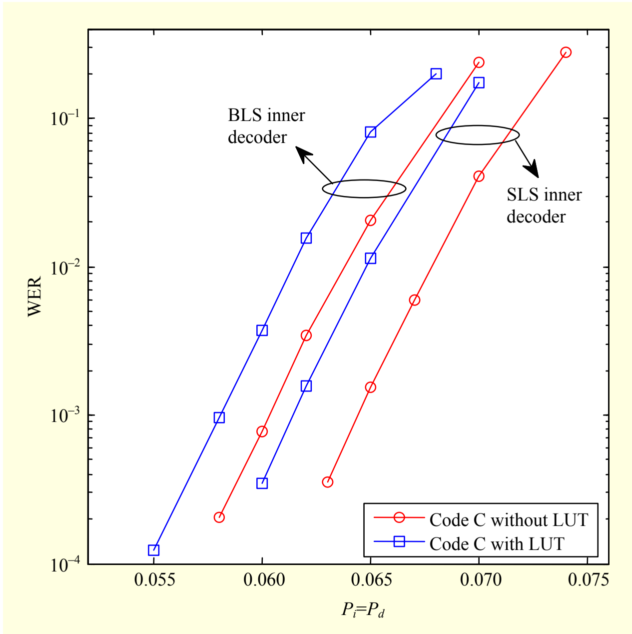

Fig. 3. WER performance of Code C with BLS and SLS inner decoders with and without LUT.

SLS as the decoding methods. As in [2], the following settings are used: P_S=0, I=5 and $|\chi_i|\leq 35$. Further, where the LUT is used, the middle metrics corresponding to sequence lengths less than n–1 and greater than n+1 are all taken to be zero.

Figure 1 shows that for both BLS and SLS and for both codes, there is hardly any deterioration in WER performance when an LUT of size $7{\cdot}2^{k+n-1}$ is used. This is not surprising since $P_e(1)$ is at least an order of magnitude smaller than P_w^0. For example, when $P_i=P_d=5\times10^{-3}$, P_w^0 is equal to 6×10^{-4} for Code A and 3.5×10^{-3} for Code B under SLS, while $P_e(1)\approx 5\times10^{-5}$. Attempting to use an LUT smaller than $7{\cdot}2^{k+n-1}$ will adversely affect WER performance. To demonstrate this, we reduce the LUT size to $q(2^n+2^{n+1})$ and $q(2^{n-1}+2^n)$. In the former (latter) case, the LUT only takes into account received sequences of length n and n+1 (n–1 and n). Figure 2 shows a marked deterioration in the resulting error performance, with the SLS inner decoder practically failing to work.

Next, we study the impact of using an LUT size of $7{\cdot}2^{k+n-1}$ on WER performance when $P_e(1)$ is greater than P_w^0. To this end, we consider another realization of the DM coding scheme, namely, Code C, which comprises a (777, 333) LDPC code over GF(2^k=8) and an inner code of rate k/n=3/6. Figure 3 shows the WER performance of this code with and without the proposed LUT under the following settings: P_S=0, I=5, and $|\chi_i|\leq 160$. Unlike the situation depicted in Fig. 1, an LUT size of $7{\cdot}2^{k+n-1}$ is now not sufficient to achieve negligible performance degradation in this case. This is not surprising because, now, $P_e(1)$ is greater than P_w^0. For example, when $P_i=P_d=6\times10^{-2}$, P_w^0 is equal to 7×10^{-4} under BLS while $P_e(1)\approx 7.2\times10^{-3}$. Nevertheless, unlike the situation depicted in Fig. 2 for the SLS inner decoder with LUT, the deterioration in error performance here is not catastrophic. Hence, if storage complexity is a key concern, one might choose to accept such performance loss in exchange for smaller memory requirements to store the proposed LUT. Alternatively, one could choose a larger LUT based on the analysis presented in section II to reduce the degradation in performance, albeit at the expense of increased storage complexity.

In all our computer simulations, we use the log-domain belief-propagation decoder [5] to decode the outer code, with the maximum number of iterations set to 50.

IV. Complexity Analysis

A straightforward counting exercise yields the computational complexity of the BLS and SLS inner decoders with and without an LUT, which we summarize in Table 1. There, N is the length of the outer code, $x_{\max}$ is the maximum $|\chi_i|$, and M_{RA} (M_{RM}) denotes the number of real additions (multiplications) incurred by a single middle metric computation. One checks that

$$M_{\mathrm{RA}}=\frac{n(n-1)}{2}(I+1)^2 \quad \text{and} \quad M_{\mathrm{RM}}=\left[\frac{n(n-1)}{2}(I+1)+n\right](I+2).$$

With Table 1, it is easy to see that the LUT-based approach significantly reduces the computational complexity of the inner decoder. For example, for Code A and the channel parameters used in our simulations, the BLS inner decoder without an

Table 1. Computational complexity of BLS and SLS inner decoders with and without LUT.

Inner decoder	No. of real additions	No. of real multiplications
BLS without LUT	$2(2x_{\max}+1)(I+1)Nn+(2x_{\max}+1)^2 Nq(M_{RA}+1)$	$2(2x_{\max}+1)(I+2)Nn+(2x_{\max}+1)^2 Nq(M_{RM}+2)$
BLS with LUT	$2(2x_{\max}+1)(I+1)Nn+(2x_{\max}+1)^2 Nq$	$2(2x_{\max}+1)(I+2)Nn+2(2x_{\max}+1)^2 Nq$
SLS without LUT	$3(2x_{\max}+1)^2 Nq(M_{RA}+1)$	$(2x_{\max}+1)^2 Nq(3M_{RM}+4)$
SLS with LUT	$3(2x_{\max}+1)^2 Nq$	$6(2x_{\max}+1)^2 Nq$

LUT requires 343 (222) times more real additions (multiplications) than the LUT approach, while the SLS inner decoder without an LUT requires 361 (228) times more real additions (multiplications) than the LUT approach.

V. Conclusion

To summarize, we have shown that the LUT approach significantly reduces the computational complexity of the inner decoder. Further, we have shown that under the assumption of identical insertion and deletion rates, negligible performance degradation can be achieved with an LUT as small as $7{\cdot}2^{k+n-1}$, but no smaller, when $P_e(1)$ is at least an order of magnitude smaller than P_w^0. Moreover, based on the analysis presented in section II, when this condition is not satisfied, a larger LUT can be chosen to reduce the degradation in performance, albeit at the expense of increased storage complexity.

Acknowledgements

We thank the anonymous reviewers for their thoughtful comments, which helped improve the quality of this letter.

References

[1] J. Hu et al., "Achievable Information Rates for Channels with Insertions, Deletions, and Intersymbol Interference with i.i.d. Inputs," *IEEE Trans. Commun.*, vol. 58, no. 4, Apr. 2010, pp. 1101-1111.

[2] M. Davey and D.J.C. MacKay, "Reliable Communications over Channels with Insertions, Deletions, and Substitutions," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, Feb. 2001, pp. 687-698.

[3] J.A. Briffa, H.G. Schaathun, and S. Wesemeyer, "An Improved Decoding Algorithm for the Davey-MacKay Construction," *IEEE Int. Conf. Commun.*, Cape Town, South Africa, May 2010.

[4] X. Jiao and M.A. Armand, "Interleaved LDPC Codes, Reduced-Complexity Inner Decoder and an Iterative Decoder for the Davey-MacKay Construction," *IEEE Int. Symp. Inf. Theory*, St. Petersburg, Russia, Aug. 2011.

[5] H. Wymeersch, H. Steendam, and M. Moeneclaey, "Log-Domain Decoding of LDPC Codes over GF(q)," *IEEE Int. Conf. Commun.*, Paris, France, June 2004.