Bi-directional SOVA Decoding for Turbo-codes

Jinghu Chen, Marc P. C. Fossorier, Shu Lin
Department of Electrical Engineering
University of Hawaii
2540 Dole Street
Honolulu, Hawaii 96822
USA
Email: jinghu@spectra.eng.hawaii.edu
marc@spectra.eng.hawaii.edu
slin@spectra.eng.hawaii.edu

Chengqi Xu
Department of Information Engineering
Nanjing University of Posts & Telecommunications
Nanjing 210003
China
Email: xucq@em.njupt.edu.cn

Abstract: In this paper, we present a bi-directional SOVA-based decoding scheme for turbo-codes, which combines the soft-outputs provided by both forward and backward SOVA decodings. The complexity of this bi-directional SOVA decoding only doubles that of the conventional SOVA proposed by Hagenauer and Hoeher, with very limited additional time delay for turbo decoding. The potential improvements achieved by bi-directional SOVA over conventional SOVA are explained, and simulation results show that bi-directional SOVA can achieve about the same performance as the Max-Log-MAP algorithm in turbo decoding with much less computational complexity.

Index Terms: SOVA, MAP decoding, Max-Log-MAP decoding,

Turbo decoding, block codes.
I Introduction

Turbo-codes, proposed in [1], can achieve a performance close to Shannon limit if the code length is enough long. A turbo-encoder consists of two parallel concatenated recursive systematic convolutional (RSC) codes encoders, which are connected by an interleaver. On the decoder side, an iterative decoding is adopted such that the two constituent codes can help each other. The iterative process requires the decoding algorithms being able to make use of a priori information as well as deliver reliability information for each decoded information bit in addition to the hard decision. The symbol-by-symbol maximum a posteriori (MAP) algorithm, introduced in [2], is an optimal decoding algorithm for minimizing the bit error probability of convolutional codes. It can readily provide reliability values for each decoded bit in term of logarithmic likelihood ratio (LLR). It can be adopted in a straightforward way for turbo decoding [1][3][4]. Unfortunately, the MAP algorithm needs large memory size as well as a large number of computations, which complicate its hardware implementation. The Soft Output Viterbi Algorithm (SOVA), proposed in [5], is a soft-input soft-output (SISO) Viterbi algorithm and can use a priori information after some improvement [4][6]. It can be implemented for turbo decoding with moderate complexity and consequently, received favorable consideration from the viewpoint of implementation. However, the coding gain achieved by the SOVA is generally about 0.7 dB less than that of MAP decoding for turbo decoding [7].

The MAP algorithm can be approximated by the Max-Log-MAP algorithm [4][7]. Operating in the logarithmic domain and with some approximations, the Max-Log-MAP algorithm avoids logarithmic and exponential operations at the expense of a few tenths of a dB degradation in error performance. Compared with the SOVA, it still remains harder to implement in hardware. Both Max-Log-MAP and SOVA can be viewed as sub-optimal reduced complexity versions of the MAP algorithm.

Some research has been done to improve the performance of SOVA for turbo decoding. In particular, another version of SOVA, based also on a modified VA and first proposed in [10], can be used in turbo decoding [8], and it was shown in [9] that with this modification, SOVA becomes equivalent to Max-Log-MAP. But this algorithm is significantly harder and not as flexible to implement compared with the original SOVA of [5]. In [8][11], normalizations are proposed to combat the over-estimation of soft outputs delivered by the SOVA. Significant improvements can be achieved by this approach, but the normalization factors strongly depend on the bit error rate (BER).

In this paper, we propose a decoding scheme based on the original SOVA of [5]. This decoding scheme employs an independent backward SOVA decoding in addition to the forward one. The hardware complexity only doubles that of the original one, with very limited additional delay for turbo decoding. After a brief review of SISO decoding algorithms in Section II, the improvements achieved by the bi-directional SOVA with respect to the original SOVA are analyzed in Section III. Simulation results are finally given in Section IV. Interestingly, these results not only show that bi-directional SOVA significantly outperforms the original SOVA for
turbo decoding, but surprisingly can also occasionally slightly outperform the iterative Max-
Log-MAP algorithm.

II A Brief Review of MAP, Max-Log-MAP and SOVA Algorithms

In this section, some conceptual rather than detailed descriptions of the MAP, Max-Log-MAP
and SOVA decoding algorithms are given.

Let \( \mathbf{uc} \) be a transmitted codeword mapping the information sequence \( \mathbf{u} = (u_1, \cdots, u_K) \), with \( c_i \in \{-1, +1\} \) and \( u_j \in \{0,1\} \) for \( i = 1, \cdots, N \) and \( j = 1, \cdots, K \). After transmission over an additive white Gaussian noise (AWGN) channel, the corresponding received vector is \( \mathbf{r} = \mathbf{c} + \mathbf{n} \), where \( \mathbf{n} \) is an \( N \)-dimensional Gaussian vector with zero-mean vector and
diagonal covariance matrix \( \frac{N_0}{2} \mathbf{I}_N \). For \( i = 1, \cdots, K \), the MAP algorithm evaluates the LLR:

\[
\Lambda_i = \ln \left( \sum_{\mathbf{c} \in \Omega_1(i)} P(\mathbf{c} | \mathbf{r}) \right) - \ln \left( \sum_{\mathbf{c} \in \Omega_0(i)} P(\mathbf{c} | \mathbf{r}) \right)
\]  

where \( \Omega_1(i) = \{ \mathbf{c} = s(\mathbf{u}): u_i = 1 \} \), \( \Omega_0(i) = \{ \mathbf{c} = s(\mathbf{u}): u_i = 0 \} \). For an RSC code, each codeword \( \mathbf{c} \) is represented by a path in the code trellis diagram. Hence, for a given \( i \), the MAP algorithm divides the set of paths into two subsets, \( \Omega_1(i) \) and \( \Omega_0(i) \), where \( \Omega_1(i) \) consists of all the paths corresponding to the information bit \( u_i = 1 \), and \( \Omega_0(i) \) consists of all the paths corresponding to \( u_i = 0 \). Then it calculates the sum of the probabilities of all the paths in each set, conditioned on \( \mathbf{r} \), and finally computes \( \Lambda_i \). In [2], an efficient bi-directional trellis-based method to evaluate (1) is presented.

Based on the approximation \( \ln \left( \sum_{\mathbf{c} \in \Omega_1(i)} \delta_i \right) = \ln \left( \max_{\mathbf{c} \in \Omega_1(i)} \delta_i \right) \), (1) can be approximated by:

\[
\tilde{\Lambda}_i = \ln \left( \max_{\mathbf{c} \in \Omega_1(i)} P(\mathbf{c} | \mathbf{r}) \right) - \ln \left( \max_{\mathbf{c} \in \Omega_0(i)} P(\mathbf{c} | \mathbf{r}) \right)
\]  

For each information bit \( u_i \), the Max-Log-MAP algorithm approximates \( \Lambda_i \) by \( \tilde{\Lambda}_i \) based on (2).

As a result, for each \( i \), the Max-Log-MAP algorithm considers only two paths to evaluate \( \tilde{\Lambda}_i \).

One path is selected from \( \Omega_1(i) \) and has the largest probability conditioned on \( \mathbf{r} \); we call it the best path (or most likely) in \( \Omega_1(i) \). The second path is selected from \( \Omega_0(i) \) with the largest probability conditioned on \( \mathbf{r} \); we call it the best path in \( \Omega_0(i) \). It is readily seen that one of the two best paths must be the maximum likelihood (ML) decoding path.

The SOVA is based on modifying the Viterbi algorithm (VA) so that a reliability value for each hard decision can be provided. It is well-known that the probability of path-\( l \), conditioned on \( \mathbf{r} \), is proportional to \( e^{M_l} \), where \( M_l \) is the metric of path-\( l \) computed by the VA. For each \( i \),
consider two paths, say path-1 and path-2, associated with $c_1$ and $c_2$, respectively, with one path in $\Omega_1(i)$ and the other in $\Omega_0(i)$. Then the metric difference between the two paths is:

$$\Delta = M_1 - M_2 = \ln e^{M_1} - \ln e^{M_2} \propto \ln P(c_1 | r) - \ln P(c_2 | r)$$  \hspace{1cm} (3)

Comparing (3) with (2), we observe that $|\Delta|$ can also be chosen as an approximated measure of $|\Lambda_i|$. The SOVA searches for the ML path in the code trellis, and records all $\Delta$ values between the ML path and the paths remerging with it. The approximated values of $|\Lambda_i|$ can be updated by trace-back.

For each $i$, both Max-Log-MAP and SOVA decodings consider only two paths to calculate the soft outputs, one from $\Omega_1(i)$ and the other from $\Omega_0(i)$. However, the Max-Log-MAP algorithm always chooses the best path in each set, while the SOVA only guarantees to find one best path (the ML path). The other path is not necessarily the best one, due to the fact that the best path with opposite $u_i$ to the ML path may be discarded before it remerges with the ML path. For example, in Fig-1 (a), suppose: (a) path-1 is the ML path with information bit $u_1 = 0$; (b) path-3 is the best path with $u_3 = 1$; and (c) path-2 survives at time-$i$ and remerges with path-1 later. In this case, the SOVA does not consider path-3, as it has been discarded at time-$i$. It follows that in general, the soft outputs delivered by the SOVA are over-estimated with respect to that of the Max-Log-MAP decoding. These inaccurate reliability values degrade the performance of the iterative turbo decoding.

### III Bi-directional SOVA for Turbo-decoding

Given a received sequence $r = (r_1, \ldots, r_N)$, both the VA and the SOVA can start decoding simultaneously the first received value $r_1$ and the last received value $r_N$, from both ends of the trellis. The hard decisions of the forward and backward decodings are the same as in both cases, as the same ML path is obviously chosen. We also find that the reliability values of the forward and backward SOVA's have the same “quality”, because if we replace all the forward SOVA decoders in turbo decoding by the backward ones, the simulations show there is no degradation nor improvement in performance, as expected. As a result, the backward Viterbi algorithm has no advantages over the forward one if they are processed independently. However, we notice there are differences between the soft output values delivered by the forward and backward SOVA, which suggests to exploit these differences to improve the soft outputs of the SOVA.

The differences in the soft outputs between forward and backward SOVA's result from the different selections of paths in calculating the reliability $|\Delta|$. In Section II, we show that for a given $i$, the original forward SOVA may not choose one of the two best paths, either in $\Omega_1(i)$ or $\Omega_0(i)$, because this path is discarded before it remerges with the ML path. However, the best path discarded by the forward SOVA can survive until it remerges with the ML path in a
backward SOVA, and the resultant $\Delta$ provides a more reliable value about $u_i$. Unfortunately, there are also cases in which the SOVA will fail to choose the best path in both directions. These two possible cases are illustrated in Fig-1 (a) and (b), respectively. In Fig-1 (a), path-3 discarded by the forward SOVA at time-$i$ can survive until it remerges with the ML path (i.e. path-1) in the backward SOVA. However, in Fig-1 (b), path-3 is discarded in both ways. Also even if the best path does not survive, the backward SOVA may still find another path better than that in forward SOVA, and get a more accurate soft-output for $u_i$. If the reliability value $|\Lambda^b_i|$ of $u_i$ in backward SOVA is less than the value $|\Lambda^f_i|$ obtained in forward SOVA, i.e., $|\Lambda^b_i|<|\Lambda^f_i|$, we simply substitute $\Lambda^b_i$ for $\Lambda^f_i$ to improve the soft-output of the SOVA. This decoding is referred to as bi-directional SOVA (bi-SOVA). Importantly, the event depicted in Fig-1 (a) is generally more likely than that in Fig-1 (b) as it involves paths of shorter lengths. In general, we can expect the bi-SOVA to provide reliability values close to those of the Max-Log-MAP since if no event depicted in Fig-1 (b) occurs, both algorithms become equivalent [8].

This idea can be applied to turbo decoding, because the delay resulting from interleaving is unavoidable in turbo decoding. In Fig-2, the corresponding modified structure of the decoding module for one iteration is given. Two backward SOVA decoders, B-SOVA1 and B-SOVA2 are added to the two conventional SOVA decoders. Because of the parallel structure, the forward and backward SOVA can be carried out at the same time after the initial iteration, so that the total extra delay is only that spent on comparing and selecting the soft-values, which is obviously very small.

One thing worth mentioning is that because of interleaving, the trellis of the second constituent code can not always be terminated to the all-0 state. So in B-SOVA2, the initial state is unknown, and the initial metrics of all states are set to zero.

IV Simulation Results:

Simulations have been conducted to compare the performance of bi-SOVA with those of MAP, SOVA and Max-Log-MAP in turbo decoding. The conditions of the simulations are the following. We used two component codes, each with rate $R=1/2$ and generator polynomial $G=[37, 21]$. The interleaver size is $N=1024$, with the so-called non-uniform interleaver introduced in [1]. All algorithms assume no finite-length decoding window.

Fig-3 (a) depicts the bit error performances with these four decodings after the first iteration. We observe that bi-SOVA is better than SOVA in performance, but slightly worse than Max-Log-MAP. This is in accordance with our analysis of Section III, since bi-SOVA can improve the soft-outputs of SOVA, and this benefit will be reflected in the performance after the first decoding iteration.

Fig-3 (b) depicts the BER's with these four decodings after 8 iterations. We find that bi-SOVA can achieve about 0.35dB coding gain with respect to the original SOVA at the BER=$10^{-4}$, and an even higher coding gain at the BER=$10^{-5}$. Surprisingly, after 8 iterations, the performance of bi-SOVA is even slightly better than that of Max-Log-MAP at low SNRs, and almost the same as
that of Max-Log-MAP when the SNR is as high as 2.5dB (i.e. in the error floor region). A similar behavior was observed with other convolutional codes and other interleaver sizes. In Fig-4, the curves of the first four iterations and the last iteration are represented. From the figure, we observe that the iteration at which the bi-SOVA starts outperforming the Max-Log-MAP depends on the SNR. A justification of this result is beyond the scope of this paper.

V Conclusions
In this paper, we analyzed the differences between the soft outputs of the forward and backward SOVA decodings, and proposed a SOVA-based bi-directional decoding scheme. This doubles the complexity of the original SOVA, but simulations show that bi-directional SOVA can achieve about the same error performance as Max-Log-MAP in the turbo decoding, or even slightly outperform this latter occasionally. This approach extends in a straightforward way to non-binary codes, although additional considerations may be needed for bi-directional SOVA to approach the error performance of Max-Log-MAP. Also, the effects of finite length window for bi-directional SOVA worth investigation for a practical viewpoint.
References:
Fig-1 Possible Cases of Path-selecting in Decoding with SOVA
\( z^{(n-1)} \)

\( \mathbf{y}^1 \)

\( \mathbf{y}^2 \)

\( \hat{\mathbf{u}}^{(n)} \)

\( z^{(n)} \)

\[ \min(\ ) \]

\[ \text{DI} \]

\[ \text{delay} \]

\[ I \] : Interleaving

\[ \text{DI} \] : De-interleaving

**Fig-2** One Decoding Module of Bi-directional SOVA
Fig-3 Performances of the SOVA, Bi-SOVA, Max-Log-MAP and MAP after: (a) one iteration; (b) eight iterations.
Fig-4 Iterative Decoding Processes of Bi-SOVA and Max-Log-MAP