

A New Method of Improving SOVA Turbo Decoding for AWGN, Rayleigh and Rician Fading Channels

Stylianios Papaharalabos, Peter Sweeney, Barry G. Evans
Centre for Communication Systems Research (CCSR)
University of Surrey
Guildford, Surrey, GU2 7XH, UK
s.papaharalabos@surrey.ac.uk

Abstract—The iterative soft output Viterbi algorithm (SOVA) is a sub-optimum algorithm when it is used to decode turbo codes. By normalizing its extrinsic information we get a performance improvement compared to the standard SOVA. In particular, if the extrinsic information is increased in the last decoding iteration, an additional coding gain improvement is noticed. For example, this is 0.25 dB for a frame length of 1000 bits in the additive white Gaussian noise (AWGN) channel as well as in an uncorrelated Rician fading channel at bit error rate (BER) of 10^{-6} . Also, this normalized SOVA is only about 0.25 dB worse than a turbo decoder using the Log-MAP algorithm, both in the AWGN channel and in an uncorrelated Rayleigh fading channel at BER of around 10^{-6} . Furthermore with an 8-state component code, a frame length of 500 bits performs 0.125 dB better than a 16-state Bidirectional (Bi)-SOVA turbo decoder at BER of 10^{-5} in the AWGN channel.

Keywords—turbo codes, iterative SOVA decoder, extrinsic information

I. INTRODUCTION

Turbo codes are among the most powerful error correcting codes approaching very close to the Shannon's limit with a very large interleaver size [1]. To decode such codes, the maximum a posteriori (MAP) algorithm [2] was modified to be suitable for decoding recursive systematic convolutional (RSC) codes in an iterative process.

However, in practice the MAP turbo decoder is too complex to be implemented due to the large number of multiplications and the need of non-linear functions. For that reason, two simplified versions of it were proposed in the past, namely Log-MAP and Max-Log-MAP [3]. The latter algorithm is sub-optimum in terms of bit error rate (BER) performance but easier to be implemented, as it requires only additions and the max operator.

Another sub-optimum algorithm that is suitable for turbo decoding is the soft output Viterbi algorithm (SOVA). It is a modified Viterbi algorithm (VA) that produces, in addition to the most likely path sequence, a reliability value of each estimated bit [4]. It was found that the iterative SOVA is 0.7 dB worse than the MAP algorithm at BER of 10^{-4} [3]. This is largely because the SOVA considers only two path sequences

to update its soft output, namely the survivor and the concurrent path sequences.

A first attempt to improve the SOVA was reported in [5] with two proposed modifications, so as to correct its soft output and to follow a Gaussian distribution. In the first modification, the extrinsic information is normalized by multiplying with a correcting factor c that depends on the variance of the decoder output, while in the second one (that is less effective) the correlation in the decoder input is eliminated by inserting two more correcting coefficients. Another attempt to improve the SOVA was described in [6] where the reliability of the soft output is limited to a small range of values. In [6] was also described the SOVA updating soft output rule by *Battail* and it was later shown that this is equivalent to the Max-Log-MAP algorithm [7]. Finally, the Max-Log-MAP turbo decoder was improved in [8] by following a normalization method similar to [5] but keeping constant the correcting factor c .

In this paper, we propose a new normalization scheme based on the observations of [5], [6], [8] using *Hagenauer's updating rule* (HR) [4]. We keep the same correcting factor c in every iteration step as in [8], which is constant and does not depend on the variance of the decoder output as it does in [5]. We refer to this as *norm1* method. In addition, due to the observed small average absolute reliability values after the first iterations compared to the ones by the *Battail's updating rule* (BR) [6], we increase the correcting factor c only in the last decoding iteration. This new normalization method is referred to as *norm2*. By doing this, a coding gain improvement of 0.25 dB at BER of around 10^{-6} is achieved compared to the first normalization scheme (*norm1*) for specific turbo encoding parameters and correcting factors. The complexity that is added to the turbo decoder is relatively small, just one more multiplication of the extrinsic information with a constant factor, per component decoder and decoding iteration.

In section II the basic procedure of the iterative SOVA is described, followed by the proposed improving algorithm in the next session. In section IV simulation results are shown and are compared to existing references. Finally, conclusions are drawn in section V.

II. THE ITERATIVE SOVA

A. A Basic Description of the SOVA

The SOVA is based on the classical process of the VA, followed by an updating rule to produce soft outputs on the estimated bit sequence.

At an instant time k the VA finds the *survivor* path, which is the path that has the smallest path metric between all the metrics of paths that enter each state. The path metric is the summation of all the branch metrics of a state sequence. To do this, we define the *branch metric* between two states of a path, based on the squared Euclidean distance, as

$$\lambda_k = \sum_{i=0}^{n-1} (y_{k,i} - x_{k,i})^2 \quad (1)$$

where $1/n$ is the code rate, n is the codeword size, $x_{k,i}$ is the i -th transmitted symbol (or bit assuming BPSK modulation) and $y_{k,i}$ is the corresponding received value at the receiver.

When the process of the VA has finished, only two paths are needed for the SOVA, the survivor or the best path and its strongest competitor (*concurrent*) path, that is the path which had diverged at a past time $k-v$ and merged to the same state as the survivor path at time k . Their path metric difference Δ is also stored and the process of the SOVA is starting from the last state of the trellis by tracing back.

To produce the bit Log Likelihood Ratio (LLR) values, we first initialize all the reliabilities of the survivor sequence to $L_s = +\infty$ and then we update as

$$L_s = \min(L_s, \Delta), \quad \text{if } u_s \neq u_c \quad (2)$$

only when the estimated bit of the survivor path sequence (u_s) and the corresponding estimated bit of the concurrent path sequence (u_c) at an instant time k are different each other.

In addition to the above rule (HR), the BR updates the reliability values in case of $u_s = u_c$ as well by

$$L_s = \min(L_s, \Delta + L_c), \quad \text{if } u_s = u_c \quad (3)$$

where L_c is another reliability value representing the concurrent path. That additional updating rule makes the BR superior compared to the HR.

We also note that the noise variance of the channel is not required when calculating the bit LLR values. That leads to a simpler implementation method.

B. The SOVA Turbo Decoder

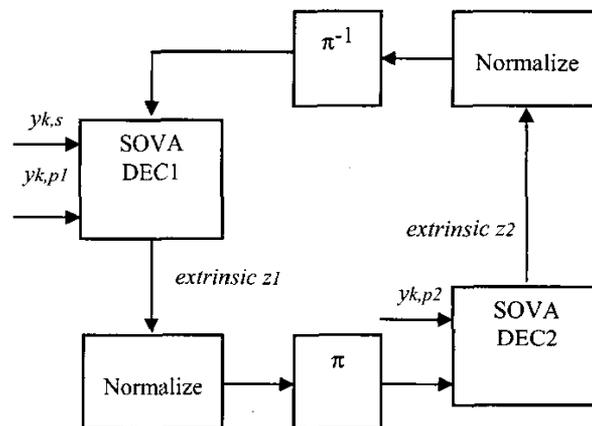


Fig. 1. SOVA turbo decoder block diagram.

The basic turbo decoding process is shown in Fig. 1 where the key issue is the 'effective' use of the extrinsic information that is passed from one component decoder to the other in every iteration step.

The concept of normalizing the extrinsic information had already been introduced in [1] where the MAP algorithm was used. In that case, the extrinsic information needed to be divided by a proper number to avoid increasing BER at low signal-to-noise ratio (SNR) values. This kind of normalization can be avoided by passing LLR values, as opposed to channel values between the two component decoders [9]. A more detailed description of the normalization of the extrinsic information when either the MAP or the SOVA iterative decoding is used can be found in [10].

III. IMPROVING THE SOVA

A. A Normalization Approach of the SOVA Output

In [6] it was proposed to limit the reliability values of the SOVA to a smaller range by defining an optimum threshold value. This was observed by plotting both the average absolute reliability value of the soft output and the average absolute extrinsic information in two cases. In the first one, the BR updating rule was followed, while in the second case the HR updating rule was followed.

In more detail the BR gives better performance, as the reliability values are smaller than the HR during the first iterations. That is the scope of this normalization method; to reduce the over-estimated reliability values of the HR that are observed in the first iterations by limiting the soft output values into a smaller range.

B. The Proposed Method

Assume that the total number of decoding iterations is N . The extrinsic information that is passed from the one component decoder to the other one can be normalized, if we multiply by a proper value m such that

$$m[i] = c, \quad 1 \leq i \leq N \quad (4)$$

We refer to this method as *norm1* method. If we further increase the normalization factor m in the last decoding iteration, we get the *norm2* method. That means

$$m[i] = c, \quad 1 \leq i \leq N-1 \quad (5)$$

and
$$m[i] > c, \quad i = N \quad (6)$$

What we propose as *norm2* is the inverse of the normalization method that is was described in [6]. As far as the BR produces soft outputs that are greater than the HR after the first iterations, we can easily correct the HR soft output by increasing the extrinsic information during the last decoding iteration only. According to the plots in [6], there should be at least four decoding iterations to achieve a noticeable coding gain improvement.

The search for good m values is an exhaustive process and it is based on computer simulation tests. They depend on the turbo encoding parameters such as the code memory length and the code rate. An example can be found in [11] where the *norm1* method was used for a specific turbo encoder. In case of the *norm2* method, the best m value in the last decoding iteration had to be around twice the m value in the previous decoding iterations.

IV. SIMULATION RESULTS

A. The AWGN Channel

In this section we consider the case of the additive white Gaussian noise (AWGN) channel. This is the case of a communication link between a fixed terminal and a satellite. Two different frame lengths of bits, 500 bits and 1000 bits are considered in order to compare with existing results. In all the simulations 50 millions bits are transmitted, while the rest of the parameters are shown in Table I.

TABLE I SIMULATION PARAMETERS

| | |
|---|---------------------------------------|
| Generator polynomials (recursive, feed-forward) | (13,15) _o , 8-states, 3GPP |
| Interleaver type | pseudo-random |
| Code rate | 1/3 |
| Modulation type | BPSK |
| Channel type | AWGN/Uncorrelated Rice fading |
| Decoding iterations | 8 |
| SOVA updating rule | Hagenauer (HR) |

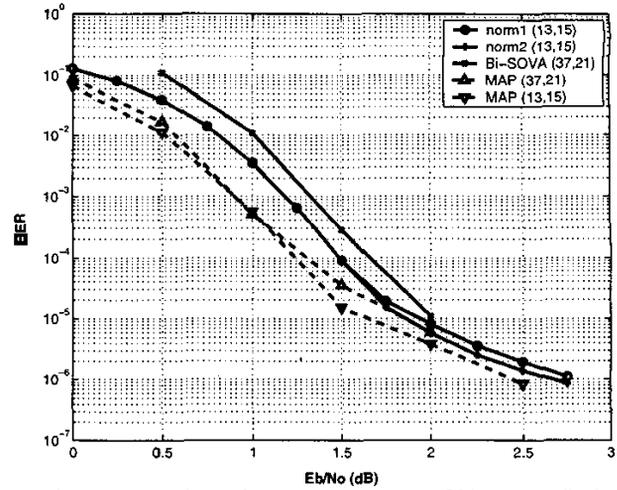


Fig. 2. Turbo coding performance with two types of SOVA normalization and reference performance comparison in the AWGN channel after 8 decoding iterations and 500 bits frame.

In Fig. 2 is plotted the turbo coding performance considering both the *norm1* and the *norm2* methods. At BER of 10^{-6} the coding gain improvement using the *norm2* instead of the *norm1* method is 0.125 dB. This was explained in the previous section where the soft output decoder values in the *norm2* method are increased to approximate the BR updating rule.

In the same figure is also plotted the turbo coding performance of the classical (37,21)_o, 16-state turbo encoder using both the Bi-(Bidirectional) SOVA and the MAP algorithm [12]. The other parameters in that reference are exactly the same as in Table I. It is noticed that both the *norm1* and the *norm2* methods perform better than the Bi-SOVA at any BER. For example the *norm2* method performs 0.125 dB better than the Bi-SOVA at BER of 10^{-5} . Also, the *norm2* method performs exactly the same as the MAP algorithm at BER lower than 10^{-5} . When the comparison is done with the same 8-state turbo encoder but with MAP iterative decoding, the *norm2* method is only 0.125 dB worse at BER of around 10^{-6} .

We also assume a second frame length equal to 1000 bits and 6 decoding iterations. In this case, the *norm2* method performs 0.25 dB better than the *norm1* method at BER of 10^{-6} (Fig. 3).

In the same figure it is noticed that at BER of 10^{-5} both the two normalization schemes approach the second normalization method of Fig. 5 in [13], although in this reference a different interleaver was used, namely the *prime interleaver* which has a superior performance compared with the pseudo-random interleaver. For example it is 0.2 dB better at BER of 10^{-6} using a 1280 bits frame over a frequency selective Rayleigh fading channel [14].

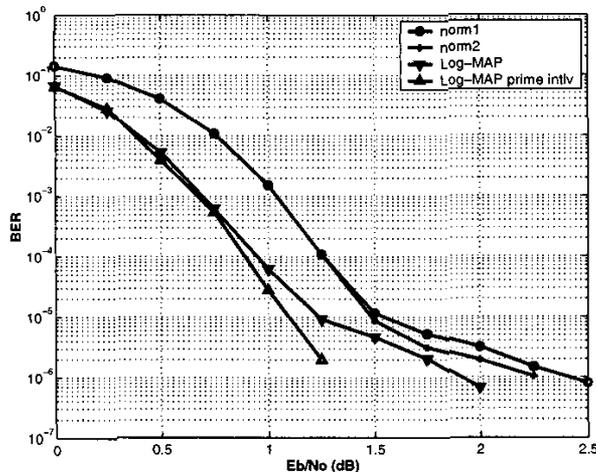


Fig. 3. Turbo coding performance with two types of SOVA normalization and reference performance comparison in the AWGN channel after 6 decoding iterations and 1000 bits frame

For comparison, the turbo coding performance with pseudo-random interleaver when using the Log-MAP algorithm is also plotted. It can be seen that the *norm2* method is 0.25 dB worse than the Log-MAP algorithm with pseudo-random interleaver at BER of around 10^{-6} and 0.75 dB worse than the Log-MAP algorithm with prime interleaver at the same BER, which is acceptable given the complexity savings.

B. Uncorrelated Rayleigh Fading Channel

The uncorrelated (independent) Rayleigh fading channel is considered in this paragraph. This is the case of the propagation channel between a mobile terminal and a base station. The frame length is equal to 1000 bits and the rest of the parameters are shown in Table I. Also, no channel state information is available at the decoder in order to compare with [15], [16].

The 8-state turbo coding performance with the normalized SOVA (either the *norm1* or the *norm2*) as well as the classical (37,21)_o 16-state turbo encoder with the standard SOVA and the MAP algorithm are plotted in the same figure (Fig. 4).

The coding gain improvement using the *norm2* instead of the *norm1* method is 0.25 dB at BER of around 10^{-6} , exactly the same as in the AWGN channel for the same frame length. That means this improvement is independent of the channel characteristics when the same weighting factors are used.

Either the *norm1* or the *norm2* method performs 0.25 dB better than the classical (37,21)_o 16-state turbo encoder with the standard SOVA at BER of around 10^{-5} . If the MAP algorithm is considered to decode the same 16-state turbo code, the normalized SOVA is 0.25 dB worse at BER of around 10^{-6} .

In order to keep the same performance improvement of the normalized SOVA at lower BER, we consider the case of another 16-state turbo encoder with generator polynomials (31,33)_o and the rest of the parameters from Table I. As can be

seen from Fig. 3, the coding gain improvement using the *norm2* instead of the *norm1* method is still 0.25 dB at BER of around 10^{-6} . In addition, this 16-state turbo encoder is 0.25 dB better than the 8-state turbo encoder at BER of around 10^{-6} when using either the *norm1* or the *norm2* methods respectively. Also, it approaches to 0.25 dB of the MAP algorithm of the classical (37,21)_o 16-state turbo encoder at BER of around 10^{-6} and outperforms the SOVA algorithm of the same classical turbo encoder by 0.25 dB at BER of around 10^{-6} .

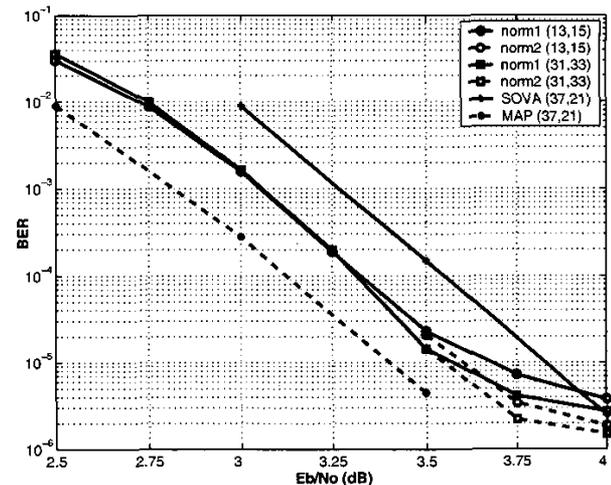


Fig. 4. Turbo coding performance with two types of SOVA normalization and reference performance comparison in an uncorrelated Rayleigh fading channel after 8 decoding iterations and 1000 bits frame.

C. Uncorrelated Rician Fading Channel

Here we extend the previous simulation results to include an uncorrelated (independent) Rician fading channel. This is the situation of a communication link between a mobile terminal and a satellite. It is known that when the Rice factor k is zero, this is equivalent to a Rayleigh fading channel, while when the Rice factor k is infinite, this is equivalent to an AWGN channel. The simulation parameters are the same as in Table I with a 1000 bits frame and no channel state information available at the decoder.

The 8-state turbo coding performance with the normalized SOVA (either the *norm1* or the *norm2*) for various values of the Rice factor k is plotted in Fig. 5. In the same figure is also shown the Log-MAP algorithm performance in the AWGN channel.

The coding gain improvement using the *norm2* instead of the *norm1* method is still 0.25 dB at BER of 10^{-6} , exactly the same as in the previous types of channel for the same frame length.

When the Rice factor k is approaching the infinite, the turbo code performance with the normalized SOVA (either the *norm1* or the *norm2*) is expected to require less than 1.5 dB at BER of 10^{-5} . This is taken account by observing the Fig. 3

where it is plotted the same turbo code performance but after 6 decoding iterations. That means, in case of $k \rightarrow \infty$, the normalized SOVA is about 0.25 dB worse than the Log-MAP decoding at BER of 10^{-5} . Also, the coding gain improvement with the normalized SOVA when $k \rightarrow \infty$ instead of $k = 10$ is about 0.5 dB at BER of 10^{-5} .

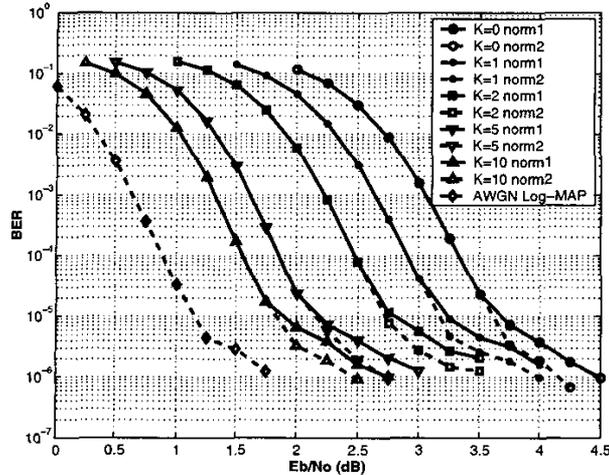


Fig. 5. Turbo coding performance with two types of SOVA normalization in an uncorrelated Rician fading channel and with Log-MAP in the AWGN channel after 8 decoding iterations and 1000 bits frame.

V. CONCLUSIONS

Two different normalization schemes were considered in order to correct the SOVA output that follows the HR updating rule. In the first normalization (*norm1*) we multiply the extrinsic information by a constant factor that does not depend on the variance of the decoder output. When we increase this factor in the last decoding iteration only, we obtain a new normalization scheme (*norm2*) so as to approach better the BR updating rule. The complexity that is added to the turbo decoder is relatively small, just one more multiplication of the extrinsic information with a constant factor, per component decoder and decoding iteration.

It was shown that the *norm2* method performs 0.25 dB better than the *norm1* method at BER of 10^{-6} in the AWGN channel as well as in an uncorrelated (independent) Rayleigh/Rician fading channel. This was done for specific turbo encoding parameters and correcting factors. When considering a smaller frame of bits, the coding gain improvement of the *norm2* method compared to the *norm1* method in the AWGN channel was less, about 0.125 dB at BER of 10^{-6} , but the *norm2* method of a turbo encoder that has half the number of states performs 0.125 dB better than the Bi-SOVA at BER of

10^{-5} and exactly the same as the MAP algorithm at BER lower than 10^{-5} .

ACKNOWLEDGMENT

The authors would like to thank Mr. Atta Quddus, Research Fellow at CCSR, for his valuable help and discussion.

REFERENCES

- [1] C. Berrou, C., A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes (1)", *Proc. IEEE ICC '93*, Geneva, Switzerland, May 1993, pp. 1064-1070.
- [2] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear code for minimizing symbol error rate", *IEEE Trans. Inform. Theory*, vol. IT-20, Mar. 1974, pp. 284-287.
- [3] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain", *Proc. IEEE ICC '95*, Seattle, June 1995, pp. 1009-1013.
- [4] J. Hagenauer, and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications", *Proc. IEEE GLOBECOM '89*, Dallas, Nov. 1989, pp. 1680-1686.
- [5] L. Papke, and P. Robertson, "Improved decoding with SOVA in parallel concatenated (Turbo-code) scheme", *Proc. IEEE ICC '96*, July 1996, pp. 102-106.
- [6] L. Lin, and R. Cheng, "Improvements in SOVA-based decoding for turbo codes", *Proc. IEEE ICC '97*, June 1997, pp. 1473-1478.
- [7] M. P. C. Fossorier, F. Burkert, S. Lin, and J. Hagenauer, "On the equivalence between SOVA and Max-Log-MAP decodings", *IEEE Comm. Letters*, vol. 2, no. 5, May 1998, pp. 137-139.
- [8] J. Vogt, and A. Finger, "Improving the max-log-MAP turbo decoder", *IEE Elect. Letters*, vol. 36, no. 23, Nov. 2000, pp. 1937-1939.
- [9] P. Robertson, "Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes", *Proc. Globecom '94*, Dec. 1994, pp. 1298-1303.
- [10] G. Colavolpe, G. Ferrari, and R. Raheli, "Extrinsic information in iterative decoding: a unified view", *IEEE Trans. Commun.*, vol. 49, no. 12, Dec. 2001, pp. 2088-2094.
- [11] S. Papaharalabos, P. Sweeney, B.G. Evans, "Modification of branch metric calculation to improve iterative SOVA decoding of turbo codes", *IEE Elect. Letters*, vol. 39, no.19, Sep. 2003, pp. 1391-1392.
- [12] W. Feng, and B. Vucetic, "A list bidirectional soft output decoder of turbo codes", *1st Int. Symp. on Turbo Codes*, Brest, France, 1997, pp. 288-292.
- [13] R. A. Stirling-Gallacher, "Performance of sub-optimal normalisation schemes for a turbo decoder using the soft output Viterbi algorithm", *Proc. IEEE Int. Symp. on Pers., Ind., and Mob. Radio Commun. (PIMRC) '00*, Sep. 2000, pp. 888-892.
- [14] A. Shitani, H. Suda, F. Adachi, "Complexity reduction of turbo decoding", *Proc. IEEE Veh. Tech. Conf. (VTC) '99 Fall*, Sep. 1999, pp. 1570-1574.
- [15] B. Vucetic, and J. Yuan, *Turbo codes, principles and applications*, Kluwer Academic Publishers, 2000.
- [16] J. Yuan, W. Feng, and B. Vucetic, "Performance of parallel and serial concatenated codes on fading channels", *IEEE Trans. Commun.*, vol. 50, no. 10, Oct. 2002, pp. 1600-1608.