Trellis-Based Decoding of High-Dimensional Block Turbo Codes

Sooyoung Kim, Woo Seok Yang, and Ho-Jin Lee

This paper introduces an efficient iterative decoding method for high-dimensional block turbo codes. To improve the decoding performance, we modified the soft decision Viterbi decoding algorithm, which is a trellis-based method. The iteration number can be significantly reduced in the soft output decoding process by applying multiple usage of extrinsic reliability information from all available axes and appropriately normalizing them. Our simulation results reveal that the proposed decoding process needs only about 30% of the iterations required to obtain the same performance with the conventional method at a bit error rate range of 10^{-5} to 10^{-6}.

I. INTRODUCTION

Iterative decoding of parallel concatenated convolutional codes, also known as turbo codes [1], provides a huge coding gain with a performance close to the Shannon limit. This powerful iterative decoding technique using soft-output information has also been applied to block codes, referred to as block turbo codes. Pyndiah et al. first introduced block turbo codes [2] using product codes [3]. Because of several advantages of block turbo codes, including flexibility of modification, commercial block turbo code chips are now available and can be applied to many areas, including satellite communications [4]-[7]. Our previous investigation proposed a block turbo code scheme to be used as an adaptive coding scheme to counteract rain attenuation in a satellite communication link [8].

Most of the soft output decoding algorithms for block turbo codes are based on algebraic decoding methods, which inherently do not have soft decision decoding capability and usually have to be resorted by the decoder several times to estimate soft outputs. For example, Pyndiah used the Chase algorithm [9] to compute soft outputs. Efficient Channel Coding, Inc [4] and Advanced Hardware Architecture [5] used their patent-pending algebraic-based decoding algorithm [10].

All linear block codes can be represented as trellis structures [11], and thus iterative decoding using the soft output Viterbi algorithm (SOVA) [12] or the maximum *a posteriori* (MAP) algorithm [13] is possible for block codes. Because the MAP algorithm is prohibitively complex, we used the SOVA in this paper. We previously investigated trellis decoding for block turbo codes to take advantage of the inherent soft-in soft-out decoding capability of a trellis decoder [14]. We used the less

---

Manuscript received Dec.12, 2001; revised Aug. 31, 2002.

This work has been supported by the Korea Ministry of Information and Communication.

Sooyoung Kim (phone: +82 42 860 4907, e-mail: sookim@etri.re.kr) and Woo Seok Yang (e-mail: wsyang@etri.re.kr) are with Broadband Wireless Transmission Team, ETRI, Daejeon, Korea.

Ho-Jin Lee (e-mail: hjlee@etri.re.kr) is with Satellite Multimedia Research Team, ETRI, Daejeon, Korea.

complex SOVA to estimate soft output and incorporated several performance improvement techniques [15], [16] into it to overcome its inferiority to the MAP algorithm.

In this paper, we consider high dimensional block turbo codes using a trellis-based iterative decoding method. High dimensional turbo codes can be implemented using *m*-dimensional (*m*>2, *m* is an integer) product codes. At the expense of bandwidth, they can provide a more powerful coding gain than typical two-dimensional product codes. Their applicability will be discussed in more detail in section II. We propose a highly efficient SOVA-based iterative decoding method for high dimensional product codes. In our algorithm, in addition to the enhanced error correction capability that is due to the added dimension, the decoding speed is accelerated when we simultaneously apply soft outputs from all available axes and use a suitable normalization method.

This kind of "multiple usage of extrinsic information" was investigated by Ramamurthy and Ryan [17]. They applied another differential encoder to a serial concatenation of a convolutional code with a differential encoder, which Hoeher and Lodge called turbo differential phase shift keying (DPSK) [18]; the result was a serial concatenation of a convolutional code with two differential encoders. In their scheme, called the double turbo DPSK, only the second decoder (in the middle) used the extrinsic information produced by the other two decoders, while in our scheme, every decoder uses the extrinsic information produced by the other available decoders.

In section II, we will briefly introduce the basic concept of *m*-dimensional block turbo codes. Section III presents the proposed iterative decoding algorithm, and section IV demonstrates the simulation results for an Additive White Gaussian Noise (AWGN) channel. Section V presents our conclusions.

II. HIGH DIMENSIONAL BLOCK TURBO CODES

1. Code Construction

The original concept of turbo codes comes from the iterative decoding for two parallel concatenated convolutional codes with an interleaver between them [1]. However, in a block turbo code, serial concatenation with a block interleaver is usually used. Although parallel concatenation is also possible for block turbo codes, serial concatenation is usually preferred, leading to the so-called product codes. Figure 1 shows the procedure for constructing classical 2-dimensional product codes using an (*n*1,*k*1) block code and an (*n*2,*k*2) block code. In the product code of Fig. 1, the check digits on check digits are the same whether the checks on rows or on columns are computed first. Figure 2 shows the configuration of a 3-dimensional product code (also from Efficient Channel Coding, Inc.); *m*-dimensional product codes are theoretically possible for *m* larger than 2.

Block codes modified so as to increase their minimum distance are often used as component codes; this results in a large increase in the minimum distance of the product code at the expense of a small decrease in its rate, hence of a small increase in the needed bandwidth [2]. Such modified codes include expurgated and extended codes [19].

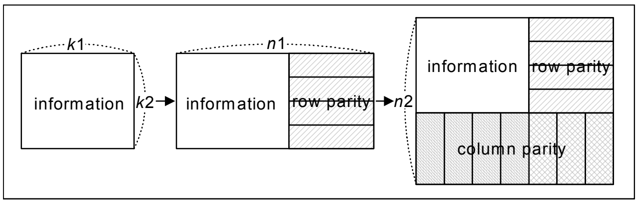


Fig. 1. Procedure for constructing 2-dimensional product codes.

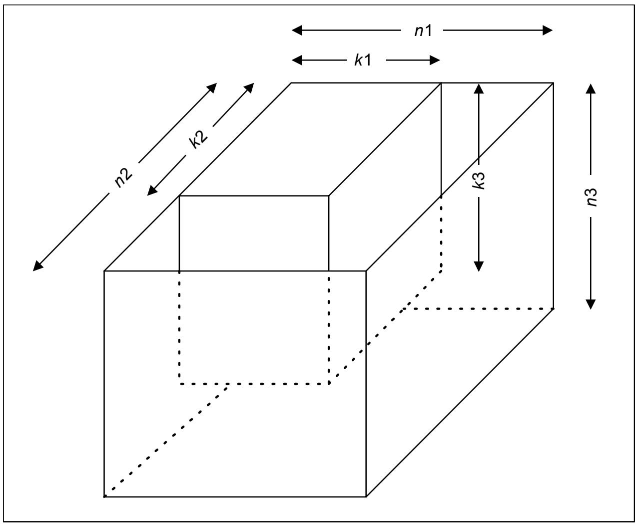


Fig. 2. Configuration of 3-dimensional product codes.

2. Applicability

Increasing the dimension of the product codes, which is an extension of making 2-dimensional product codes from linear block codes, results in an increase in the minimum distance of the product code. We can improve the decoding performance but at the expense of the code rate and the data frame length. Figure 3 demonstrates this: it shows the performances of the bit error rate (BER) of various codes whose component code is the (16,11) extended BCH code. We used soft decision Viterbi decoding for the 1-dimensional code and iterative decoding using the modified SOVA [14] for the 2-dimensional code and the 3-dimensional code. The performances shown in Fig. 3 are the BERs of the iterative decoding at the saturation points: at 6 iterations for the 2-dimensional code and at 30 iterations for the 3-dimensional code.

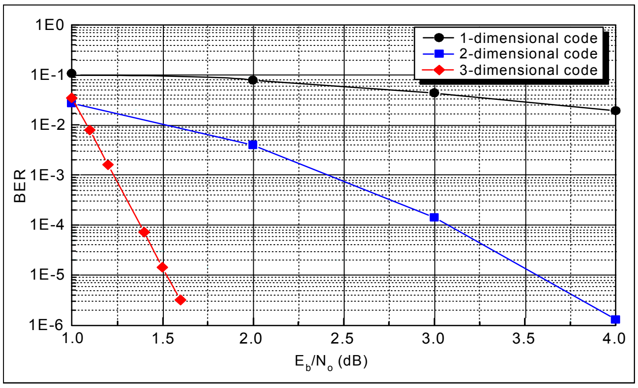


Fig. 3. BER performances of various codes whose component code is the (16,11) extended BCH code.

Sometimes, with a fixed code rate and data frame length, only a multi-dimensional scheme can satisfy the constraint since we have a larger degree of freedom in choosing the component codes in each axis.

High dimensional schemes often produce better performances than low dimensional schemes with a similar code rate or data frame length. Figure 4 demonstrates this: it shows the performance of the BER of a 3-dimensional code with a (16,11) extended BCH code compared to the performance of two 2-dimensional codes with a similar code rate and with the same data frame length. The 2-dimensional code with a (64,57) extended BCH code used the same data frame length as the 3-dimensional code, that is, 4096. The 2-dimensional code with a (31,20) expurgated BCH code used a code rate similar to the 3-dimensional code: a code rate of about 0.42 for the 2-dimensional code and a code rate of about 0.32 for the 3-dimensional code.

The same iterative decoding algorithm was used as for the case in Fig. 3, and the performances shown in Fig. 4 are the BERs of the iterative decoding at the saturation points: at 10 iterations for the 2-dimensional code and at 30 iterations for the 3-dimensional code. We note that the 3-dimensional code needs many more iterations to get a saturated performance. This is because the component code of a 3-dimensional code is simpler and a good performance results from the added dimension.

We need to use much more complex component codes for a 2-dimensional code in order to produce both the same code rate and the same frame length as those of a 3-dimensional code. In this example, the 2-dimensional scheme with component codes of the (64, 39) code and the (64, 33) code would approximately satisfy this, but of course the decoding of the components would be much more complex. Application of the SOVA to these codes is almost impossible because of the complexity of the decoder, unless a very efficient complexity reduction algorithm is used.

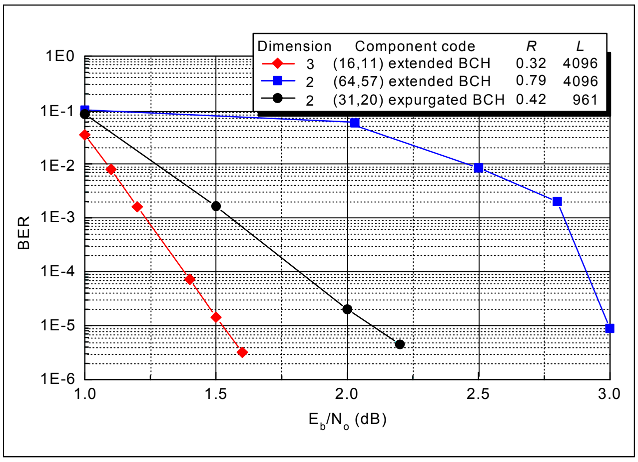


Fig. 4. BER performances of 2 and 3-dimensional block turbo codes (*R*: code rate, *L*: data frame length).

If we assume that a high dimensional block turbo code can produce a performance similar to that of a 2-dimensional code with a given code rate or data frame length, we can employ a less complex component code in the high dimensional block turbo code at the expense of the number of iterations. This can be an important advantage for high dimensional schemes because in block turbo codes parallel decoding of codeword sequences is possible in each axis. Therefore, the overall decoding speed of the high dimensional code can be faster than that of the 2-dimensional code.

3. Decoding Procedure

We decoded product codes iteratively on each axis using soft output information from the decoding on the other axes. For the 2-dimensional block turbo codes, we performed the decoding of the *x* axis codewords with the extrinsic information estimated from the soft output information on the *y* axis, and vice versa.

For the 3-dimensional block turbo codes, let us assume that decoding is performed in the order of *z* axis, *y* axis, and then *x* axis. The conventional decoding method consists of using the soft-output information from the last decoded axis for decoding the next one, i.e., using the soft-output information from the *x* axis for decoding on the *z* axis, that from the *z* axis for decoding on the *y* axis, etc.

For decoding a high-dimensional product code on an axis, we can use at the same time the soft output information available from all the other axes, instead of only that of the previously decoded axis. For instance, in the 3-dimensional case, decoding on the z axis can be performed using the soft information from both the x axis and the y axis instead of only that from the x axis. Doing so, we improve the performance of each iteration step, hence diminishing the total number of needed steps. We refer to this decoding method as *multi-incorporation of soft output* (MISO). In the next section, we will describe the detailed iterative decoding procedure for MISO with the SOVA.

III. MISO WITH THE SOVA

1. Decoding Algorithm

An iterative decoding procedure using the SOVA for block turbo codes is described in detail in [14]. Soft output from the SOVA decoder was normalized using the normalization factor introduced in [15], and a modified up-dating rule [16] was used to compensate for the SOVA's inferior performance compared to a MAP decoder. The iterative decoding performance of the SOVA decoder is usually very sensitive to the normalization procedure. Hence in this paper, we propose an appropriate normalization method for MISO, which will be explained in the next subsection.

Figure 5 shows the inputs and outputs of the SOVA component decoder used in block turbo codes. The SOVA component decoder processes the inputs of the intrinsic information $L(u)$ from the channel and the extrinsic information $Le(u)$ for the information sequence u. $L(u)$ and $Le(u)$ are log-likelihood ratios as defined in [15]. The decoder estimates the hard decision information sequence u' and its reliability (or soft output) $L(u')$ [12]. $Le(u)$ is estimated from $L(u')$ as normally done in an iterative decoder as follows

$$Le(u) = L(u') - Le(u') - L(u), \tag{1}$$

where $Le(u')$ is the extrinsic information used for decoding u'.

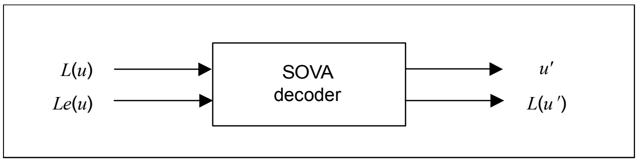


Fig. 5. SOVA component decoder used in block turbo codes.

Using the SOVA decoder of Fig. 5, we will review the iterative decoding procedures using MISO with the SOVA for m-dimensional block turbo codes (Fig. 6). First the decoder initializes extrinsic information for the m-axes. We note that the decoder has m buffers of n^m elements each, because there are n^m bits in an m-dimensional code and each axis needs its own buffer for extrinsic information. This will be explained in more detail later. The decoder then estimates the extrinsic information which will be used for decoding on the i axis by collecting all the available extrinsic information estimated on other axes, as follows.

$$Le(u_i) = \sum_{j=1, j \neq i}^{m} Le'(u'_j), \tag{2}$$

where u_i is the information sequence on the i axis, u'_i is the decoded information sequence on the i axis, $Le(u_i)$ is the extrinsic information which will be used for decoding on the i axis, and $Le'(u'_i)$ is the extrinsic information estimated from decoding on the i axis to be used for decoding on the other axes as a part of the total extrinsic information.

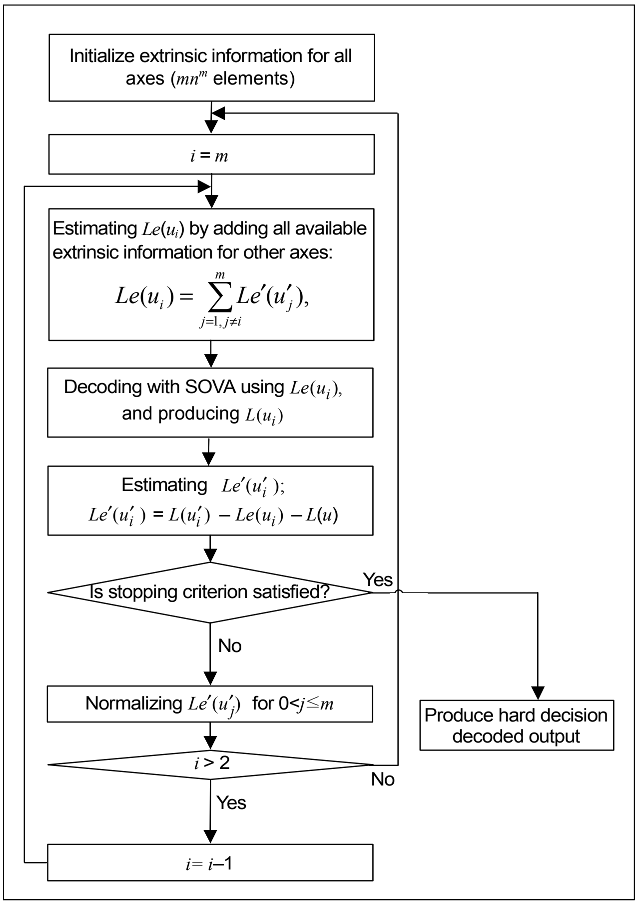


Fig. 6. Flow chart for the decoding procedure of MISO with the SOVA.

The decoder next performs the SOVA on the i axis using the extrinsic information $Le(u_i)$ and produces soft outputs

$L(u_i')$ as a result of the decoding. $Le'(u_i')$ is then estimated as follows.

$$Le'(u_i') = L(u_i') - Le(u_i) - L(u), \tag{3}$$

where $L(u)$ is the channel reliability of the systematic symbols.

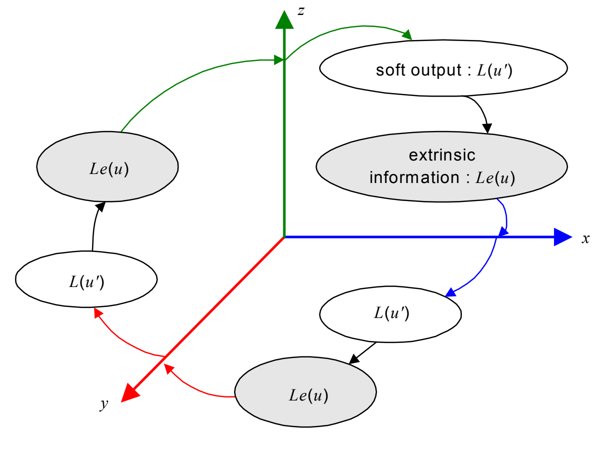
If we assume a 3-dimensional block turbo code, the extrinsic information and the soft outputs on the x, y, and z axes in (2) and (3) can be written as follows.

$$\begin{aligned}
Le(u_z) &= Le'(u_x') + Le'(u_y') \\
Le'(u_z') &= L(u_z') - Le(u_z) - L(u) \\
Le(u_y) &= Le'(u_x') + Le'(u_z') \\
Le'(u_y) &= L(u_y') - Le(u_y) - L(u) \\
Le(u_x) &= Le'(u_y') + Le'(u_z') \\
Le'(u_x) &= L(u_x') - Le(u_x) - L(u).
\end{aligned} \tag{4}$$

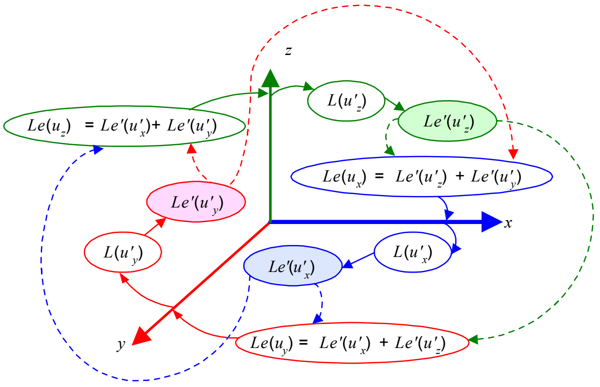
Figure 7 shows the relationship between the soft outputs and the extrinsic information in the conventional and MISO schemes for 3-dimensional block turbo codes. In the conventional scheme, the extrinsic information is updated in turn without any distinction of individual axes, while in the MISO scheme, each axis produces its own soft output and estimates its own extrinsic information.

In order to do it in this way, the decoder needs to store extrinsic information for each axis (Fig. 7). Therefore, it does require more memory than the conventional decoder, but the complexity increment is negligible except that all the available extrinsic information has to be added before the decoding on each axis is performed. However, as far as the normalization procedure is concerned, the decoder needs to make an effort m times that of the conventional decoder in order to get an enhanced performance. In other words, m normalization procedures are required after decoding on an axis (Fig. 6). We will discuss the normalization problem in more detail in the next subsection.

After decoding on the i axis, the decoder then tests if a stopping criterion is satisfied. In this paper, we used a simple hard-decision-aided stopping criterion [20]. If it is satisfied, the decoder produces hard-decision decoded output and stops the iterative routine. If the stopping criterion is not satisfied, the decoder normalizes the extrinsic information for all existing axes. The above procedures are repeated for the next axis.



(a) conventional



(b) MISO

Fig. 7. Relationship between the soft outputs and the extrinsic information for the 3-dimensional block turbo codes.

2. Normalization of the Soft Output

The normalization technique is one of the most important factors in determining the decoding performance of the SOVA for block turbo codes. The normalization factor c is defined by [15],

$$c = \frac{2m_v}{\sigma_v^2}, \tag{5}$$

where m_v and σ_v are the mean and standard deviations of the magnitudes of the reliability values, respectively.

Let us first consider the normalization procedure for the conventional SOVA-based block turbo codes in Fig. 8(a). The decoder simply normalizes the extrinsic information $Le(u)$ at each axis since there is only one extrinsic information set. In the MISO scheme, the decoder has to deal with m extrinsic information sets at each axis. The most general way is normalizing the extrinsic information for the corresponding axis, that is, normalizing $Le'(u_i')$ after SOVA decoding on the i axis. We refer to this normalization method as Norm I. However, Norm I for MISO will result in a too optimistic estimation of the reliability values. This is because, compared to the conventional method, the reliability is divided into m parts in MISO and thus the normalization effect will be much less than that in the conventional method.

To compensate for this loss, we tried Norm II, where the decoder normalizes $Le'(u_i')$ m times after SOVA decoding on the i axis. However, Norm II for MISO resulted in a performance much inferior to that of the conventional method as we will see in the simulation results in section IV. Normalization of all available extrinsic information, after SOVA decoding on the i axis, resulted in the best performance. We refer to this normalization scheme as Norm III. In other words, the Norm III scheme normalizes $Le'(u_j')$, for $0<j\le m$. Figure 8 shows the flowcharts of the above explained normalization schemes for MISO. We note that Norm II normalizes the same extrinsic information m-times, while the Norm III scheme normalizes m-different extrinsic information.

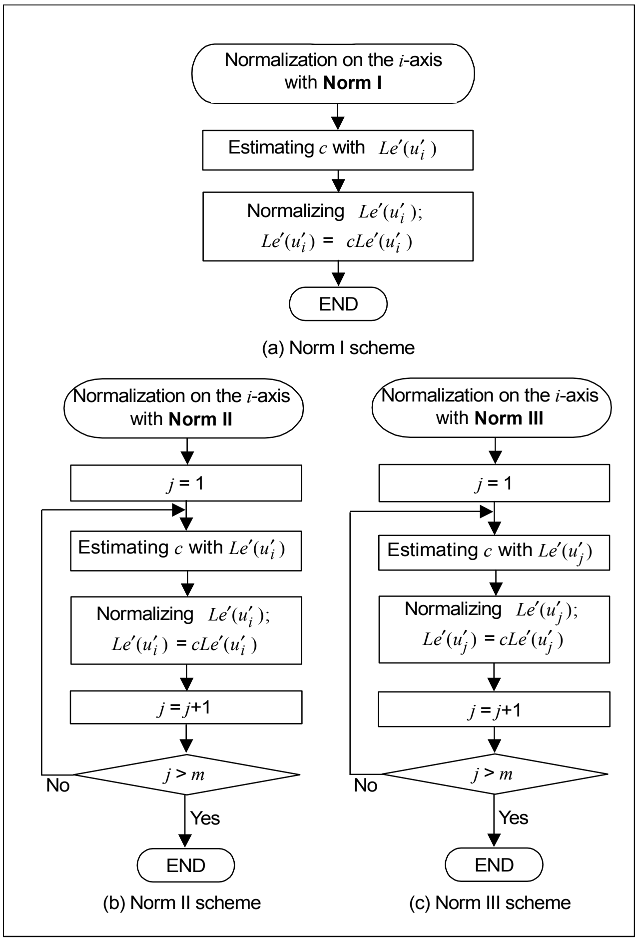


Fig. 8. Flow charts of the normalization schemes for MISO.

In addition, in order to make the performance rapidly converge, the maximum normalization factor was limited. Having investigated the variation of c with the iterations, we discovered that it converged to a value in the range of 0.2 to 0.5 (Fig. 9). Figure 9 shows the variation of c for the 2-dimensional block turbo code using the (31,26) BCH codes, and very similar curves were produced for the other codes. By limiting the maximum value of c, we were able to achieve an enhanced performance especially at an early stage of the iterations.

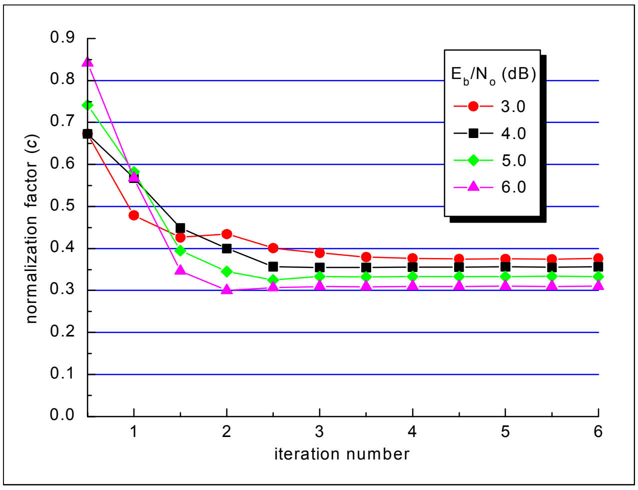


Fig. 9. Variation of c with iterations for the 2-dimensional block turbo codes with the (31,26) BCH code.

IV. SIMULATION RESULTS

Our simulations, performed on an AWGN channel, used a 3-dimensional block turbo code with a (16,11) extended BCH code. Figure 10 shows the performance of the block turbo code using the conventional decoding algorithm with the SOVA. The figure shows that the decoding performance can be enhanced by limiting the maximum value of the normalization factor c as discussed in the previous section. The results demonstrate that we can enhance the performance especially at an earlier stage of the iterations. This could also be investigated in the simulation results for 2-dimensional block turbo codes.

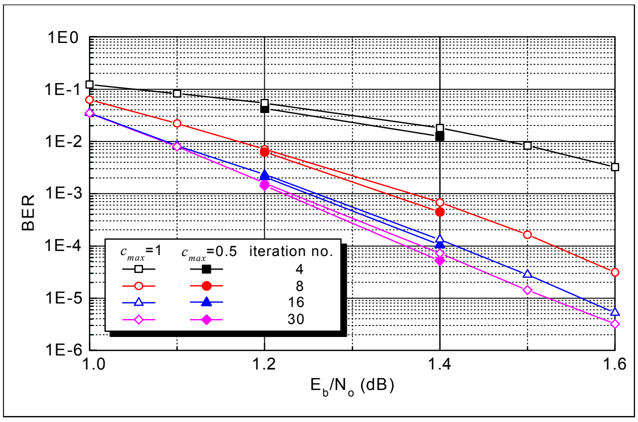


Fig. 10. BER performance of the 3-dimensional block turbo code using the (16,11) extended BCH code and the conventional SOVA.

Figure 11 shows the BER performance of the same block turbo code with the various normalization schemes discussed in section III. 2. Although MISO with a general normalization scheme (Norm I) produces enhanced performance compared to the conventional SOVA decoder, the amount of the enhancement is very small. Norm II deteriorates the decoding performance rather than enhancing it. By applying Norm III, MISO with SOVA achieves comparable performance enhancement, which can be clearly seen in Fig. 12. These results demonstrate that to improve iterative decoding performance in high dimensional block turbo codes, it is essential to not only incorporate multiple extrinsic information but also to adopt an appropriate normalization scheme.

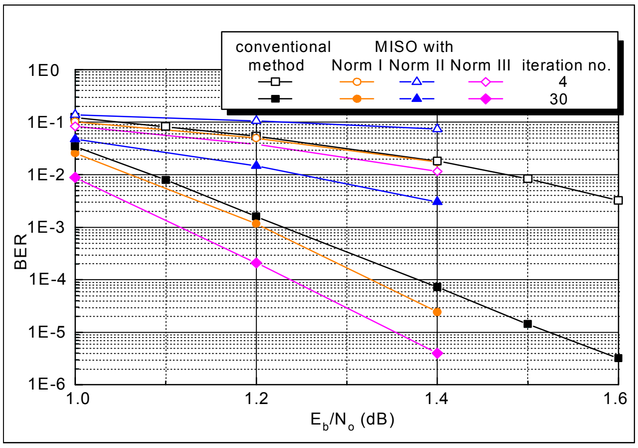


Fig. 11. BER performance of the 3-dimensional block turbo code using the (16,11) extended BCH code and MISO with various normalization schemes.

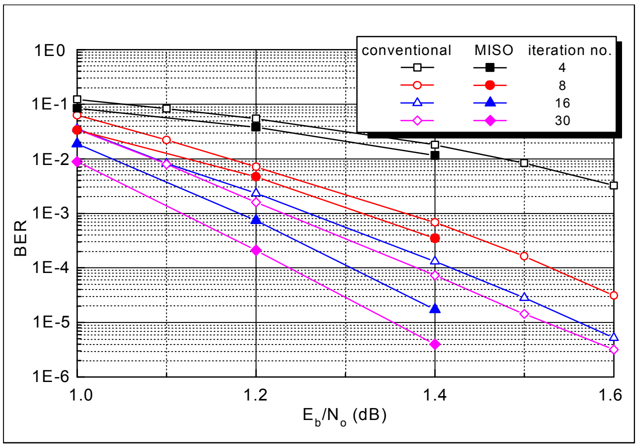


Fig. 12. BER performance comparison of the 3-dimensional block turbo code using the conventional SOVA and MISO with SOVA.

Figure 12 compares the performance of the conventional scheme and that of MISO with SOVA, where the decoder not only limits the maximum value of *c* but also uses Norm III. In contrast to the case of only limiting the maximum value of *c* (Fig. 10), the BER performance is more greatly enhanced in a later stage of iterations, resulting in about a 0.2 dB coding gain over the conventional method. As far as the number of iterations is concerned, the proposed decoder can achieve the same BER performance with just 1/3 of the iterations that the conventional decoder requires. In other words, the proposed decoder can produce a better performance with the same number of iterations, and it can also push back the performance saturation point.

V. CONCLUSIONS

We have proposed an efficient SOVA-based iterative decoding algorithm. The algorithm can be used for high dimensional block turbo codes, and it can reduce the number of iterations by about 60 percent with an acceptable complexity increase. Incorporating multiple soft outputs and appropriately normalizing them improves the performance. The proposed idea of multiple incorporation of soft outputs can be used for other soft output decoding methods, such as the MAP algorithm and other algebraic soft output decoding algorithms; hence this idea can play an important role in increasing decoding speed.

ACKNOWLEDGEMENT

The authors wish to thank all reviewers for their helpful comments to improve the quality of the paper.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon Limit Error-Correction Coding and Decoding: Turbo-Codes,” *Proc. of Int’l Conf. on Communications, ICC*, May 1993, Geneva, Switzerland, pp. 1064-1070.
[2] R. Pyndiah, A. Glavieux, A. Picart, and S. Jacq, “Near Optimum Decoding of Product Codes,” *Proc. of the IEEE Global Communications Conf., GLOBECOM*, Nov. 1994, San Francisco, USA, pp. 339-343.
[3] P. Elias, “Error-Free Coding,” *IRE Trans. Information Theory*, PGIT-4, Sept. 1954, pp. 29-37.
[4] http://www. eccincorp.com/.
[5] http://www.aha.com/.
[6] http://www.comtechefdata.com/.
[7] M. Vanderaar, R.T. Gedney, and E. Hewitt, “Comparative Performance of Turbo Product Codes and Reed Solomon / Convolutional Concatenated Codes for ATM Cell Transmission,” *Proc. of the fifth Ka Band Utilization Conf.*, Oct. 18-20, 1999, Taormina, Italy, pp. 409-416.

[8] Sooyoung Kim Shin, Kwangjae Lim, Kwonhue Choi, and Kunseok Kang "Rain Attenuation and Doppler Shift Compensation for Satellite Communications," *ETRI J.*, vol. 24, no. 1, Feb. 2002, pp. 31-42.

[9] D. Chase, "A Class of Algorithm for Decoding Block Codes with Channel Measurement Information," *IEEE Trans. on Information Theory*, vol. 18, no. 1, 1972, pp. 170-182.

[10] Efficient Channel Coding Inc., *Technical Description of Turbo Product Codes*, ver. 4.0, June 1999.

[11] J.K. Wolf, "Efficient Maximum Likelihood Decoding of Linear Block Codes Using a Trellis," *IEEE Trans. on Information Theory*, vol. 24, no. 1, 1978, pp. 76-80.

[12] J. Hagenauer and P. Höher, "A Viterbi Algorithm with Soft Decision Outputs and its Application," *Proc. of the IEEE Global Communications Conf., GLOBECOM*, Nov. 1989, pp. 47.1.1-47.1.7.

[13] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Trans. on Information Theory*, vol. IT-20, Mar. 1974, pp. 284-287.

[14] S.K. Shin, S.I. Lee, and S.P. Lee, "Evaluation of Block Turbo Code Performance with the Reduced Search Trellis Decoding Method," *IEE Proc. Communications*, vol. 148, no. 3, June 2001, pp. 125-131.

[15] L. Papke and P. Robertson, "Improved Decoding with the SOVA in a Parallel Concatenated (turbo-code) Scheme," *Proc. of Int'l Conf. on Communications, ICC*, June 1996, Dallas, USA, pp. 102-106.

[16] L. Lin and R.S. Cheng, "Improvements in SOVA-Based Decoding for Turbo Codes," *Proc. of Int'l Conf. on Communications, ICC*, June 1997, Montreal, Canada, pp. 1473-1478.

[17] R. Ramamurthy and W.E. Ryan, "Convolutional Double Accumulate Codes (or Double turbo DPSK)," *IEEE Communications Lett.*, vol. 5, no. 4, Apr. 2001, pp. 157-159.

[18] P. Hoeher and J. Lodge, "'Turbo DPSK': Iterative Differential PSK Demodulation and Channel Decoding," *IEEE Trans. on Communications*, vol. 47, no. 6, June 1999, pp. 837-843.

[19] G.C. Clark, Jr. and J.B. Cain, *Error-Correction Coding for Digital Communications*, Plenum Press, New York, 1981.

[20] Rose Y. Shao, Marc Fossorier, and Shu Lin, "Two Simple Stopping Criteria for Iterative Decoding," *Int'l Symposium on Information Theory*, Cambridge, Aug. 1998, p. 279.

Sooyoung Kim received the BS degree in electrical and electronics engineering from Korea Advanced Institute of Science and Technology, Korea, in 1990. After having worked at the Satellite Communication Technology Division of ETRI, Korea from February 1990 to September 1991, she received MSc and PhD degrees in electrical and electronics engineering from University of Surrey, UK in 1992 and 1995. From November 1994 to June 1996 she was employed as a Research Fellow at the Centre for Satellite Engineering Research, University of Surrey, UK. In 1996 she re-joined the Satellite Communication Technology Division, ETRI, Korea. She is currently a Team Leader of the Broadband Wireless Transmission Team responsible for developing efficient adaptive multi-carrier transmission schemes to compensate channel impairments in satellite communication systems. Her work also includes developing highly efficient coding techniques for digital communication systems.

Woo Seok Yang received the BS degree in electronics engineering from Yonsei University, Korea, in 1998 and the MS degree in electrical and electronics engineering from Korea Advanced Institute of Science and Technology, Korea, in 2000. In 2000, he joined ETRI, Korea as a Member of Research Staff. His research interests include channel coding, channel estimation, and satellite communication system.

Ho-Jin Lee received his BS, MS, PhD degrees in electronics engineering from Seoul National University (SNU), Korea, in 1981, 1983 and 1990. He joined ETRI in 1983 and has been involved with TDX, a full electronic digital switching system development project, satellite ground mission/network control system development, KOMPSAT ground control system development, and satellite communication earth stations/service development. He was with TRW, USA, as a Visiting Engineer for 2 years. He worked as the Director of the Satellite Communications Application Department of ETRI and is now the Project Manager of the DVB-RCS system, two-way satellite broadband access system development project.