Piecewise Phase Recovery Algorithm Using Block Turbo Codes for Next Generation Mobile Communications

Sunheui Ryoo, Sooyoung Kim, and Do Seob Ahn

This paper presents an efficient carrier recovery algorithm combined with a turbo-coding technique in a mobile communication system. By using a block turbo code made up of independently decodable block codes, we can efficiently recover the fast time-varying carrier phase as well as correct channel errors. Our simulation results reveal that the proposed scheme can accommodate mobiles with high speed, and at the same time can reduce the number of iterations to lock the phase.

Keywords: Block turbo codes, iterative decoding, phase recovery.

I. Introduction

Lots of channel impairments in wireless communication systems call for powerful error correction coding schemes such as turbo codes [1]. An impressive performance of turbo codes can be achieved only if an ideal coherent receiver is assumed. Unfortunately, this is not the case for most wireless mobile communication systems with time varying channel characteristics. In this case, we need a special effort to lock the phase in coherent detection.

The beyond 3G (B3G) system has been developed to fully integrate various networks including satellite systems [2] by targeting mobiles with a high speed of up to 300 km/h. For example, the 3rd Generation Partnership Project standards are now targeted to support a maximum user speed of 200 km/h [3], while the IEEE 802.16e standard is targeted to support a user speed of 60 km/h and will be extended to a maximum 250 km/h [4]. These high-speed mobile terminals also will suffer from the fast time-varying phase, and this will penalize phase recovery.

From this perspective, many studies on developing accurate carrier recovery techniques have been presented [5]-[10]. They show that carrier estimation algorithms can achieve a reasonably short acquisition time without a complex phase locked loop. In addition, it has been shown that such algorithms can reduce power consumption since they don't have to employ preamble or pilot signals.

As far as carrier phase estimation is concerned, C. Morlet and others [5] gave a good example of a low-complexity algorithm. They used the polar coordinates to reduce the complexity of the phase estimation. In [6], the carrier phase estimator was designed

Manuscript received Sept. 26, 2005; revised Apr. 04, 2006.

Sunheui Ryoo (email: shryoo@etri.re.kr, phone: + 82 42 860 5116) and Do Seob Ahn (ahnds@etri.re.kr) are with Radio & Broadcasting Research Division, ETRI, Daejeon, Korea.

Sooyoung Kim (email: sookim@chonbuk.ac.kr) is with Division of Electronics and Information Engineering, Chonbuk National University, Jeonju, Jeonbuk, Korea.

jointly with an open-loop maximum-likelihood carrier frequency estimator. In these methods, the phase estimator is operated independently of the channel decoders.

In [7]-[10], the estimator recovers the carrier phase iteratively, combined with various error correction coding schemes. The technique illustrated in [7] for convolutional encoding was based on tentative symbol decisions calculated by recombining the output bits and the memory state of the Viterbi decoder for a given decoding depth. A similar approach, specifically geared for turbo coding, is presented in [8], where the tentative decisions of the first soft-input/soft-output (SISO) decoder (based on a soft-output Viterbi algorithm) are used in the phase recovery system. Both algorithms in [8] and [9] make use of tentative decisions during the decoding process, but they still do not use any extrinsic information from the decoder.

Unlike the approaches mentioned above, the estimation method proposed in [9] uses soft decisions in the form of a posterior probability (APP) at each iteration of the expectation-maximization (EM) algorithm, which is not very practical since their computational complexity is fairly high.

In [10], the authors proposed a low-complexity blind carrier recovery technique suited for turbo codes. Their algorithm estimated the phase value iteratively by using the information from the decoder. In the iterative process, a constant phase was assumed in a codeword. However, this may cause a problem if the Doppler spread of the channel, that is, the coherence time, is shorter than a codeword. In this situation, this algorithm will produce a constant phase value averaged over a codeword even though the actual phase error varies. Motivated by this, we propose a new approach to estimate the fast time-varying phase.

Many recent mobile applications require error correcting codes with a very long codeword to provide high quality multimedia services [11], which makes it difficult for a phase error to be kept constant over such a long codeword. In addition, even after the phase compensation, received samples with residual phase rotation would reduce the instantaneous signal-to-noise ratio, and this again decreases the decoding performance.

In order to solve this problem, we propose a piecewise carrier recovery algorithm using block turbo codes that were first introduced by Pyndiah and others [12]. In our approach, we use a multi-dimensional block turbo code consisting of independently decodable component codes, and additional extrinsic information in the iterative decoding process [13]. Here we mean the component code consisting of a one-dimensional (n, k) block code. We estimate the phase reference at each component code, instead of estimating at the whole product code. In this way, we can correct the time-varying phase within a period of a component code length.

In addition, the proposed piecewise phase recovery algorithm will be applied in the receiver side to enhance the performance, and it can be applied to existing mobile systems or standards without any modification to the current specification. Therefore, this performance enhancement of the receiver may lead to an increase in system capacity.

In section II, we describe the principle of the conventional phase recovery algorithm with the iterative decoder, and then introduce the proposed algorithm. In section III, we demonstrate the performance of the proposed algorithm compared to the conventional scheme. Finally, we draw some conclusions in section IV.

II. Iterative Phase Recovery Algorithm

1. Conventional Approach

In [10], the authors proposed a carrier phase estimation algorithm with a reasonably short acquisition time and no preamble. The algorithm used a pseudo-maximum-likelihood estimation approach, as well as extrinsic information provided by the SISO decoders iteratively. In doing so, iterative decoding and carrier phase recovery work jointly.

In this process, the m-th received signal $r_0[m]$ is updated with decoding iteration as

$$r_i[m] = r_{i-1}[m] \cdot e^{-j\tilde{\theta}_i}, \tag{1}$$

where $r_i[m]$ is the phase-compensated signal with the estimated carrier phase at the i-th decoding iteration.

The carrier phase in a code block is calculated by using the m-th received signals $r_{i-1}[m]$ and the soft-detected symbol $\alpha_i[m]$ from the iterative decoder as

$$\tilde{\theta}_i = \arg\left\{\sum_{m=1}^{N} r_{i-1}[m] \cdot \alpha_i^*[m]\right\}, \tag{2}$$

where m is the bit index in a codeword and N is the codeword length.

The authors in [10] applied the above phase estimation algorithm to high-order quadrature amplitude modulation receivers, and they define the soft-detected symbol $\alpha_i[m]$ for various modulation schemes. For example, for quadrature phase shift keying (QPSK), $\alpha_i[m]$ is derived as

$$\alpha_i[m] = \tanh\frac{L_i\left(a^{[m]}\right)}{2} + j\tanh\frac{L_i\left(b^{[m]}\right)}{2}, \tag{3}$$

where $a^{[m]}$ and $b^{[m]}$ represent the two-bit Gray mapped signal onto the m-th QPSK symbol and $L_\mathrm{i}(a)$ denotes the extrinsic information for the symbol a at the i-th decoding iteration.

Therefore, this phase estimation algorithm will produce a fixed estimated phase value over a codeword because a constant phase value was assumed initially. In order to deal with the fast time-varying phase error for the case when the phase error value changes in a codeword, we need a fine phase-correction mechanism. Our previous study results demonstrated this [14], and we will extensively show the various results in section III.

2. Piecewise Iterative Phase Recovery

Figure 1 shows a block diagram of the proposed carrier phase recovery algorithm. The carrier phase recovery is applied to a signal packet at the receiver, wherein the signals have been subjected to block turbo encoding at the transmitter. The phase estimation and recovery jointly operate with the turbo decoder as in the conventional algorithm.

In our algorithm, however, we estimate the phase error in a very short time period by using the characteristics of the block turbo codes. Due to the independency of the component codes in the block turbo codes, the component decoder estimates the extrinsic information in parallel and passes them to the corresponding sub-block phase estimators as shown in Fig. 1.

The phase estimator calculates the phase using the soft-output of the turbo decoder, and we can then correct the received signal for each sub-block with the estimated phase error. Therefore, (1) for the conventional algorithm will be changed for the proposed algorithm into

$$r_{il}[m] = r_{(i-1)l}[m] \cdot e^{-j\tilde{\theta}_{il}}, \tag{4}$$

where $r_{il}[m]$ is the phase compensated signal and $\tilde{\theta}_{il}$ is the estimated carrier phase in the l-th component code at the i-th decoding iteration. The phase estimator consists of several phase estimators for each component code, as shown in Fig. 1. It first estimates the phases of the corresponding component codes and corrects them accordingly. Therefore, the estimation and correction process is done in a component code basis instead of in a whole codeword. This is repeated at each decoding iteration.

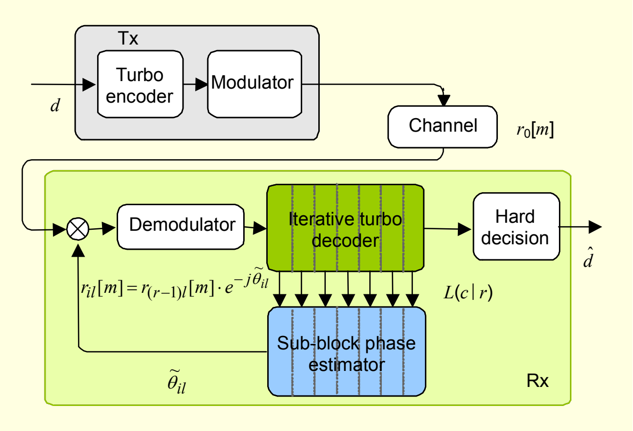

Fig. 1. System model of piecewise phase recovery.

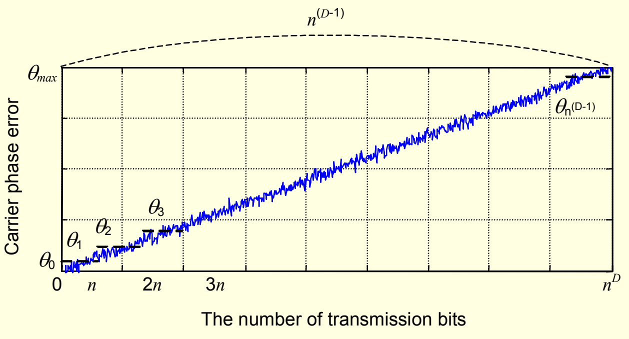

Fig. 2. Example of phase error in a codeword.

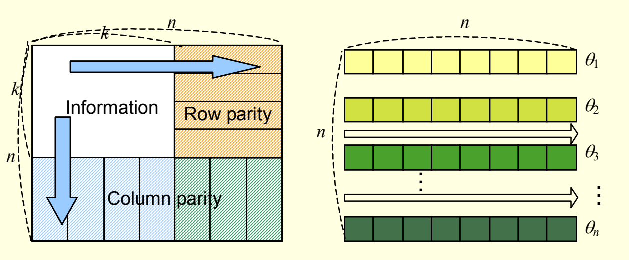

Fig. 3. Piecewise phase estimation in a 2D block turbo code.

Figure 2 depicts an example of a channel response within a length of a codeword. We divide it with a component code length, and assume a constant value within the component code length. This is because we can decode each component code independently in the block turbo codes.

Figure 3 illustrates this by showing the two-dimensional (2D) block turbo codes and the phase corresponding to each component code block in the row direction ($\tilde{\theta}_l$ represents the phase of the l-th component code). Because we can form a D-dimensional block turbo code consisting of (n, k) component block code [13], we can divide the time-varying phase with $n^{(D-1)}$ time slots in the codeword. The carrier phase can be estimated for each component code block and updated during the iterative decoding by changing the decoding results. The carrier phases in a component code are also calculated by using the received signals and the soft-detected symbols from the iterative decoder:

$$\tilde{\theta}_{il} = \arg\left\{\sum_{m=1}^{n} r_{(i-1)l}[m] \cdot \alpha_{il}^{*}[m]\right\}, \tag{5}$$

where $r_{(i-1)l}[m]$ and $\alpha_{il}[m]$ are the received signal and soft detected symbol of the m-th bit of the l-th component code at the i-th iteration.

Nevertheless, this piecewise block phase recovery algorithm hardly requires additional complexity compared to a conventional one. The main computational load in the phase recovery is estimating the phase using (2), and compensating the phase using (1). In this sense, the computational loads of the conventional and proposed algorithms are the same. The only additional requirement of the proposed algorithm is that it requires additional memory to store the estimated phase value at each sub-block instead of a single value in a codeword.

By estimating the phase error for each sub-block, the proposed piecewise phase recovery algorithm with block turbo codes can compensate for a fast time-varying phase error accurately. Figure 4 demonstrates this, where we can see an example of the estimated phase value from the conventional and proposed phase recovery algorithms. In Fig. 4, we assume the channel phase error value changes continuously during a codeword length of 4096 bits, and its mean value is zero. In this situation, the conventional algorithm will integrate the phase error over a codeword to zero, and thus it estimates the phase error as zero as shown in Fig. 4. On the other hand, the proposed algorithm estimates the phase error in a component code basis, and thus it produces an approximating phase error to the actual value. In our example, we employed 3-dimensional (3D) block turbo codes with component codes of the (16, 11) extended BCH code, and thus the minimum phase error estimation length is 16 bits. This means that we can improve the time resolution of phase error estimation by 256 times (=4096/16). In the next section, we will describe the details of the proposed algorithm.

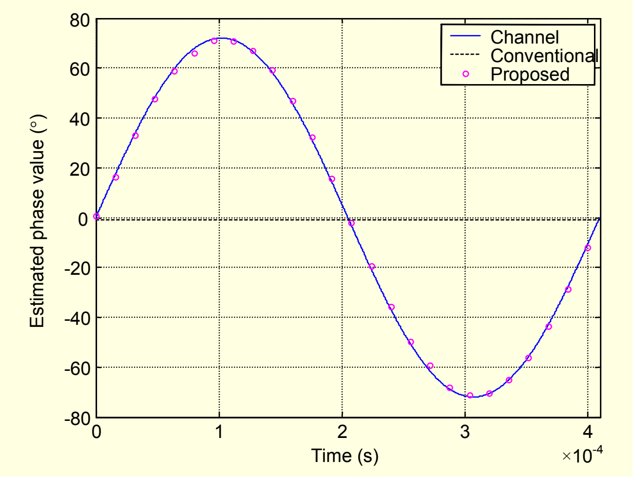

Fig. 4. Estimated phase error from the phase recovery algorithms.

III. Simulation Results

We evaluated the performance of the proposed piecewise phase recovery algorithm in a mobile communication channel with frequency bands of 2 GHz. We used a 3D block turbo code consisting of the (16, 11) extended BCH codes, where the codeword length is 16×16×16 = 4096 bits. We compared the performance to the conventional phase recovery in [10] applied to the same code.

As discussed in [10], the proposed phase recovery algorithm is applicable to high-order modulations. However, we used QPSK modulation with a data rate of 5 Mbps in the simulation. This is because the focus of this paper is investigating the performance of the piecewise phase recovery algorithm under a high-speed mobility condition. In the 3D block turbo code, the SISO decoder for each component code can process its job independently, and thus the decoding process goes in parallel at each axis [13].

We first investigate the trade-off between the estimation accuracy and tracking capability of the proposed algorithm, and find the optimum observation length. Next, we compare the performance of the proposed algorithm to the conventional algorithm with various mobile speeds and conditions. Finally, we investigate the performance in terms of bit error rate (BER).

1. Optimum Observation Length and Period

In a classical error correction mechanism, generally the estimation accuracy is increased at the expense of tracking capability (or convergence time), and vice versa. In this sense, the conventional algorithm extracts the phase error value with a single step, and thus the accuracy will be low. In our algorithm, we extract the phase error value at each component code, and thus we can increase the estimation accuracy even in a situation with a fast time-varying phase.

If the observation length (or sub-block length) to estimate a phase error is too short, we may not estimate the accurate phase

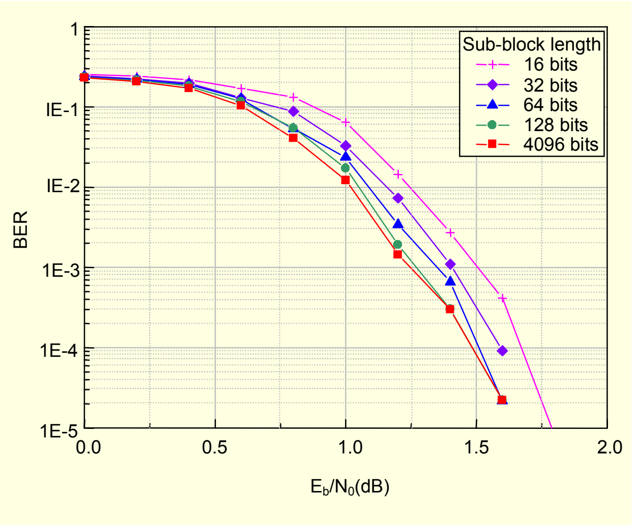

Fig. 5. BER performance according to the sub-block length.

value. From the simulation, we found the optimum sub-block length of the target code. Figure 5 shows the BER performance of the block turbo code with the proposed phase recovery algorithm according to sub-block length. As shown in the figure, a sub-block length of 128 bits, which corresponds to eight component codes, is sufficient to estimate the phase. Figure 5 also indicates that we may reduce the length of a sub-block in high signal-to-noise ratio ranges.

Another important trade-off analysis topic is the period of phase recovery process during iterative decoding. For easy explanation, let us consider a 2D block turbo code first. If the bits are transmitted row-by-row, then we can apply the phase recovery process when decoding the rows. For column decoding, however, we can apply the phase recovery process only after a de-interleaving process, and this of course induces additional computational time. In the 3D block turbo code, we can apply the phase recovery process without an additional de-interleaving process in *z*-axis decoding if the bits are transmitted in that direction.

Figure 6 shows the BER performance of the block turbo code with the proposed phase recovery algorithm according to the period when the mobile speed is 300 km/h. It shows that the performance enhancement due to additional phase recovery processes in the *y*- and *x*-axis is minor.

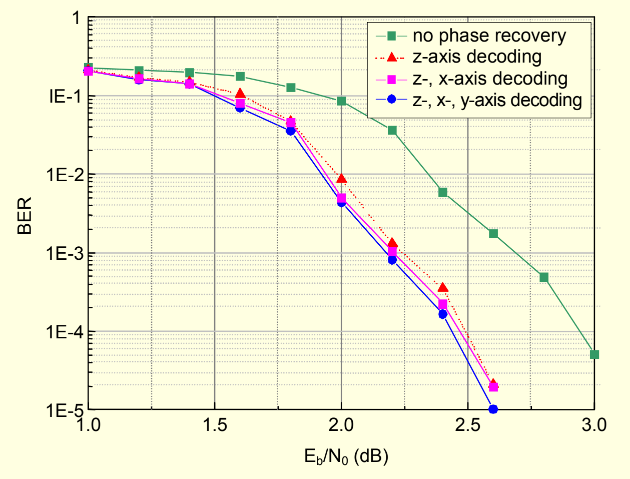

Fig. 6. BER performance according to phase recovery period.

2. Phase Error Correction Performance

The maximum phase error in a given time duration depends on the speed of mobile terminals. First, we assumed the velocity of the mobile is constant, and the direction to the base station is not changed (that is, continuously coming towards or going away) during the observation time of a codeword length. In this case, the phase value changes linearly, and its value continuously increases or decreases monotonously. Figure 7 illustrates a phase value change in a codeword in this case according to the mobile speed.

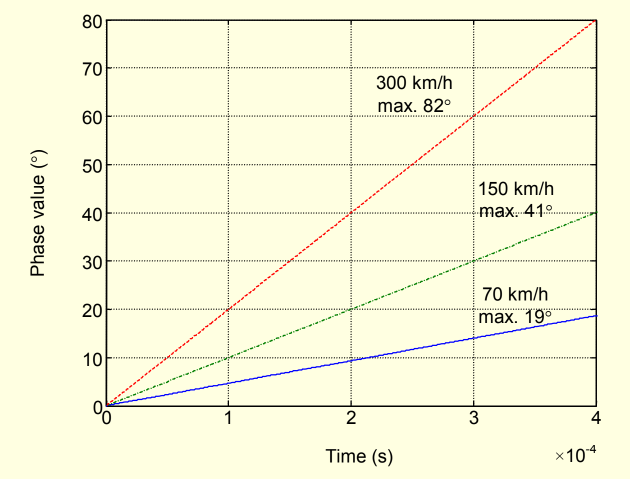

Fig. 7. Example of time varying phase by mobile speed.

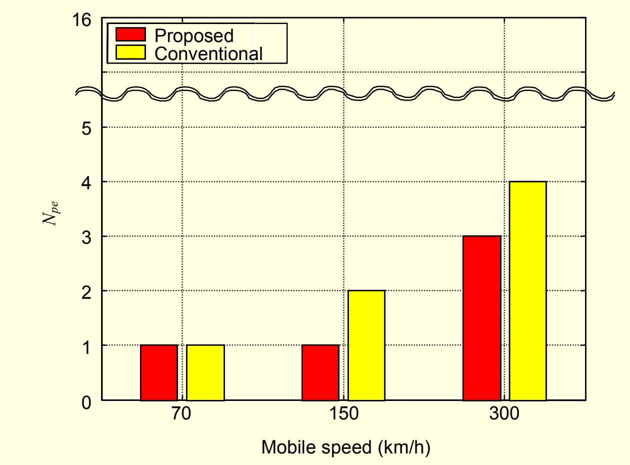

Fig. 8. Required number of iterations to lock the phase.

We need more iteration at high mobile speed because of a larger induced phase error. The conventional scheme estimates a phase value in a codeword with (2). On the other hand, the proposed scheme estimates a phase value with (5) in a component codeword that is much shorter than the codeword itself.

As investigated in the previous section, the optimum sub-block length is 128 symbols. Therefore, in our example, the proposed scheme will have 32 estimated phase values in a codeword. For comparison to the conventional scheme, in the performance of the proposed scheme of Fig. 8, we plotted the number of iterations to lock the phase error, N_{pe}. As shown in Fig. 8, the proposed scheme reduces the number of iterations to lock the phase error.

Figure 9 shows the performance of the phase estimator in terms of root mean squared error (RMSE) according to the mobile speed.

As we can see, the proposed scheme has less residual phase error than the conventional scheme for all mobile speeds. Although the proposed scheme can reduce the number of iterations to correct phase errors, we could not see great advantages over the conventional scheme in this example. We will see the advantage of the proposed scheme in the next example.

Let us imagine a case in which a mobile with a constant velocity comes toward a base station during half of the observation time and goes away in the opposite direction during the other half. In this case, the phase error value will show odd symmetry, and its value at the middle of the observation period will be zero as shown in Fig. 10. Therefore, the average value during the observation time will be zero.

Figure 10 shows the estimated phase value of the conventional and proposed schemes in this case. We plotted the estimated phase value at the first iteration for the proposed estimator, and the one after the full iterations for the conventional estimator. The conventional scheme will estimate the phase error over a codeword as zero, and it is not changed even after several iterations. On the other hand, with the proposed estimator we can estimate the accurate phase value with only a single iteration.

Figure 11 shows the convergence behavior of the proposed phase estimator according to the iteration process. In Fig 11, we assumed the maximum phase error is 70 degrees in a codeword. We can see that a phase error of less than 30 degrees can be estimated and thus corrected at the first iteration. In the next process, the estimator corrects the residual phase errors. After just two phase estimation and recovery processes, we can correct all phase errors in a codeword. Figure 12 shows the number of iterations to lock the phase error, N_{pe}, of the conventional and proposed schemes. The conventional phase recovery scheme recognizes a phase error of zero. Even after the end of the

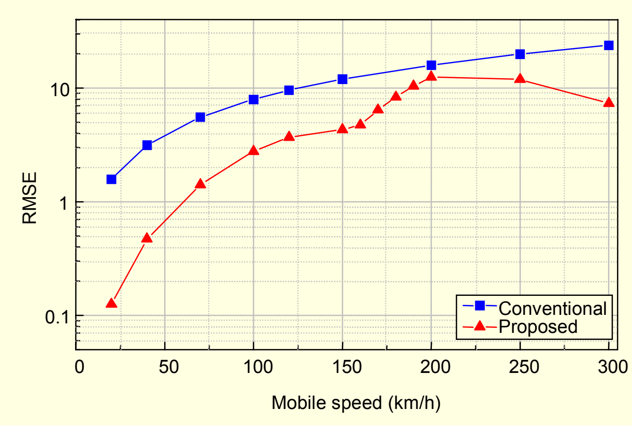

Fig. 9. RMSE performance of the phase estimator.

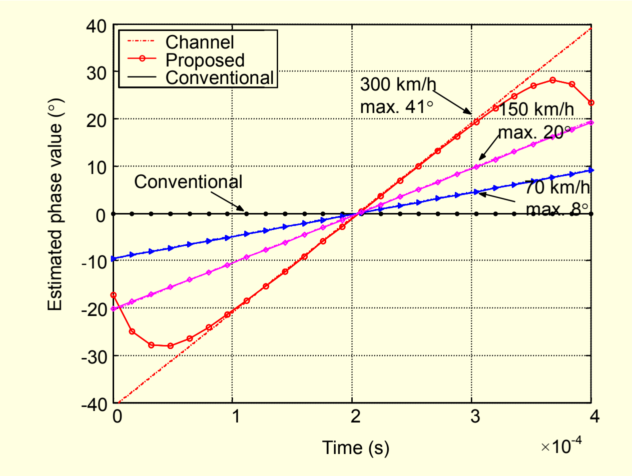

Fig. 10. Estimated phase in a codeword with the conventional and proposed schemes.

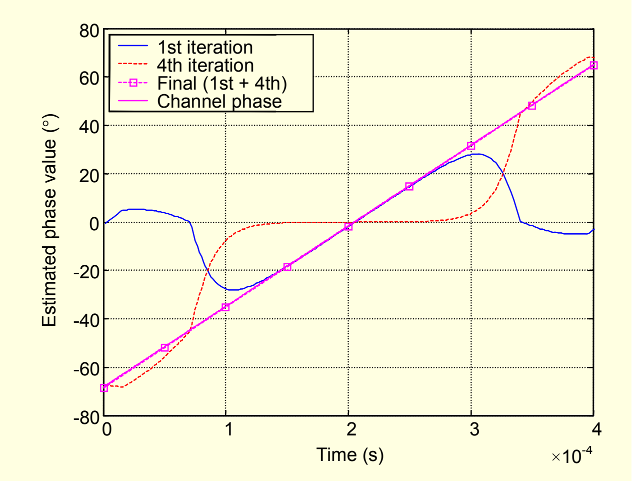

Fig. 11. Phase estimation behavior of the proposed algorithm.

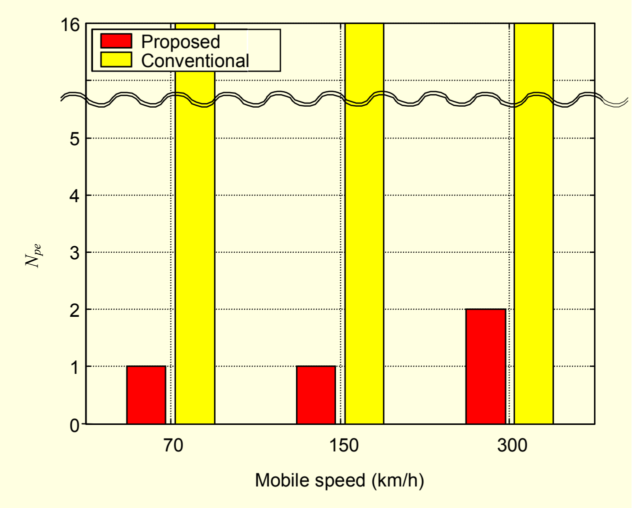

Fig. 12. Required number of iterations to lock the phase.

iterations, the conventional estimator cannot correct the phase error. However, the proposed scheme needs only 1 or 2 phase recovery iterations depending on the mobile speed.

Figure 13 demonstrates the RMSE performance for the above example. The RMSE of the conventional scheme with high speed mobiles has a large residual phase error as shown in Fig. 13. On the contrary, the proposed algorithm can effectively correct the phase errors precisely, and thus the RMSE converges to a relatively small value. The proposed scheme can also accommodate high speed mobiles with fast fading fluctuation.

Let us now discuss the complexity issues. Compared to the conventional iterative phase recovery schemes, the proposed scheme does not require additional computational complexities, but it does require additional memory to store the estimated phase values for the component codes. Because the piecewise phase recovery algorithm needs a fewer number of iterations, eventually the overall complexity of the proposed algorithm will be reduced.

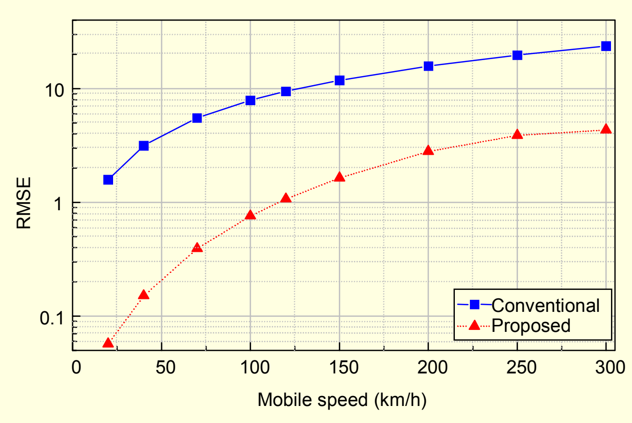

Fig. 13. RMSE performance of the phase estimator.

3. BER Performance

In this subsection, we compare the BER performance of the proposed algorithm to the conventional algorithm with various mobile speeds. Figure 14 shows the BER performance of the block turbo code without any phase recovery algorithm. A high mobile speed implies a large phase error. As we can easily guess, the BER performance worsens as the mobile speed increases.

Figure 15 shows a BER performance comparison of a block turbo code without a phase recovery algorithm, with a conventional phase recovery algorithm, and with the proposed algorithm when the mobile speed is 70 km/h. In this situation, we cannot see any difference in BER performance among the three algorithms. This is because the maximum phase error in a codeword is 10 degrees, and this can be corrected by using a

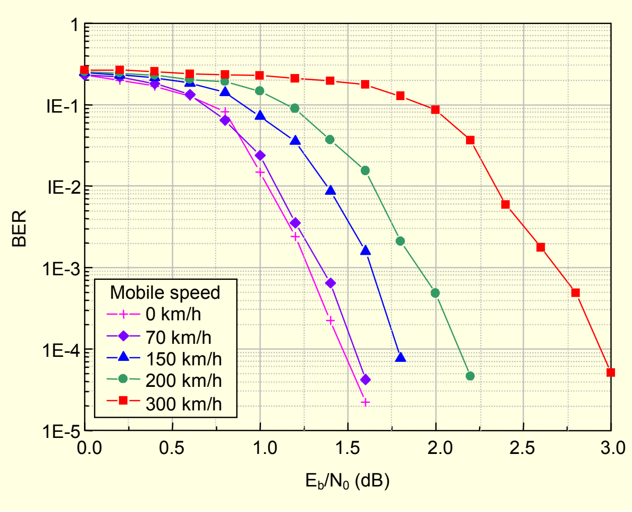

Fig. 14. BER performance comparison according to mobile speed.

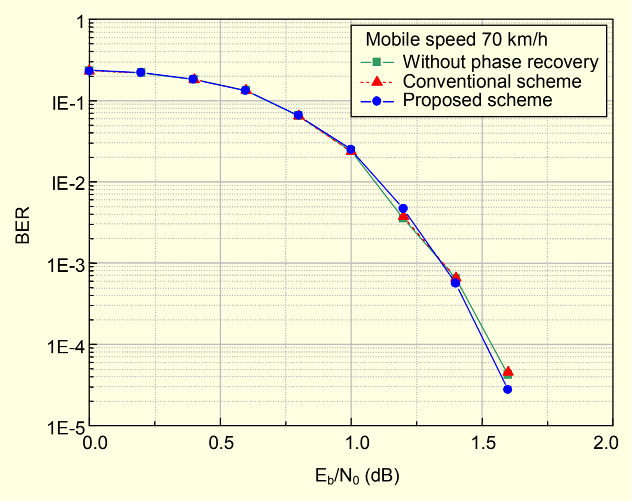

Fig. 15. BER performance with a mobile speed of 70 km/h.

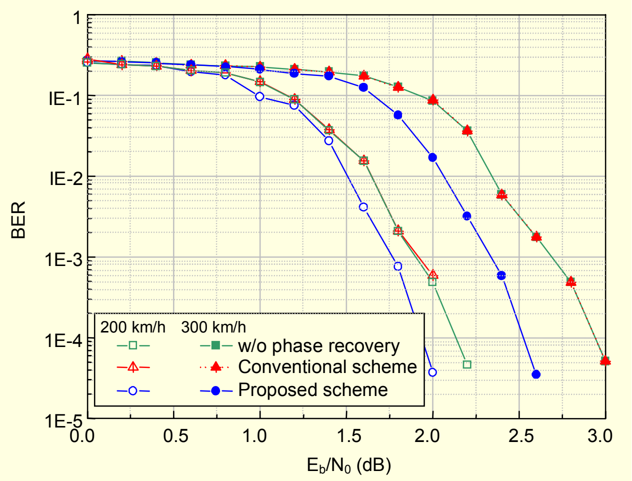

Fig. 16. BER performance with mobile speeds of 200 km/h and 300 km/h.

powerful channel coding scheme only.

However, if a mobile speed increases, we can see the advantages of the proposed algorithm. As shown in the previous section, the conventional phase recovery scheme has a larger residual phase estimation error in a fast time-varying channel condition. This leads to the BER performance degradation. Figure 16 shows a BER performance comparison of the proposed algorithm to the conventional algorithm when the mobile speeds are 200 km/h and 300 km/h. As we can see, the proposed algorithm shows stronger advantages with high mobile speeds.

IV. Conclusion

In this paper, we introduced an efficient carrier phase recovery algorithm combined with block turbo codes. Usually, a powerful code has a comparatively long length, and thus a phase error in a codeword cannot be assumed as a constant. In this situation where phase error fluctuates quickly, the proposed algorithm can efficiently correct the phase error by estimating it in a component code basis. In the simulation results investigated in this paper, it is shown that the proposed algorithm can enhance the time resolution of phase estimation by more than several tens of times. This results in reducing the number of decoding iterations and thus enhancing the BER performance, especially in high speed mobile terminals. For example, we can expect about a 0.5 dB power gain at a BER of about 10^{-5} with the proposed algorithm.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon Limit Error Correcting Coding and Decoding: Turbo Codes,” *Proc. ICC'93*, Geneva, Switzerland, May 1993, pp. 1064-1070.

[2] Kwangjae Lim, Sooyoung Kim, and Ho-Jin Lee, “Adaptive Radio Resource Allocation for a Mobile Packet Service in Multibeam Satellite Systems,” *ETRI Journal*, vol. 27, no.1, Feb. 2005, pp. 43-52.

[3] 3GPP TS 25.308 V5.4.0 (2003-03) High Speed Downlink Packet Access (HSPDA) Overall Description; Stage 2 (Release 5)

[4] IEEE P802.16e/D7, *Draft IEEE Standard for Local and Metropolitan Area Networks—part 16: Air Interface for Fixed and Mobile Boradband Wireless Access Systems*, April 2005.

[5] C. Morlet, M. L. Boucheret, and I. Buret, “Low-Complexity Carrier-Phase Estimator Suited on-Board Implementation,” *IEEE Trans.Commun.*, vol. 48, Sept. 2000, pp. 1451-1454.

[6] A. D’Amico, A. N. D’Andrea, and R. Reggiannini, “Efficient Non-Data-Aided Carrier and Clock Recovery for Satellite DVB at Very Low Signal-to-Noise Ratios,” *IEEE J. Select. Areas Commun.*, vol. 19, Dec. 2001, pp. 2320-2330.

[7] C. Morlet, I. Buret, and M. L. Boucheret, “A Carrier Phase Estimator for Multimedia Satellite Payloads Suited to RSC Coding Schemes,” *Proc. IEEE Int. Conf. Communications,* vol. 1, June 2000, pp. 455-459.

[8] C. Langlais and M. Helard, “Phase Carrier for Turbo Codes over a Satellite Link with the Help of Tentative Decisions,” *Proc. 2nd Int. Symp. Turbo Codes & Related Topics*, vol. 5, Sept. 2000, pp. 439-442.

[9] M. J. Nissila, S. Pasupathy, and A. Mammela, “An EM Approach to Carrier Phase Recovery in AWGN Channel,” *Proc. IEEE Int. Conf. Communications*, vol. 7, June 2001, pp. 2199-2203.

[10] V. Lottici and M. Luise, “Embedding Carrier Phase Recovery into Iterative Decoding of Turbo-Coded Linear Modulations,” *IEEE Trans. Commun.*, vol. 52, no. 4, April 2004, pp.661-669.

[11] Eun A Choi, Ji-Won Jung, Nae-Soo Kim, and Deock-Gil Oh, “Complexity-Reduced Algorithms for LDPC Decoder for DVB-S2 Systems,” *ETRI Journal*, vol. 27, no. 5, Oct. 2005, pp. 639-642.

[12] R. M. Pyndiah, “Near-Optimum Decoding of Product Codes: Block Turbo Codes,” *IEEE Trans. Commun.*, vol. 46, no. 8, 1998, pp. 1003-1010.

[13] S. Kim, W. S. Yang, and H. Lee, “Trellis-Based Decoding of High-Dimensional Block Turbo Codes,” *ETRI Journal*, vol. 25, no.1, Feb. 2003 pp. 1-8.

[14] S. Ryoo, S. Kim, and D. S. Ahn, “Efficient Carrier Phase Recovery with Block Turbo Codes,” *VTC 2005 Fall*, Sep. 2005, pp1417-1420.

Sunheui Ryoo received the BS and MS degrees in electrical and electronics engineering from Pohang Science and Technology, Korea, in 2001 and 2003. In 2003, she joined ETRI, Korea as a member of Research Staff and has worked for the development of efficient transmission algorithms for the B3G/4G wireless communication system. Her main research activities are in the areas of adaptive transmission, synchronization and acquisition, cross-layer design, and channel coding.

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 in the Satellite Communication Technology Division of ETRI, Korea, from February 1990 to September 1991, she received the MS and the 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, and worked as a Team Leader of the Broadband Wireless Transmission Team until February 2004. She is currently in a faculty position in Chonbuk National University, Korea.

Do Seob Ahn received the BS and MS degrees in electronics engineering from Kyungpook National University in 1988 and 1990. He has worked for ETRI since 1990, where he is currently a principle member of the Research Staff in the Digital Broadcasting Research Division and is working in the area of stratospheric communication systems and satellite communication systems.