ADAPTIVE SLIDING WINDOW METHOD FOR TURBO CODES IN CDMA CELLULAR SYSTEM WITH POWER CONTROL ERROR

Sook Min Park, Sang Sic Yoon, Sang Wu Kim, Kwyro Lee
Dept. of E.E., Korea Advanced Institute of Science and Technology
307-1 Kusung-dong, Yusung-gu, Taejon, Korea 305-701

ABSTRACT

This paper presents a method that can be used to reduce the decoding computational complexity in turbo codes. To reduce the decoding complexity we proposed an adaptive sliding window method which control the learning period of Viterbi sliding window method depending on channel signal to interference ratio (SIR). When received signal to interference ratio (SIR) is relatively high, we can reduce the decoding complexity without a noticeable degradation of BER performance at CDMA cellular system with power control error.

1. INTRODUCTION

Turbo codes has been studied extensively in 1990's. By combining a concatenation of convolutional codes, connected by an interleaver, with an iterative decoding algorithm, these codes achieve good BER performance close to information-theoretic limits [1]. Since these powerful codes can achieve near-Shannon-limit performance, they have been adopted as an optional coding technique for next-generation CDMA systems. But its decoding complexity and decoding delay have been problems in implementation. A considerable amount of work has been done for reducing the complexity and delay of turbo codes. Recently Viterbi proposed the sliding window method, which is widely used for preventing pipeline delays [2]. However it increases the amount of computation of reverse state metric, i.e. β, by twice. In this paper we proposed a method, which can reduce the amount of computation of β without degradation of average BER performance in CDMA cellular systems with power control error. In a practical power control system, the received power of signal may not be constant. The performance of a power control system depends on the speed of adaptive power control, dynamic range of the transmitter and propagation statistics. These factors influence the probability density function (PDF) of the received signal power. It can reasonably be assumed that the PDF of the received signal power is log-normal distribution [3]. The received signal power is varying with variance σ_i^2, which can measure imperfection in power control. The average BER performance depends on the condition of a poor received power more than relatively high received power. When the power of received signal is higher than average signal power we have a margin to reduce computation complexity without impacting average BER performance. In section 2 a brief description of the decoding algorithm of turbo codes and Viterbi sliding window method are given. System model of adaptive sliding window method is described in section 3. Our method is proposed in section 4. Simulation results are presented in section 5 and conclusions are given in section 6.

2. DECODING ALGORITHM OF TURBO CODES

2.1 MAP Algorithm

A block diagram of a turbo codes is shown in Fig. 1. The turbo encoder is implemented with two recursive, systematic convolutional (RSC) encoders in parallel concatenation. A frame of N information bits is encoded by the first encoder, while the interleaver creates a prespecified, random-like permutation of information, which is then encoded by second encoder. The transmitted code sequence consists of the information bits along with the parity bits produced by the two encoders. Each constituent decoder generates soft outputs in the form of a *posteriori* probabilities (APP) for information bits. From these probabilities, the decoder extracts "extrinsic information" values that are provided to the other decoder as soft inputs that play the role of a *priori* probabilities for the information bits. The output is a hard-quantized log-APP ratio of an information bit u_n produced by the final decoding cycle. More precisely, the a posteriori log-likelihood ratio (LLR) of an information bit u_n is expressed as (1).

$$\lambda_n = \log \frac{\Pr(u_n = 1 \mid R_1^N)}{\Pr(u_n = 0 \mid R_1^N)} \qquad (1)$$

Where $R^N_1 = (R_1, R_2, \ldots, R_N)$ denotes the received observations, with $R_i=(x_i, y_i, Le_i)$ consisting of the information samples x_i, the parity bit samples y_i and the extrinsic information Le_i. The APP decoder computes the a posteriori probabilities as (2).

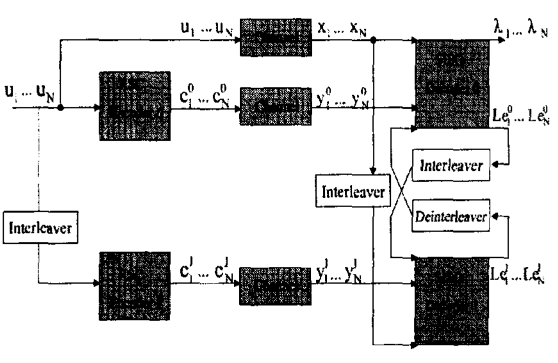

Figure 1. Turbo Codes Encoder, Channel, and Turbo Codes Decoder

$$\Pr(u_n = i \mid R_1^N) = \frac{1}{\Pr(R_1^N)} \sum_{m,m',u_n=i} \Pr(u_n = i, S_n = m, S_{n-1} = m', R_1^N) \quad (2)$$

Here S_n refers to the state at time n in the trellis of the constituent convolutional code. The terms in the summation can be expressed as (3)~(6).

$$\Pr(u_n = i, S_n = m, S_{n-1} = m', R_1^N) = \alpha_{n-1}(m')\gamma_n^i(m',m)\beta_n(m) \quad (3)$$

$$\gamma_n^i(m',m) = \Pr(S_n = m, u_n = i, R_n \mid S_{n-1} = m') \quad (4)$$

$$\alpha_n(m) = \Pr(S_n = m, R_1^N) \quad (5)$$

$$\beta_n(m) = \Pr(R_{n+1}^N \mid S_n = m) \quad (6)$$

We call γ_n^i as a branch metric, α_n as a forward state metric and β_n as a reverse state metric. The forward and reverse state metrics are computed recursively by forward and backward recursion given at (7),(8).

$$\alpha_n(m) = \sum_{m',i} \alpha_{n-1}(m')\gamma_n^i(m',m) \quad (7)$$

$$\beta_{n-1}(m') = \sum_{m,i} \beta_n(m)\gamma_n^i(m',m) \quad (8)$$

2.2 Viterbi Sliding Window Method

The reverse state metric calculation through the trellis from the last step of trellis results in a large pipeline delay and memory length. The key idea of the Viterbi sliding window method is that the reverse state metric calculation, which is performed via a backward recursion through the trellis as (8), does not have to start from the last time step of the trellis. Through the use of a sliding window of some length L and starting from some time point k in the trellis, reverse state metric calculations through L time steps will produce a good approximation of the reverse state metrics at time step k-L. The next L reverse state metrics can then be calculated starting from the approximation at time k-L. This method can be implemented with dual reverse state metric calculators to prevent pipeline delays. Thus this method increases decoding computational complexity. Table I shows the timing information for the various metric calculators, where FSMC is the forward state metric calculator, RSMC0 and RSMC1 are the reverse state metric calculators, and output of LLR calculator at time k is denoted λ_k. The dark areas of the table indicate the learning period to construct approximate reverse state metrics. Table II shows the calculation procedure for each time in case of no learning period, which results in same pipeline delay and memory length comparing with Viterbi sliding window method. Since there is no learning period, RSMC1 is not needed. Consequently the computational amount for calculating β has been reduced to 50% of Viterbi sliding window method. Table III shows the number of operation per information bit for Log-MAP algorithm, where I is the number of iterations, M is the memory size of a RSC encoder and LLC is the LLR calculator. We can find that the computation complexity of dual reverse state metric calculators is considerable amount of total decoding complexity.

Time	2L	3L	4L	5L	6L	7L	8L
FSMC	0-L	L-2L	2L-3L	3L-4L	4L-5L	5L-6L	
RSMC0	2L-L	L-0	4L-3L	3L-2L	6L-5L	5L-4L	
RSMC1		3L-2L	2L-L	5L-4L	4L-3L		6L-5L
Output		$\lambda_L \sim \lambda_0$	$\lambda_{2L} \sim \lambda_L$	$\lambda_{3L} \sim \lambda_{2L}$	$\lambda_{4L} \sim \lambda_{3L}$	$\lambda_{5L} \sim \lambda_{4L}$	$\lambda_{6L} \sim \lambda_{5L}$

Table I. Viterbi Sliding Window Method Pipelining Timing [6].

Time	2L	3L	4L	5L	6L	7L	8L
FSMC	0-L	L-2L	2L-3L	3L-4L	4L-5L	5L-6L	
RSMC0		L-0	2L-L	3L-2L	4L-3L	5L-4L	6L-5L
Output		$\lambda_L \sim \lambda_0$	$\lambda_{2L} \sim \lambda_L$	$\lambda_{3L} \sim \lambda_{2L}$	$\lambda_{4L} \sim \lambda_{3L}$	$\lambda_{5L} \sim \lambda_{4L}$	$\lambda_{6L} \sim \lambda_{5L}$

Table II. Pipelining Timing of Viterbi Sliding Window Method without Learning Period.

	additions	mult.	max ops	look-ups
FSMC	$4I \cdot 2^M$		$2I \cdot 2^M$	$2I \cdot 2^M$
RSMC0,1	$8I \cdot 2^M$		$4I \cdot 2^M$	$4I \cdot 2^M$
LLC	$4I \cdot 2^M + 2I$	$2 \cdot 2^M$	$8I \cdot 2^M - 2I$	$8I \cdot 2^M - 2I$

Table III. The number of operation per information bit for Log-MAP algorithm, where I is the number of iterations, M is the memory size of a RSC encoder.

3. SYSTEM MODEL

The system model to be applied in proposed scheme is the CDMA cellular system with power control error. Because of the restriction of the current cellular system technology and varying channel, power control cannot be perfect. It can be assumed as a log-normal distributed, that is normal distributed in dB scale, random variable [5]. If S_{pce} denotes the power of signal when power control error exist, its distribution can be expressed as following:

$$f(S_{pce}) = \frac{1}{\sqrt{2\pi}\sigma_{pce}S_{pce}} \cdot \exp\left(-\frac{(\ln S_{pce})^2}{2\sigma_{pce}^{\ 2}}\right). \tag{9}$$

In the above formula, S_{pce} has zero mean and standard deviation σ_{pce}. S_{pce} can be modeled as a normal distributed random variable with 0dB mean and variance 1~2dB [7]. Received signal can be expressed as following:

$$\frac{E_b}{I_0} = \overline{\left(\frac{E_b}{I_0}\right)} + S_{pce}[dB] \tag{10}$$

where E_b is an energy of bit and I_0 is an interference power.

4. PROPOSED ADAPTIVE SLIDING WINDOW METHOD

Formula (10) shows that the received power is varying with variance σ_{pce} by power control error. The average SIR is determined by target BER. The average BER performance dominantly depends on the condition of a poor received power. Therefore, when a received power is relatively high, reduction of computational complexity has a less effect on the target BER. Viterbi sliding window method has a learning period to construct approximate reverse state metrics. Without learning period as described in table II, BER performance degrades as Fig. 2. The simulation condition of Fig. 2 is described in section 5. However only one RSMC can be used, result in 50% computational reduction of reverse state metric. So, we controlled the learning period considering the received power. If S_{pce}[dB] is higher than pre-specified threshold i.e.δ_c[dB], we only use single RSMC. Otherwise we use dual RSMC, which is conventional Viterbi sliding window scheme. This proposed scheme is adaptive sliding window method. A comparison of BER performance between adaptive sliding window method and Viterbi sliding window method has been performed in section 5 with different values of threshold.

5. SIMULATION RESULTS

In this section we demonstrate the performance of proposed adaptive sliding window method in CDMA cellular system with power control error. The system model is given in section 3. That means the received instant signal power is normal distributed with variance σ_{pce}. Therefore the x axis of Fig.2 and Fig.3 is average SIR. A perfect SIR estimation at receiver is assumed in this paper. The turbo codes we choose is rate 1/3 binary turbo codes. We followed the specification of turbo codes which 3GPP(3rd Generation Partnership Project) suggested for IMT-2000. The two recursive convolutional constituent encoders have tap weight $\{13,15\}_8$. Multistage interleaver is used. Frame length is N=320. We employed Log-MAP algorithm with Viterbi sliding window method and 8 iterations for turbo decoding. Simulation for σ_{pce}= 1dB has been performed. In Fig.2 we can figure out that Viterbi sliding window method has 0.6dB coding gain at 10^{-5} bit error rate than Viterbi sliding window method without learning period. Fig.3 shows BER performance with adaptive sliding window method at the CDMA cellular system with power control error. There is only 0.029 dB (δ_c=0dB) and 0.078dB (δ_c=-0.8dB) gain loss than Viterbi sliding window method at 10^{-5} bit error rate. If δ_c is 0dB, it means adaptive sliding window method use single RSMC with probability of 50%. So we figure out that adaptive sliding window method results in about 25% reduction of computation of reverse state metrics at δ_c=0dB and 39.45% reduction at δ_c=-0.8dB.

6. CONCLUSIONS

Adaptive sliding window method has been proposed to reduce the computational complexity of turbo codes. The key idea of our method is changing decoding complexity depending on varying SIR condition. To change decoding complexity, we control learning period of Viterbi sliding window method. Consequently we showed adaptive sliding window method achieves almost the same performance as Viterbi sliding window method with less decoding computational complexity.

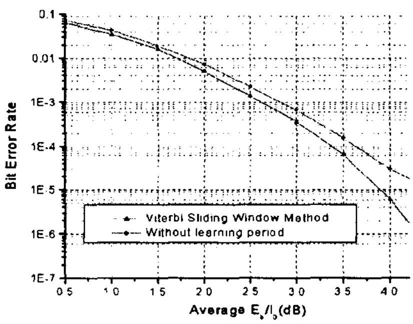

Figure 2. Performance comparison between Viterbi Sliding Window Method and Viterbi Sliding Window Method without learning period (σ_{pce}=1dB , N=320)

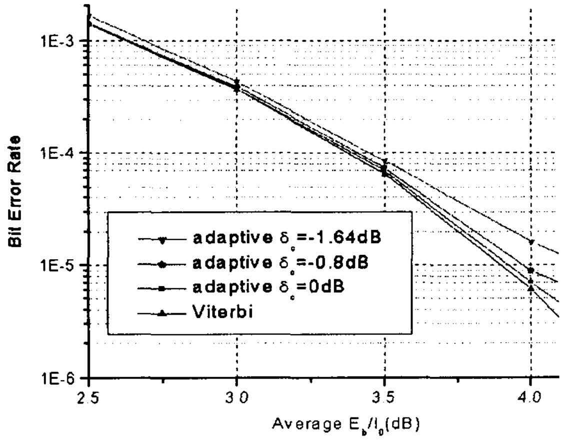

Figure 3. Performance comparison between Sliding Window Method with various thresholds and Viterbi Sliding Window Method. (σ_{pce}=1dB, N=320)

7. REFERENCES

[1] C.Berrou and A. Glavieux, "Near Shannon limit error correcting coding and decoding: Turbo-codes", *Proceedings of ICC*, pp. 1064-1070, May 1993

[2] Andrew J. Viterbi ,"An Justification and a Simplified implementation of the MAP Decoder for Convolutional Codes," *IEEE Journal on Selected Areas in Communications*, Vol.16, No.2, Feb. 1993

[3] Andrew J. Viterbi, "Performance of Power-Controlled Wideband Terrestrial Digital Communication," *IEEE Transactions on Communications*, Vol.41, No. 4, April 1993

[4] L. R. Bahl, et al, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Transactions on Information Theory*, Vol. IT-20 , Mar. 1974

[5] R.Prasad, A.Kegel and M.G.Jansen, "Effect Of Imperfect Power Control On Cellular Code Division Multiple Access System," *Electronic Letters*, 23rd, Vol.28, No.9, April 1992

[6] Steve Halter et al, "Reconfigurable Signal Processor for Channel Coding & Decoding in Low SNR Wireless Communications," *Signal Processing Systems*, pp. 260-274, 1998

[7] S. Gunaratne et al, " Performance of SIR-based power control for UMTS," *3G Mobile Communication Technologies*, pp. 16-20, 2001

ACKNOWLEDGEMENT

This work was supported by MICROS Research Center in KAIST