Differentiated Assignment of Extrinsic Information Based on Code Weights of Single Parity Check Product Code

Wonsun Bong, and Yong Cheol Kim Dept. of Electrical and Computer Engineering, University of Seoul, Korea
Conning-dong, 90, Dongdaemun-gu
Emails: gaam@uos.ac.kr, yckim@uos.ac.kr

***Abstract*—Constant amplitude multi-code CDMA (CAMC) removes the amplitude fluctuation in multi-code CDMA, by which the large power consumption of a power amplifier with a strict linearity can be avoided. The nature of CAMC is a recursive single parity check product code (SPCPC). CAMC for $N = 4^k$ is equivalent to k-dim SPCPC. As a top-level CAMC codeword is recursively constructed from lower-level codewords, log likelihood ratio (LLR), *a priori* information (API) and *extrinsic* information (EI) of upper-level codewords are obtained as a function lower-level codewords. The codewords of CAMC were found to have fixed code weights, $(N \pm \sqrt{N})/2$. In this paper, we show that differentiated assignment of EI in the iterated decoding can boost the BER performance. If any of the lower-level codewords have weights other than the fixed value, the EI associated with these erroneous lower-level codewords has lower confidence. EI from a wrong codeword is given lower weights in the computation of LLR, API and EI of the upper-level codeword. With differentiated assignment of EI, the BER performance improved by 0.1∼0.3 dB.**

Keywords*—*Weight Distribution, Turbo Code, Extrinsic Information, Single Parity Check Product Code

I. Introduction

Multi-code CDMA (MC-CDMA) assigns multiple channels to a single user. Since the signal of MC-CDMA is a linear sum of random binary signals, it has a large amplitude variation. To minimize the amplitude distortion, a power amplifier with highly linear characteristics is required, which entails a large power consumption.

Constant amplitude multi-code CDMA (CAMC) perfectly removes the amplitude fluctuation MC-CDMA. Kim [1] presented a recursive single parity check product code (SPCPC), which was originally introduced as CAMC.

SPCPC has the same basic features of turbo codes: interleaving, iteration and soft-output decoding [2]. Rankin presented a soft input, soft output-based decoding algorithm for SPCPC [3]. In the encoder, a parity bit is appended to each of $(n-1)$-bit sequences along all the dimensions of a Q-dimensional hypercube consisting of $(n-1)^Q$ information bits. The encoded output of n^Q bits is a Q-dimensional product code with a code rate of $(1-1/n)^Q$.

Hagenauer [4] developed a soft input, soft output-based decoding algorithm for a multi-dimensional product code. Rankin proposed to apply an iterative decoding to SPCPC [3]. He presented a decoding algorithm for SPCPC which refines the log likelihood ratio (LLR) for each bit by iteratively exchanging information in a relaxational scheme.

The decoding process of a product code is similar to solving a crossword puzzle. In the product array, one symbol is associated with two values through *diversity* effect: One is the very value of the received symbol itself and the other is the *extrinsic* value which can be inferred from the other symbols. With transmission errors, these two values may be different. This discrepancy can be relaxed in the iterative decoding process where the range of the possible values of a target symbol is adjusted through exchange of extrinsic information (EI) among neighboring symbols.

In this paper, we show that differentiated assignment of EI based on the correctness of code weight can improve the BER performance. It was shown that, unlike conventional SPCPC, the weights of CAMC codewords are evenly distributed at two fixed symmetrical values [6]. If the weight of a codeword is different from the fixed value, then it is evident that the codeword is at fault. The EI associated with these erroneous codewords has lower confidence.

In the proposed decoding scheme, a wrong codeword is given lower weights in the computation of EI associated with it. With differentiated assignment of EI based on the integrity of code weights, the BER performance improved 0.1∼0.3 dB when compared to previous works.

In Section II, observations from previous study are described. In Section III, CAMC and its weight distribution are briefly described. In Section IV, we describe the performance improvement by differentiated computation of EI. In Section V, computer simulation results is presented. Finally, a conclusion is drawn.

II. Observations from Previous Study

The performance of CAMC compared with that of corresponding SPCPC with $n = 4$ is shown in Fig. 1 [5]. BPSK modulated signal is transmitted through a binary-input AWGN channel. The code rates are $R_2 = 9/16$ (2-D), $R_3 = 27/64$ (3-D) and $R_4 = 81/256$ (4-D), respectively. For BER of 10^{-5}, CAMC is advantageous than SPCPC by 1.3∼1.4 dB.

Furthermore, the decoding of CAMC benefits from the processing gain in the despreading after the iterative decoding [6].

In this iterated decoding, EI plays an important role. As a rough estimate of the reliability for the noisy received

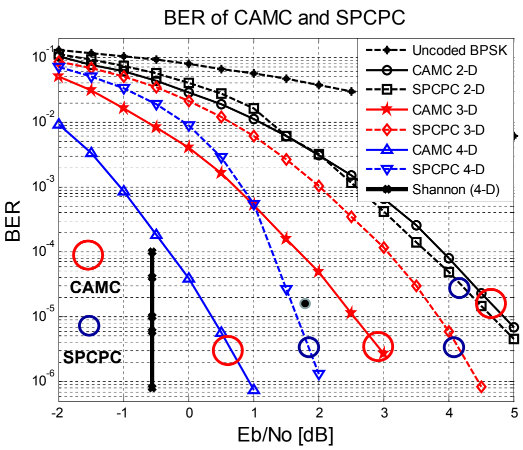

Fig. 1. CAMC, SPCPC and Shannon limit

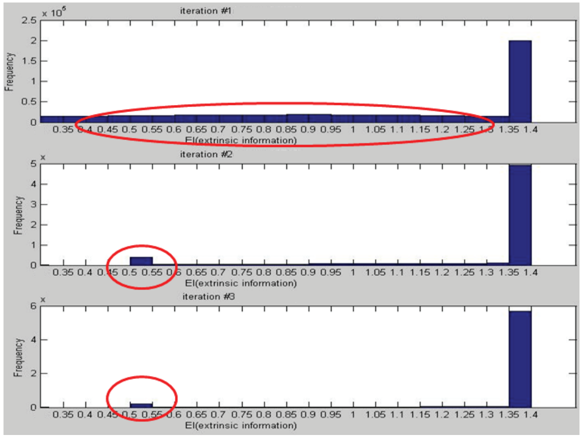

Fig. 2. Histograms of EI over iterations (3-D CAMC)

signal, EI supports the *good* component and suppresses the *bad* component out of the noisy signal.

We examined the distribution of EI over the iterative steps of turbo decoding [6]. Fig. 2 shows the gradual change of histograms of the values of EI. The initial values of the EI are rather randomly distributed. Over the iterations, they converge to either $(+E_{max})$ or $(-E_{max})$.

The distribution of EI is highly correlated with the bit error corrections. Fast convergence to $|+E_{max}|$ implies that errors continue to be corrected. On the contrary, slow convergence implies that soft decision values are close to zero. Practically no errors are being corrected if EI does not converge.

The weights of CAMC codewords are evenly distributed at two fixed symmetrical values [6]. N is the length of a codeword.

$$\mathrm{Wt} = (\mathrm{N} \pm \sqrt{\mathrm{N}})/2 \qquad (1)$$

Hence, code weight reflects the integrity of the codeword as EI represents the confidence of a target symbol. Weights for code length of $N = 4 \sim 64$ are shown in Table I.

TABLE I. WEIGHTS OF CAMC FOR $N = 4 \sim 64$

Code Length (N)	4	16	64	4^D
Dimension	1	2	3	D
Weights	1,3	6,10	28,36	$(N \pm \sqrt{N})/2$

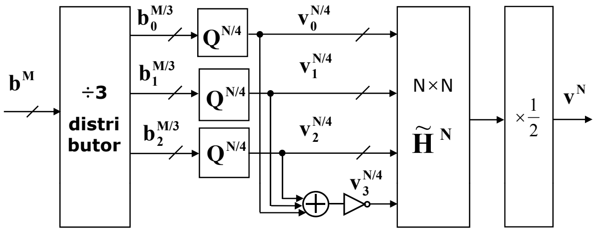

Fig. 3. Generation of N-bit CAMC vector

III. Constant Amplitude Multi-code

The generation of CAMC vectors is briefly described, with the polarity of a bit to be bipolar, either (+1) or (-1). An encoded CAMC vector at J-level has a length of $L(= 4^J)$ bits. Then, for every three J-level vectors, a bit-by-bit parity vector is generated. Spreading three J-level vectors and their parity vector generates a $(J+1)$-level vector with a length of $4L$ bits. On the other hand, despreading a J-level vector results in three $(J-1)$-level signal vectors and their parity vector, each with a length of $L/4$ bits.

In the following vector notations, a superscript represents the size of the vector and a subscript of $\{0, 1, 2, 3\}$, if any, represents the index of the four vectors.

Fig. 3 shows the top level encoder. An input of M bits is divided into $M/3$ groups of three bits each, $[b_0, b_1, b_2]$. At the basic level encoder $\mathbf{Q}^4$, a parity bit b_3 is appended as shown in Fig. 4 Then $[b_0, b_1, b_2, b_3]$ is spread by a 4×4 pseudo-Hadamard matrix, $\tilde{\mathbf{H}}^4$, into four bits of unit amplitude, $[v_0, v_1, v_2, v_3]$.

Continuing this way recursively, the output of three $\mathbf{Q}^{N/4}$ encoders ($3N/4$ bits) and their bit-by-bit parity vector ($N/4$ bits) are concatenated and then are spread by $\tilde{\mathbf{H}}^{\mathrm{N}}$, into N bits of unit amplitude, $[v_0, v_1, ..., v_{N-1}]$. In (2) and (3), concatenation of $\mathbf{a}$ and $\mathbf{b}$ is expressed as $[\mathbf{a} \mid \mathbf{b}]$. $\tilde{\mathbf{H}}^{\mathrm{N}}$ is an $\mathrm{N} \times \mathrm{N}$ pseudo-Hadamard matrix where $\mathbf{I}^{\mathrm{N}/4}$ is an $\mathrm{N}/4 \times \mathrm{N}/4$ identity matrix.

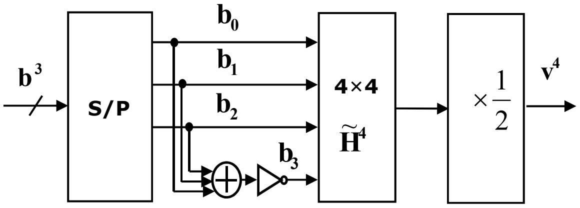

Fig. 4. Generation of 4-bit CAMC vector

$$\mathbf{v}_3^{N/4} = -\mathbf{v}_0^{N/4} \cdot \mathbf{v}_1^{N/4} \cdot \mathbf{v}_2^{N/4} \qquad \text{(bit by bit)} \tag{2}$$

$$\mathbf{v}^{N} = \frac{1}{2} \cdot [\mathbf{v}_0^{N/4} \mid \mathbf{v}_1^{N/4} \mid \mathbf{v}_2^{N/4} \mid \mathbf{v}_3^{N/4}] \cdot \tilde{\mathbf{H}}^{N} \tag{3}$$

$$\tilde{\mathbf{H}}^{N} = \begin{bmatrix} \mathbf{I}^{N/4} & \mathbf{I}^{N/4} & \mathbf{I}^{N/4} & \mathbf{I}^{N/4} \\ \mathbf{I}^{N/4} & -\mathbf{I}^{N/4} & \mathbf{I}^{N/4} & -\mathbf{I}^{N/4} \\ \mathbf{I}^{N/4} & \mathbf{I}^{N/4} & -\mathbf{I}^{N/4} & -\mathbf{I}^{N/4} \\ \mathbf{I}^{N/4} & -\mathbf{I}^{N/4} & -\mathbf{I}^{N/4} & \mathbf{I}^{N/4} \end{bmatrix}$$

IV. Differentiated Handling of Codewords

We propose to let EI quickly converge to $(+E_{max})$ or $(-E_{max})$ if the weight of the decomposed codeword is equal to the fixed value.

In the decoding of SPCPC, successive computation of EI, LLR and *a priori* information (API) iteratively refine the value of the received symbols. Since CAMC belongs to SPCPC, a similar decoding algorithm can be used for CAMC. The decoding algorithm in [3], however, cannot be directly applied to CAMC. Unlike SPCPC, there are no *raw* parity bits in a CAMC vector since parity bits are mixed with information bits through the spreading process.

Kim developed a decoding algorithm which separates parity bits and then iteratively refines the LLR of the received bits [5]. Parity bits are recursively extracted by despreading (decomposing) higher level CAMC vectors into lower level vectors. EI is also reconfigured accordingly when it is exchanged between dimensions. Finally, the iteratively decoded N bits are despread by N by N Hadamard matrix into M bits.

Fig. 5 illustrates the block diagram for the decoding of CAMC with code length= L. S(.) and D(.) represent the spreading process and the despreading process, respectively.

Expressions for LLR $(L_q(X_k^q))$, API $(A_q(X_k^q))$ and EI $(E_q(X_k^q))$ for CAMC are shown in (4), (5) and (6). X_k^q is the k-th bit of the vector reconfigured into the q-th dimension.

In the decoding of CAMC, lower level CAMC vectors are recursively extracted by decomposing higher level vectors. $[X_0^q, X_1^q, \ldots, X_{L-1}^q]$ and $[Y_0^q, Y_1^q, \ldots, Y_{L-1}^q]$, $(L = 4^q)$, is an input vector and the received vector reconfigured into the q-th dimension. The final decision on X_k^Q is obtained by hard-limiting of the top-level LLR.

A differentiated handling of EI boosts the performance of CAMC. In the proposed decoding scheme, those codewords with correct weights are given higher confidence over codewords with wrong weights. As shown in (5), EI takes on $(+E_{max})$ or $(-E_{max})$ if the weight of the decomposed codeword is equal to the fixed value in (1).

$$L_q(X_k^q) = \frac{2}{\sigma^2} Y_k^q + E_q(X_k^q) + A_q(X_k^q) \tag{4}$$

$$A_q(X_k^q) = \sum_{i=1}^{q-1} S(E_i(X_k^q)) + \sum_{i=q+1}^{Q} D(E_i(X_k^q)) \tag{5}$$

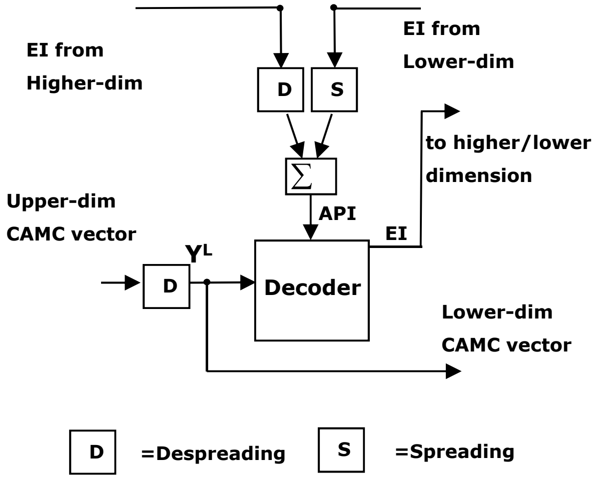

Fig. 5. EI is despread/spread between dimensions.

$$E_q(X_k^q) = \begin{cases} 2\tanh^{-1}\left[\prod_{j=0, j\neq k}^{N-1} \tanh\left(\frac{A_q(X_j^q) + \frac{2}{\sigma^2} Y_j^q}{2}\right)\right] \\ \qquad \text{for wrong code weight} \\ \pm E_{max} \\ \qquad \text{for correct code weight} \end{cases} \tag{6}$$

V. Results of Simulation

The performance of CAMC is tested in a computer simulation. Comparison with previous results [5], [6] is shown in Fig. 6 and Fig. 7. BPSK modulated signal is tested in binary-input AWGN channel with code rates through $R_2 = 9/16$ (2-D), $R_3 = 27/64$ (3-D) and $R_4 = 81/256$ (4-D).

In comparison with plain iterated decoding [5], a gain of 0.1∼0.3 dB in E_b/N_0 is achieved. In comparison with decoding with despreading control [6], a slight gain less than 0.1 dB is achieved. Most of these gains are in channels of high SNR. Under low SNR channel, little improvement is obtained. A logical analysis is as follows:

There are two possibilities of a code weight having the correct value in (1). One is that an erroneous codeword with even number of bit errors happens to have a correct code weight. When the channel SNR is low, this probability is not negligible and favored assignment of EI to this codeword does not help, but may even slow down the convergence of EI in the iterated decoding stages.

The other one is that the codeword is indeed error-free. This probability gets larger as SNR is higher. Preferred assignment of EI to these codewords help to improve BER performance.

This way, differentiated computation of EI depending on the believed integrity of a codeword helps to improve the performance.

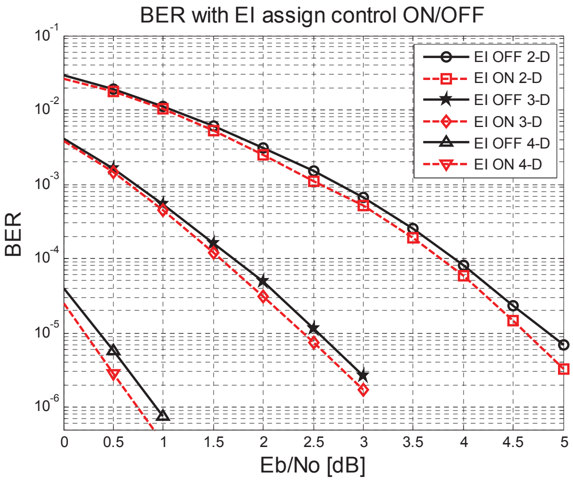

Fig. 6. BER with EI control

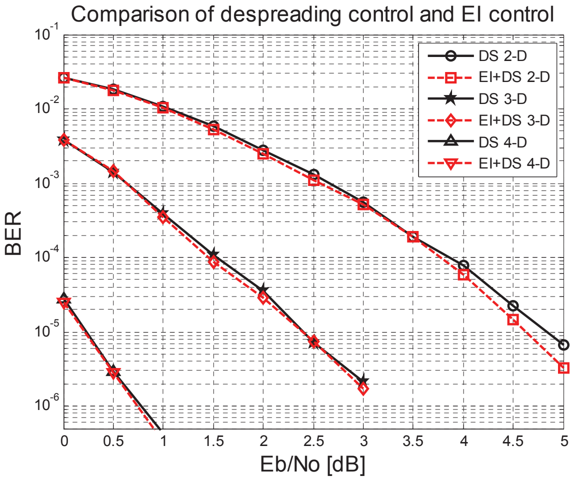

Fig. 7. BER with despreading control and EI control

VI. Conclusions

We showed that differentiated assignment of EI in the iterated decoding improves the BER performance of CAMC. In the proposed scheme, a codeword with a wrong code weight is given a lower confidence. By differentiated computation of LLR, API and EI based on the code weight, the BER performance improved by 0.1∼0.3 dB when compared to previous reports on plain turbo decoding of CAMC. The improvement is mostly found in high SNR channel.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology(grant number NRF-2013R1A1A2012745).

References

[1] Y. Kim, ”Recursive generation of constant amplitude multi-code DS-CDMA signal,” IET Electronics Letters, Vol. 39, No.25, Dec. 2003. pp.1782-1783

[2] G. Battail, ”A conceptual framework for understanding turbo codes,” IEEE J. Selected Areas Comm., Vol. 16, No.2 Feb. 1998, pp.245-254

[3] D. Rankin and T. Gulliver, ”Single parity check product codes,” IEEE Trans. Commun., Vol.49, No.8, pp.1354-1362, Aug. 2001

[4] J. Hagenauer, E. Offer and L. Papke, ”Iterative decoding of binary block and convolutional codes,” IEEE Trans. Inform. Theory, Vol.42, pp. 429-445, Mar. 1974

[5] Y. Kim, ”Constant amplitude multi-code CDMA with built-in single parity check product code,” IEEE Communications Letters, Vol. 10, No.1, 2006, Jan. pp.4-6

[6] K. Lee, I. Park, B. Kim Y. Kim, ”Processing Gain in a Recursive Single Parity Check Product C ode with Non-Gaussian Weight Distribution,” Proceedings of IEEE VTC 2009-Spring, Barcelona, Spain, Apr. 2009

[7] A. Shiozaki, M. Kishimoto and G. Maruoka, ”Close-to-capacity performance of extended single parity check product codes,” IET Electronics Letters Vol. 47 No. 1, Jan. 2011

Wonsun Bong received BS and MS degree from Chungju University in 2009 and from University of Seoul in 2011, all in electrical engineering. He is currently pursuing Ph.D. at the department of electrical and computer engineering in the University of Seoul. His research interests are wireless communication and signal processing. He is a student member of IEEK.

Yong Cheol Kim received BS degree in electronics engineering from Seoul National University in 1981 and MS degree in electrical engineering from KAIST in 1983. He received the Ph.D. degree in electrical engineering from University of Southern California in 1993. From 1993 to 1996, he was with LG Innotek. Since 1993, he was with the department of electrical and computer engineering in the University of Seoul. His research interests are mobile communications and image processing. He is a member of IEEE, ACM, IEEK, and KICS.