New Hybrid Concatenated Codes

Ya-Cheng Lu[1], Chin-Wen Shin[2], Chia-Jung Li[3], Erl-Huei Lu[4]

Department of Electrical Engineering, Chang Gung University, Tao-Yuan, Taiwan

***Abstract*— *To improve the performance of hybrid concatenated convolutional codes (HCCC); a modified Log-MAP algorithm and an enhanced HCCC are introduced and demonstrated to be efficient and practical by simulation results. The new coding scheme achieves about 1.0 dB additional coding gain, compared to the general turbo coding scheme at a BER = 10-6, with a frame length of 8192-bit. The system complexity and decoding latency of the new scheme is lower than the HCCC proposed by Divsalar and Pollara [6] within acceptable performance degradation. Since the bit-error-rate of the proposed HCCC can be dramatically reduced by slightly increasing signal-to-noise ratio, the new hybrid concatenated coding scheme is very suitable for those communication environments in which high reliability is important*.

Keywords*— *hybrid concatenated convolutional codes (HCCC), bit-error-rate, coding gain.

I. INTRODUCTION

Turbo codes, proposed in 1993 [1], show outstanding error correction performance and thus are widely used in communication areas, such as third-generation mobile and deep-space communications [2] [3]. The general encoder structure of turbo code is a parallel concatenated convolutional code (PCCC). Two component encoders are used to code the same input bits, but an interleaver is placed between the encoders. The decoder of PCCC is an iterative structure. Two component decoders, two interleavers as well as one de-interleaver are needed. During the iterative decoding procedure, only the extrinsic information of systematic bits is calculated and provided as the prior information of the decoded bits, however, the prior information of the parity bits has not been retrieved. Detail discussions for the kind of turbo codes have been given in [4].

Aiming for excellent BER performance, other concatenated codes, such as serial concatenated convolutional code (SCCC) [5] and hybrid concatenated convolutional code (HCCC) [6], are proposed. In [5] and [6], not only the extrinsic information of systematic bits, but also the extrinsic information of parity bits, is exchanged between the component decoders. Therefore, the performances of SCCC and HCCC have been shown to be superior, in some cases, to a general turbo code.

In this work, modifications of Log-MAP algorithm are presented. Based on the modified Log-MAP algorithm, an enhanced HCCC scheme is also proposed. Using the new scheme, both the extrinsic information of systematic bits and parity bits can be retrieved during iterative decoding. Thus, the decoding performance can be improved dramatically when signal-to-noise ratio (SNR) is slightly raised. With a frame length of 8192-bit, simulation results show that the enhanced scheme achieves an additional coding gain about 1.0dB over the turbo coding scheme, at a bit-error-rate (BER) = 10-6. Moreover, the decoding latency and system complexity is much lower than that of the Divsalar-Pollara HCCC [6] within acceptable performance degradation.

II. REVIEW OF LOG-MAP ALGORITHM

The performance of Log-MAP algorithm is almost same as that of MAP algorithm at a fraction of complexity [7]. Therefore it is an attractive algorithm to design the component decoder of turbo codes. Let uk be the message bit at time k and $\underline{y}=(y_1,y_2,\cdots,y_N)$ be the sequence of the received symbols. The LLR (log likelihood ratio) of bit uk is the soft-output, defined as:

$$L(u_k \mid \underline{y}) \triangleq \ln\left(\frac{P(u_k=+1 \mid \underline{y})}{P(u_k=-1 \mid \underline{y})}\right) = \ln\left(\frac{\sum_{\underline{u}:u_k=+1} P(\underline{u},\underline{y})}{\sum_{\underline{u}:u_k=-1} P(\underline{u},\underline{y})}\right)$$

$$= \ln\left(\frac{\sum_{(s',s)\to u_k=+1} \exp[A_{k-1}(s')+\Gamma_k(s',s)+B_k(s)]}{\sum_{(s',s)\to u_k=-1} \exp[A_{k-1}(s')+\Gamma_k(s',s)+B_k(s)]}\right) \tag{1}$$

where $P(u_k = \pm 1 | \underline{y})$ is the a-posteriori probability (APP) of $u_k = +1$ or $u_k = -1$, $\underline{u} = (u_1, u_2, \cdots, u_N)$ is the sequence of message bits. The s' and s denote states at the (*k-1*)th and the kth time stages in the trellis diagram, respectively. In Equation (1), the logarithm ratio of probabilities of $u_k = +1$ to $u_k = -1$ refers to the state transitions from $s' = S_{k-1}$ to $s = S_k$, denoted as ($s' = S_{k-1}$, $s = S_k$), for $1 \le k \le N$.

The branch metric $\Gamma_k(s', s)$ can be computed by

$$\Gamma_k(s^{'}, s) = C + \frac{1}{2} u_k \cdot L(u_k) + \frac{L_c}{2} \sum_{l=0}^{n-1} y_{kl} \cdot x_{kl} \tag{2}$$

where L(u_k) is the prior information of bit u_k which was given by the other component decoder, *Lc* is the channel reliability and x_{kl} is the codeword. When the initial condition $A_0(s)$ and the received sequence $y_{j<k}$ has been given, the forward metric $A_k(s')$ is calculated by the forward recursive computation:

$$A_k(s) = \ln\left(\sum_{all\ \ s^{'}} \exp[(A_{k-1}(s^{'}) + \Gamma_k(s^{'}, s)] \right) \tag{3}$$

When the $B_N(s)$ and the received sequence $y_{j>k}$ has been given, the backward metric $B_{k-1}(s)$ is calculated by the backward recursive computation:

$$B_{k-1}(s^{'}) = \ln\left(\sum_{all\ \ s} \exp[(B_k(s) + \Gamma_k(s^{'}, s)] \right) \tag{4}$$

For iterative decoding; Eq (2) can be decomposed as

$$\Gamma_k(s^{'}, s) = C + \frac{1}{2} u_k \cdot L(u_k) + \frac{L_c}{2} y_{k0} x_{k0} + \frac{L_c}{2} \sum_{l=1}^{n-1} y_{kl} \cdot x_{kl} = C + \frac{1}{2} u_k \cdot L(u_k) + \frac{L_c}{2} y_{k0} x_{k0} + \chi_k(s', s) \tag{5}$$

Substituting Eqs (3), (4) and (5) into (1), we can get

$$L(u_k | \underline{y}) = L(u_k) + L_c y_{k0} + L_e(u_k) \tag{6}$$

The *extrinsic information* $L_e(u_k)$, after interleaved (or de-interleaved), is to be used as the prior information $L(u_k)$ of systematic bit u_k for the other component decoder. Notably, there is not any *prior* information (or *extrinsic information*) of parity bits being calculated in the general Log-MAP algorithm. In the next section, we will modify Eqs (1) and (2) such that both the *extrinsic information* of the systematic and the parity bits can be calculated and used as the *prior* information in the iterative decoding.

III. MODIFIED LOG-MAP ALGORITHM

Since the soft outputs of a general Log-MAP algorithm do not involve the prior information of parity bits, this motivates us that the Log-MAP algorithm may be more efficient if both the prior information of systematic bits and parity bits can be provided before decoding operation is executed. Investigating Eq (1), it is evident that the a-posterior LLR of each bit in a codeword (including each of systematic bits and parity bits) can be computed by using the modified formula, as the following equation:

$$L(x_{kl} | \underline{y}) = \ln\left(\frac{\sum_{(s',s)\to x_{kl}=+1} \exp[A_{k-1}(s') + B_k(s) + \Gamma_k(s', s)]}{\sum_{(s',s)\to x_{kl}=-1} \exp[A_{k-1}(s') + B_k(s) + \Gamma_k(s', s)]} \right) \tag{7}$$

for $k = 1, 2, \ldots, N$ and $l = 0, 1, \ldots, n-1$; where N is the frame length and n is the number of codeword bits labeled on each branch in the trellis diagram. To compute the branch metric, Eq (2) can be modified by joining the *prior* information of parity bits as the following:

$$\Gamma_k(s', s) = C + \frac{1}{2}\sum_{l=0}^{n-1} x_{kl} \cdot L(x_{kl}) + \frac{L_C}{2}\sum_{l=0}^{n-1} y_{kl} \cdot x_{kl} \tag{8}$$

The computations for the forward metric $A_k(s)$ and the backward metric $B_{k-1}(s)$ are same as Eqs (3) and (4).

There are two points to be addressed. First, the prior information of both systematic bits and parity bits, or only one of them, can be used for computing branch metric by using Eq (8). Second, it needs to set $L(x_{kl}) = 0$ in (8) if the prior information of bit x_{kl} is unavailable.

Substituting Eqs (3), (4) and (8) into (7), the formula for calculating the *extrinsic information* of bit x_{kl} can be refined as

$$L_e(x_{kl}) = L(x_{kl} \mid \underline{y}) - L(x_{kl}) - L_c y_{kl} \tag{9}$$

IV. ENHANCEMENT OF HCCC

The encoder of enhanced HCCC is shown in Fig. 1 which consists of three identical recursive systematic convolutioanl (RSC) codes of code rate =1/2 (denoted by RSC1, RSC2 and RSC3, respectively) and two interleavers (denoted by π), as well as a puncturing and multiplexing unit (denoted by PM). As an example to illustrate the proposed HCCC, the generator polynomials of RSC codes are $1 + D + D^2$ and $1 + D^2$. In Fig. 1, the combination of RSC1 and RSC2 is exactly the same as a general parallel concatenated code; but RSC1 and RSC3 are concatenated serially. Since RSC3 encodes the interleaved parity output of RSC1, the prior information of the parity bits x_{kl} (where $l = 1$) can be produced by the corresponding component decoder, as the following description.

The iterative HCCC decoder of the new scheme consists of three modified Log-MAP component decoders (denoted by D_1, D_2 and D_3), four interleavers (denoted by π), and two de-interleavers (denoted by π^{-1}), as shown in Fig. 2. Each of component decoders is designed based on the modified Log-MAP algorithm in Section3.

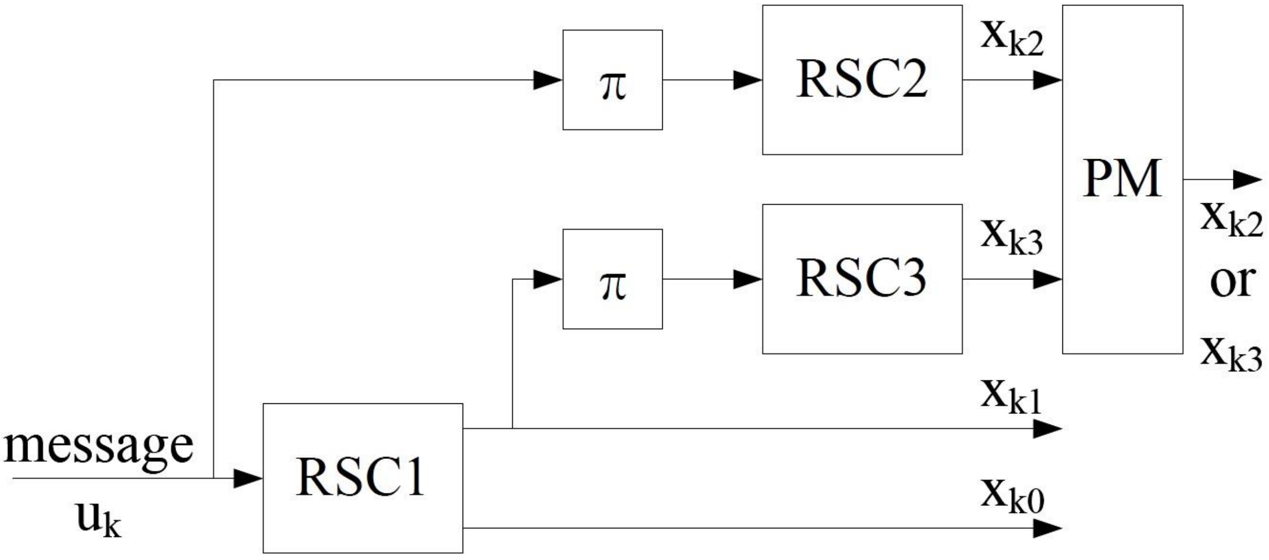

FIG.1 THE ENHANCED HCCC ENCODER SCHEME

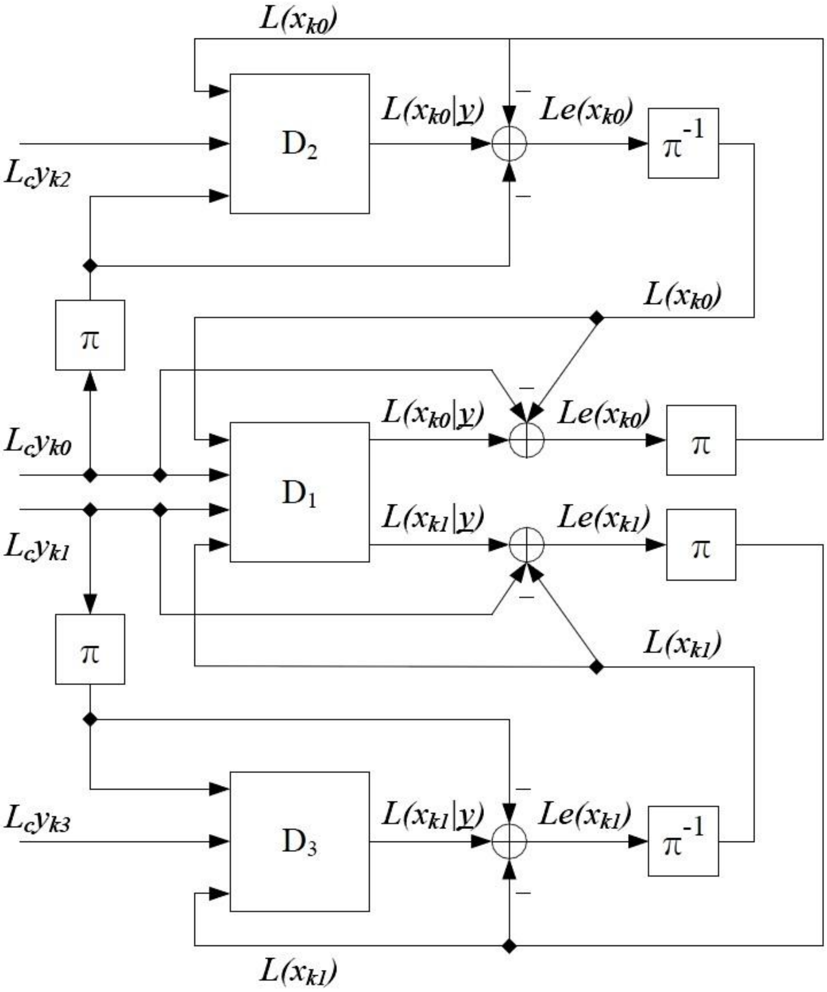

FIG.2 THE DECODER OF THE ENHANCED HCCC

First, D1 generates the *extrinsic information* $L_e(x_{k0})$ and $L_e(x_{k1})$ using y_{k0} and y_{k1}, for $k = 1, 2, \ldots, N$, as well as the prior information $L(x_{k0})$ and $L(x_{k1})$, where $L(x_{k0})$ and $L(x_{k1})$ are the *extrinsic information* generated by D_2 and D_3. The *extrinsic information* $L_e(x_{k0})$ and $L_e(x_{k1})$ of D_1 are sent to D_2 and D_3 as the *prior* information. Then, D_2 and D_3 generate the *extrinsic information* $L_e(x_{k0})$ and $L_e(x_{k1})$ using the related received sequence and the *prior* information. The detail iterative decoding procedure is as following:

In the first phrase of the first iteration, D_1 set $L(x_{k0}) = L(x_{k1}) = 0$ and takes the received systematic bits $L_c y_{k0}$ as well as parity bits $L_c y_{k1}$. Then D_1 computes $L(\mathrm{x}_{k0} \mid y)$ and $L(\mathrm{x}_{k1} \mid y)$ using Eq (7). The *extrinsic information* $L_e(x_{k0})$ and $L_e(x_{k1})$ are computed using Eq (9) and sent to D_2 and D_3 as the *prior* information of systematic bits x_{k0} and parity bits x_{k1}, respectively.

In the second phrase of the first iteration, D_2 and D_3 can operate simultaneously. That is, D_2 takes $L_c y_{k0}$, $L_c y_{k2}$ and $L(x_{k0})$, and D_3 takes $L_c y_{k1}$, $L_c y_{k3}$ and $L(x_{k1})$, as soft inputs to generate $L(\mathrm{x}_{k0} \mid y)$ and $L(\mathrm{x}_{k1} \mid y)$ using Eqs (3), (4), (8) and (7), simultaneously. Then the *extrinsic information* $L_e(x_{k0})$ and $L_e(x_{k1})$ are calculated using Eq (9), respectively. Finally, the $L_e(x_{k0})$ and $L_e(x_{k1})$ are sent to D_1 and taken as the prior information $L(x_{k0})$ and $L(x_{k1})$ of D_1.

During the second iteration, the enhanced HCCC decoder has the same decoding procedure as the first iteration, except D_1 receives $L_e(x_{k0})$ and $L_e(x_{k1})$ from D_2 and D_3, respectively, and taken them as its prior information. Once the second iteration is finished, other iterations can be continued until all predetermined iterations are performed or the decoding performance meets its stopping criteria [8]. Finally, the enhanced HCCC decoder decides message bits x_{k0} from $L(\mathrm{x}_{k0} \mid y)$ using a hard-decision operation.

There are several points to be addressed. First, the computation complexity of D_1 is only slightly heavier than that of a general Log-MAP decoder because all metrics for computing the LLR $L(\mathrm{x}_{k0} \mid y)$ and $L(\mathrm{x}_{k1} \mid y)$ are identical. Thus the overall system complexity of the new decoder is about 1.5 times compared with a general turbo decoder. Second, if the proposed decoder is implemented with a parallel architecture, the decoding latency is almost the same as a general turbo code because D_2 and D3 can decode received sequences simultaneously.

V. SIMULATION RESULTS

This section presents the performances of the proposed HCCC (PHCCC). The generator polynomials of component codes are $G_0 = 111$ and $G_1 = 101$. The length of random interleavers are *N*=1024, 4096 and 8192 bits and BPSK modulation over AWGN channel is used for simulation. And the Divsalar-Pollara HCCC of code rate = 1/4 [6] is constructed by an outer NSC code, a parallel RSC code and an inner RSC code, as well as two interleavers of length N and 2*N*.

Fig. 3 shows the BER performance of PHCCC and GTC with the frame length of 1024, 4096 and 8192 bits, in 8 iterations. For the frame of same length, the BER curve of PHCCC intersects that of GTC at a very low SNR. For example, the BER curves of GTC-8k and PHCCC-8k intersect at 0.7dB. Also, the BER performance of PHCCC is slightly worse than that of GTC when the SNR is below the intersection.

However, the PHCCC is much better than the GTC if SNR is larger than the intersection. For example, with the frame length of *N*=8192 bits, the PHCCC achieves about 1.0 dB additional coding gain compared to the GTC at a $BER = 10^{-6}$. Furthermore, the BER of PHCCC can be dramatically reduced when the SNR is increased slightly. This advantage is useful for applications in which high reliability is required.

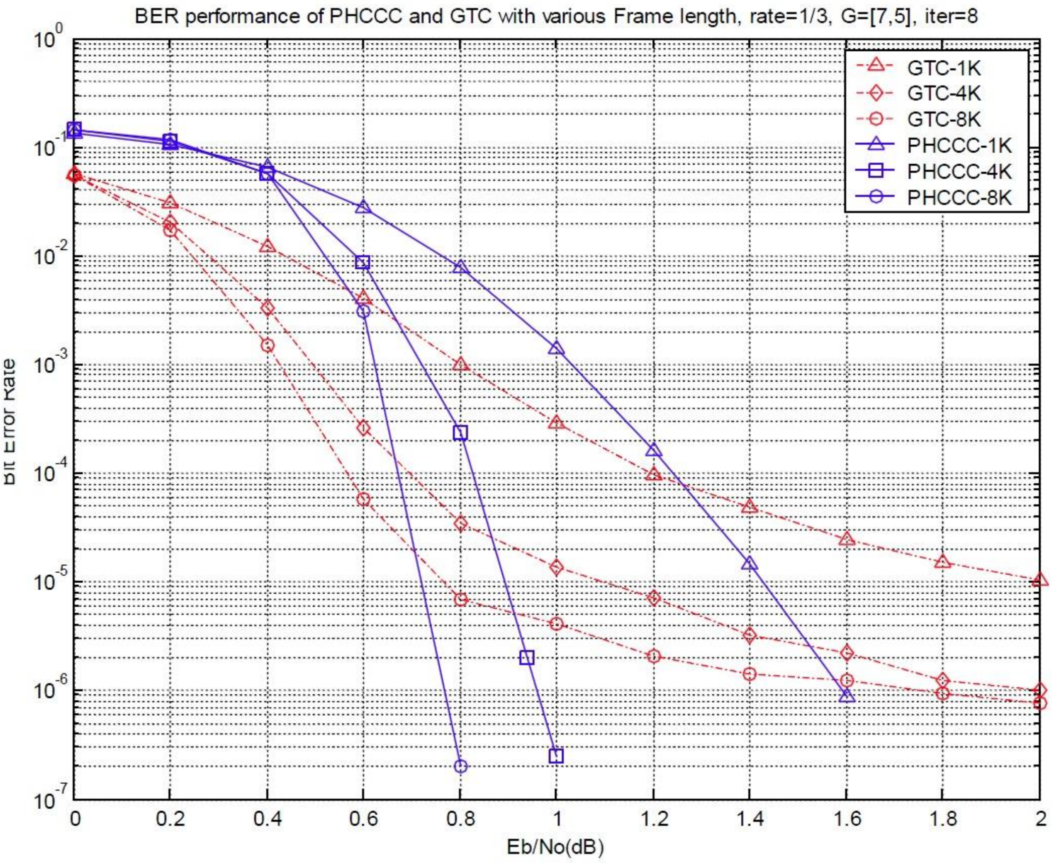

FIG.3 BER PERFORMANCE OF PHCCC, R=1/3

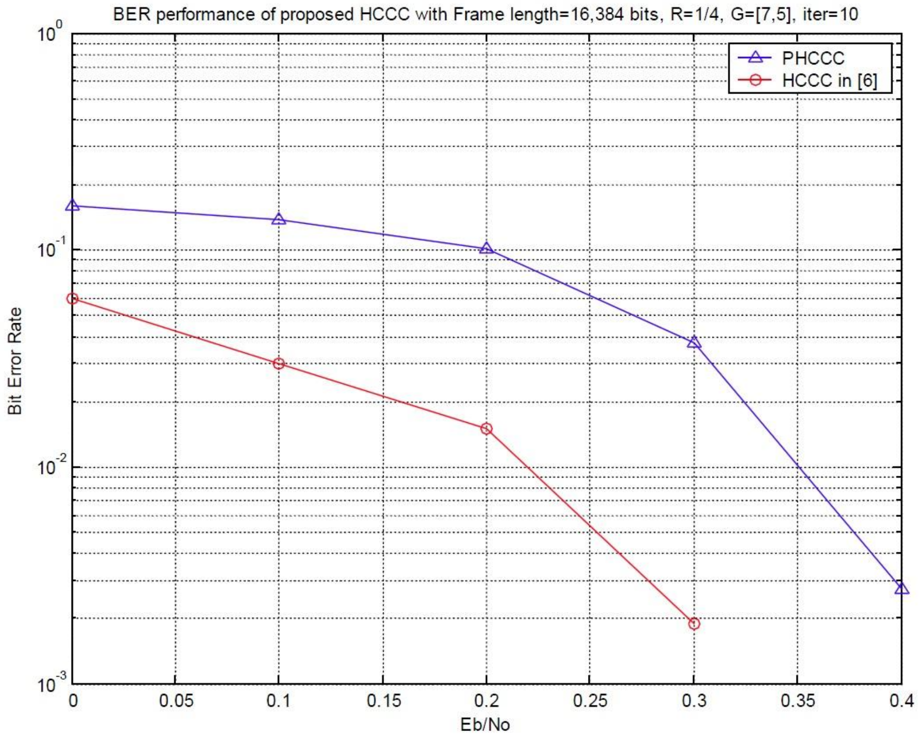

FIG.4 BER PERFORMANCE OF PHCCC, R=1/4

From Fig. 4, the BER performance of PHCCC is slightly worse than that of the HCCC in [6]. However, comparing the decoder structure of the PHCCC with that of the Divsalar-Pollara HCCC, there are some points worthy to be noted. The length of interleavers in the Divsalar-Pollara HCCC is different (i.e., N and 2N). Therefore, for each iteration, the decoding latency caused by interleavers and de-interleavers in the Divsalar-Pollara HCCC is 4*N*. Furthermore, if we assume the computation complexity of decoding a four-state convolutional code is ***u***, the computation complexity of the Divsalar-Pollara HCCC is 6***u***. The comparisons are summarized in Table. Thus, the system complexities and decoding latency of the PHCCC are lower than that of HCCC in [6] within acceptable BER degradation.

TABLE1
COMPARISONS OF THE PROPOSED HCCC SCHEME WITH THE HCCC IN [6]

	PHCCC	HCCC [6]
Interleaver length	N	$N/2N$
Latency of each iteration	$2N$	$4N$
Computation complexity	$3\boldsymbol{u}$	$6\boldsymbol{u}$
Code rate	1/3	1/4

VI. CONCLUSIONS AND REMARKS

An enhanced hybrid concatenated convolutional code (HCCC) is proposed based on a modified Log-MAP algorithm. Simulation results are given to demonstrate these modifications being efficient and practical to construct the HCCC. Although the BER of the proposed HCCC is slightly worse than a general turbo code when the SNR is very low, it is much better than the latter when the SNR is increased slightly. Comparisons of the proposed HCCC with the Divsalar-Pollara HCCC are given in Table1. From the table, it is easy to see that both the system complexity and the decoding latency of the proposed HCCC are much lower than the latter, with slight BER degradation.

REFERENCES

[1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes (1)", Proc. IEEE Int. Conf. Comm., pp. 1064–1070, Geneva, Switzerland, May 1993.

[2] Third Generation Partnership Project, www.3gpp.org.

[3] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Soft-output decoding algorithms in iterative decoding of turbo codes", *JPL TDA Progress Rep. 42-124*, pp. 63-87, Feb. 15 1996.

[4] L. Hanzo, T. H. Liew and B. L. Yeap, *Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over Fading Channels*, John Wiley & Sons Ltd, 2002.

[5] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: performance analysis, design and iterative decoding," *IEEE Trans. Info. Theory*, vol. 44, no. 3, pp. 909–926, May 1998.

[6] D. Divsalar and F. Pollara, "Hybrid concatenated codes and iterative decoding," *JPL TDA Progress Rep. 42-130*, pp. 1-23, Aug. 15 1997.

[7] P. Robertson, P. Hoeher and E. Villebrun, "Optimal and sub-optimal Maximum a Posteriori algorithms suitable for turbo decoding," European Transactions on Telecomm. (ETT), vol. 8, no. 2, 1997, pp.119-125.

[8] D. H. Kim and S. W. Kim, " Bit-level stopping of turbo decoding," *IEEE Comm. Lett.*, vol. 10, pp. 183–185, Mar. 2006.