

Low-Density Parity-Check Coded Recording Systems With Run-Length-Limited Constraints

Hsin-Yi Chen¹, Mao-Chao Lin^{1,2}, and Yeong-Luh Ueng³

¹Graduate Institute of Communication Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C.

²Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C.

³Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan, R.O.C.

We propose two techniques for the low-density parity-check (LDPC) coded partial response channel with run-length-limited (RLL) constraints. The first is a modification of the selective flipping technique so that side information is not needed. The second is based on the estimation of flipped bits for the selective flipping technique. The second technique can achieve significant performance improvement over the simple selective flipping technique either with side information or without side information. We also incorporate these two techniques into a known technique to design LDPC coded recording systems that can meet strict RLL constraints without performance degradation.

Index Terms—Constrained codes, low-density parity-check (LDPC) codes, recording, run-length-limited (RLL) codes.

I. INTRODUCTION

IN data storage systems, run-length-limited (RLL) coding [1], [2] is usually needed to avoid the adverse effect of intersymbol interference (ISI) and/or to facilitate the operation of synchronization. An RLL sequence is a binary sequence, for which the run length of consecutive 0's is under a given constraint. For example, in a (d, k) sequence, the number of consecutive 0's is at least d and at most k . An RLL code is a collection of RLL sequences, which may not have error-correcting capability.

Constructing RLL codes with error-correcting (EC) capabilities, i.e., RLL-EC coding scheme, is an interesting topic [3], [4]. A straightforward method to construct RLL code with EC capability can be divided into three stages. For the first stage, we convert a message sequence into a data sequence satisfying RLL constraint. For the second stage, a systematic EC encoder adds parity bits which may not satisfy the RLL constraint to the data sequence. For the third stage, the parity bits which may not satisfy the RLL constraint are converted into a parity sequence satisfying RLL constraint. However, the EC capability of the resultant RLL-EC coding scheme may be changed. In case that the RLL constraint is the $(0, k)$ constraint, there is a very simple method for which the EC capability of the resultant RLL-EC coding scheme will not be changed. We first encode a message sequence into a codeword of length n using the ECC (EC code) encoder and then we insert 1's into this codeword every k bits. The resultant code will satisfy the $(0, k)$ constraint. In [5]–[7], novel methods have been proposed to insert parity bits of the ECC into the message sequence which meets the $(0, k')$ constraint so that the resultant sequence will not violate the $(0, k)$ constraint, where $k' \leq k$.

In [8]–[11], the channel for data storage and detection is considered as a partial response (PR) channel. Since a combined ECC with PR channel can be considered as a concatenated coding system, we can combine the soft-in-soft-out (SISO)

equalizer for the PR channel and the SISO ECC decoder to achieve high reliability. Since the affect of intersymbol interference can be effectively removed in this concatenated structure, we can set $d = 0$ for the (d, k) sequence constraint. In particular, using a turbo code [12] or an LDPC code [13]–[15] as ECC which is combined with PR channel can achieve very high error-correcting capability [8]–[11]. In [16], the unconstrained LDPC codeword is deliberately bit flipped to satisfy the RLL constraint before being sent to the channel, while the deliberately flipped bits are expected to be removed by the powerful LDPC decoder. This scheme is novel in using the capability of ECC to eliminate the need of redundancy in RLL coding. This scheme can be applied to $(0, k)$ sequences with loose constraint, i.e., large k . In case of moderate or small k , the number of deliberately flipped bits will be so great that only a low rate LDPC code is powerful enough to correct them. To reduce the number of deliberately flipped bits, Li and Vijaya Kumar [10] proposed to try a number of test sequences which are respectively added (modulo 2) to the unconstrained LDPC codeword to find a resultant sequence which is likely to have acceptable number of flipped bits. The scheme proposed in [10] can be applied to $(0, k)$ sequence with moderate constraint, i.e., moderate k , and is an efficient one except for the need of side information which is required to satisfy the RLL constraint and requires error protection.

In this paper, we follow the thread of [10] and propose two techniques for removing the need of side information. In addition, we incorporate these two techniques into the technique in [5] to design LDPC coded recording systems which can meet strict RLL constraint without performance degradation. In Section II, we propose a side-information free LDPC coded PR channel design which uses a number of test sequences to reduce the flipped bits needed for satisfying the RLL constraint. In Section III, we present the proposed flipped bits estimation technique. In this way, we can reduce the number of test sequences needed for reducing deliberately flipped bits. In Section IV, we combine the techniques of Sections II and III with a method proposed in [5] so that we can achieve highly reliable data storage system with strict constraints. Concluding remarks are given in Section V.

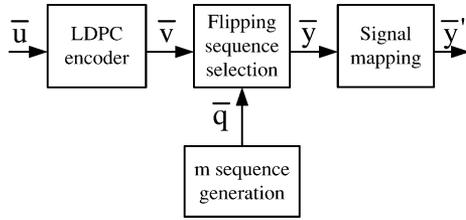


Fig. 1. Encoder for an LDPC coded recording system using the selective flipping technique.

II. SELECTIVE FLIPPING TECHNIQUES

Fig. 1 shows an encoder of a data recording system, where $\bar{u} = (u_0, u_1, \dots, u_{l_m-1})$ is an l_m -bit message to be stored in the recording system. We use \bar{u} as input to the encoder of an LDPC code, C_{ldpc} , to obtain the n -bit output $\bar{v} = (v_0, v_1, \dots, v_{n-1})$. The purpose of LDPC coding is twofold. The first is to enhance the reliability of the retrieved data. The second is to remove flipped bits. The sequence \bar{v} may not meet the RLL constraint. Let Q be a set of L maximum length sequences (m sequences) of length n . For each sequence $\bar{q} = (q_0, q_1, \dots, q_{n-1}) \in Q$, we have

$$\bar{v} \oplus \bar{q} = (v_0 \oplus q_0, v_1 \oplus q_1, \dots, v_{n-1} \oplus q_{n-1}). \quad (1)$$

We flip some bits of $\bar{v} \oplus \bar{q}$ to result in a sequence $\hat{v}(\bar{q})$ that meets the RLL constraint. Let $\bar{y} = (y_0, y_1, \dots, y_{n-1})$ be the $\hat{v}(\bar{q})$ such that the number of flipped bits, i.e., the weight of $\hat{v}(\bar{q}) \oplus \bar{v} \oplus \bar{q}$, is the smallest among all the $\bar{q} \in Q$. Each component $y_j, j = 0, 1, \dots, n-1$, in \bar{y} is either 0 or 1. Replacing each component that is equal to 0 by a component -1 , we convert the sequence \bar{y} into a corresponding sequence $\bar{y}' = (y'_0, y'_1, \dots, y'_{n-1})$. The sequence \bar{y}' will be the input of a PR channel with impulse response $\bar{h} = (h_0, h_1, \dots, h_m)$. At the reading side, we can use soft-in-soft-out (SISO) equalizer to exploit the effect of ISI caused by the PR channel. Since L candidates are used to reduce the number of flipped bits which are needed to meet the RLL constraint, we may call this multiple-candidate method a selective flipping (SLF) technique. This SLF technique for the LDPC coded PR equalized system with RLL constraint was proposed in [10]. Multiple-candidate method was also used in the technique of guided scrambling to design constrained codes with high efficiency in [17], [18].

A. Systems Using Side Information

In [10], side information \bar{s} is used to indicate the selected maximum length sequence \bar{q} . Note that \bar{s} needs to be coded into \bar{c}_s to acquire error protection and satisfy the RLL constraint. For example, we may use a (25,8) block code shortened from a (26,9) binary linear code [19] with minimum distance 9 to protect the 8-bit side information for indicating the candidate (maximum length sequence) selected from Q containing 256 candidates. To meet the RLL constraint, the code length (i.e., the length of \bar{c}_s) will be longer than 25. In case of the $(0, k)$ constraint with $k = 7, k = 6$ and $k = 3$, the code lengths of the error-protection code for side information, denoted C_s , should be increased to 29, 30, and 34 respectively, by periodically inserting 1's into the codeword of the (25,8) ECC. The input to the PR channel is actually the sequence pair (\bar{c}'_s, \bar{y}') instead of

simply \bar{y}' , where \bar{c}'_s is obtained from \bar{c}_s by replacing each component in \bar{c}_s equal to 0 by -1 . According to the channel model used in [7], [10], we have $\bar{z} = \bar{y}' * \bar{h} + \bar{w}$ and $\bar{z}_s = \bar{c}'_s * \bar{h} + \bar{w}'$ for detection, where $*$ denotes the operation of convolution, either \bar{w} or \bar{w}' is a vector representing additive noise for which each component is a zero mean Gaussian random variable with variance σ^2 .

We can apply the maximum likelihood decoder to the error-protection code for side information, C_s , to recover \bar{s} from \bar{z}_s . Then, we can find the selected candidate and employ the following algorithm for detection. Let I be the maximum number of iterations between the MAP (maximum *a posteriori*) detector (or equalizer) and LDPC decoder.

Algorithm SI

- Set the index of outer iteration $U_o = 1$.
- Step 1) The MAP detector for the PR channel takes \bar{z} and the *a priori* LLR (log-likelihood ratio) value for the bit y_i , denoted $L_a(y_i)$, as input. In each iteration, MAP detector for the PR channel generates the *a posteriori* LLR value for y_i , denoted $L_d(y_i)$, as output. The extrinsic value for y_i is obtained by $L_e(y_i) = L_d(y_i) - L_a(y_i)$.
- Step 2) Use $(L(v_0), L(v_1), \dots, L(v_{n-1}))$ as input to the decoder of the LDPC code C_{ldpc} , where $L(v_i) = L_e(y_i) \cdot (-1)^{q_i}$. After U_i iterations within the LDPC decoder, the LDPC decoder generates the *a posteriori* LLR value for v_i , denoted $L_D(v_i)$ as output.
- Step 3) Let $L_D(y_i) = L_D(v_i) \cdot (-1)^{q_i}$. The extrinsic value for the bit y_i from the decoder is obtained by $L_D(y_i) - L_e(y_i)$ which is used as the *a priori* LLR, i.e., $L_a(y_i)$, in the MAP detector. The resultant sequence is $(L_a(y_0), L_a(y_1), \dots, L_a(y_{n-1}))$ which will be used in the next iteration.
- Step 4) Increase U_o by 1. If the index of iteration $U_o = I$, then the hard decision of $L_D(v_i), i = 0, 1, \dots, n-1$, is used as the final estimate of \bar{v} . Otherwise, go back to Step 1.

Example 1: Consider the PR channel which is the EPR4 channel with impulse response $\bar{h} = (1, 1, -1, -1)$. Let the RLL constraint be the $(d, k) = (0, 7)$ constraint. Fig. 2 shows the bit error rate (BER) curve obtained by deliberate flipping as proposed in [16]. We can use the SLF technique in [10] to improve the error performance. Let the number of candidates, L , be 256. The side information is protected by a (29,8) block code C_s which is converted from a (25,8) code with distance 9 by adding bits to satisfy the (0,7) constraint. The number of message bits is $l_m = 888$. An (999, 888) LDPC code [14] is used. From Fig. 2, the BER obtained by using Algorithm SI and $L = 256$ is very close to the ideal case, i.e., the LDPC coded PR channel without any RLL constraint, where $U_i = 1$ and $U_o = 15$, E_b stands for the consumed energy per message bit and N_0 is the one-sided power spectral density of the additive white Gaussian noise. However, the SLF technique using side information has a rate loss of $100\% \times 29/(999 + 29) \approx 2.82\%$ as compared to this ideal case. To reduce the rate loss, we may

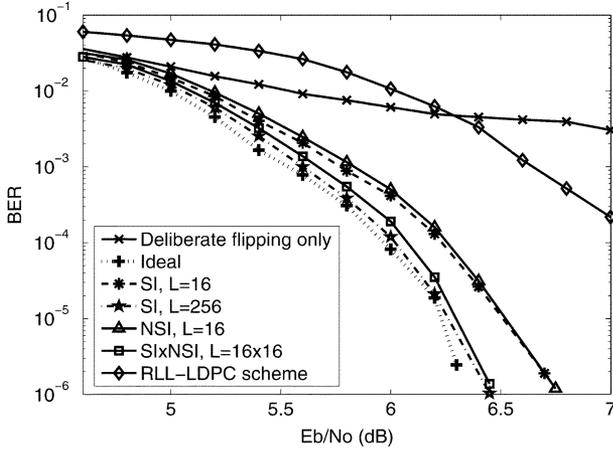


Fig. 2. Simulation results of (999, 888) LDPC coded EPR4 channel with (0,7) constraint ($U_i = 1, U_o = 15$).

use only 16 candidates instead of the original 256 candidates. To protect the 4-bit side information, we may use a (21,4) binary linear code with minimum distance 9 that is shortened from a (26,9) code [19]. To satisfy the (0,7) constraint, the code length needs to be extended to 24. Thus, the code rate loss will be $100\% \times 24 / (999 + 24) \approx 2.35\%$. However, as shown in Fig. 2, the BER obtained by using Algorithm SI and $L = 16$ is quite inferior to that obtained by using Algorithm SI and $L = 256$. □

We may compare Example 1 with Example 2 which is constructed by using the powerful RLL-LDPC concatenation scheme proposed in [5].

Example 2: According to [5, Table I], we can design a (18,17) RLL code, denoted C_1 , meeting the (0,7) constraint, for which 10 of the 18 bit positions in each codeword can be used as unconstrained bits, where each unconstrained bit means that we can arbitrarily assign 0 or 1 to this bit position without violating the constraint. In addition, we can design a (9,8) RLL code, denoted C'_1 , meeting the (0,7) constraint. Moreover, the concatenation of C_1 and C'_1 will still meet the (0,7) constraint. We can concatenate 55 C_1 and 1 C'_1 codeword to form a (999, 943) RLL code C_2 meeting the (0,7) constraint, for which 550 bit positions in each codeword can be used as unconstrained bits. At the beginning of encoding, we encode a 943-bit sequence which contains 888 message bits and 55 zero bits into a 999-bit codeword of C_2 . The 55 zero bits are located at the unconstrained bit positions. The other 944 bits of a RLL codeword of C_2 are encoded into a codeword of a (999, 944) systematic LDPC codeword, where the 55 parity bits will replace 55 zero bits in the unconstrained bit positions. In this way, we have an LDPC codeword of length 999 bits which meets the (0,7) constraint. The overall information rate is $888/999 \approx 0.889$ which is the same as Example 1. From Fig. 2, we can see that the BER performances obtained by using RLL-LDPC concatenation scheme are inferior to those obtained by using Algorithm SI with either $L = 16$ or $L = 256$. □

B. Systems Without Side Information

From Examples 1 and 2, we find that using the selective flipping technique combined with error-protected side information and LDPC coding for the RLL constrained PR channel has excellent performance at the cost of rate loss. Now we propose

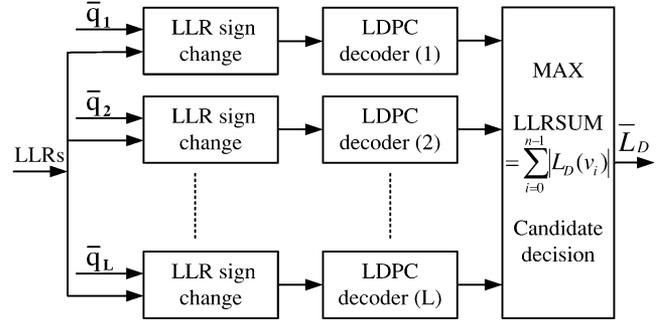


Fig. 3. Candidate decision for the selected flipping technique without side information (NSI).

to modify the selective flipping technique shown in Fig. 1 by deleting the error-protected side information. Hence, the information rate will be exactly the same as the ideal case. For this side-information free technique, \bar{z}_s is no longer available. We now only have $\bar{z} = \bar{y}' * \bar{h} + \bar{w}$ for detection. Due to the lack of side information, we have no idea of which candidate \bar{q} is selected. Thus, we have to test every candidate. As shown in Fig. 3, we have a set of L component detectors, each for a candidate $\bar{q}_l, 1 \leq l \leq L$. The detection and decoding algorithm for the side-information free system is shown as follows. Let I' be a positive integer which is smaller than I .

Algorithm NSI

- Step 1) For the l th component detector, $1 \leq l \leq L$, the iteration (Steps 1–4) for $U_o \in \{1, 2, \dots, I'\}$ is the same as that in Algorithm SI except that \bar{q} should be replaced by $\bar{q}_l = (q_{l,0}, q_{l,1}, \dots, q_{l,n-1})$.
- Step 2) At the end of the I' th iteration, we compute the sum of LLR values

$$LLRSUM = \sum_{i=0}^{n-1} |L_D(v_i)| \tag{2}$$

- for each component detector. Decide the component detector with the largest LLRSUM.
- Step 3) Let the l' -th component detector be the one with the largest LLRSUM. We complete the $(I' + 1)$ -th to the I th iteration for the l' th component detector.
- Step 4) If the index of iteration $U_o = I$, then the hard decision of $L_D(v_i), i = 0, 1, \dots, n - 1$, for the l' th component code is used as the final estimate of \bar{v} .

This side-information free selective flipping technique is designed based on an assumption that the cosets, $\bar{q}_i \oplus C_{ldpc}$ and $\bar{q}_j \oplus C_{ldpc}$, are widely separated, where $i \neq j$ and $\bar{q}_i \oplus C_{ldpc} = \{\bar{q}_i \oplus \bar{v} | \bar{v} \in C_{ldpc}\}$. If the assumption is correct, the bit error rate (BER) of the side-information free selective flipping technique will be similar to the technique using correct side information. Although we are unable to prove this fact analytically, in our simulation, we randomly pick some LDPC codes from [14] and the simulation results show that this technique works as expected.

The computational complexity of Algorithms SI and NSI is dominated by MAP equalization (detection) and LDPC decoding. The computational complexity of Algorithm NSI is about $(L'I' + (I - I'))/I$ times of that of Algorithm SI. Usually, using I' equal to 1 or 2 will be enough for the detector to identify the correct candidate. Hence, the increase of computational complexity will not be too great. For Algorithms SI and NSI implemented in hardware, the hardware complexity is highly dependent on the architectures used and the throughput and latency requirements. As compared to Algorithm SI, Algorithm NSI will increase the latency of the detector and decoder based on the same hardware resource.

Example 3: We modify Example 1 by removing the side information part and using Algorithm NSI for detection and decoding. Now we use the (999, 888) LDPC code from [14]. To reduce the computational complexity, we consider only the case of $L = 16$. In the simulation, we use $I = 15$ and $I' = 1$. The computational complexity of waiving side information is $(16 + 14)/15 = 2$ times of that of using the side information. Note that the BER curve of using side information is very close to the curve of not using side information as shown in Fig. 2, in this case of $L = 16$.

To achieve the BER performance approaching that of the ideal case, we may divide the 256 candidates into 16 groups and each group contains 16 candidates, where we use a 4-bit side information to indicate each group and the 4-bit side information is protected by a (24,4) code C_s . In the detection, the side information obtained by decoding C_s can help indicate the chosen group and the selected candidate inside the chosen group can be identified by Algorithm NSI. In Fig. 2, we see that the associated BER performance indicated by SI×NSI, $L = 16 \times 16$ is close to the ideal case. \square

In Example 3, if we want to completely remove the need of side information and have BER performance close to the ideal case, we need to increase L to a very large number, which will drastically increase the computational complexity. To avoid such significant increase in computational complexity, we can resort to the following technique.

III. ESTIMATION OF THE FLIPPED BITS

We now propose a scheme which alleviates the interference of flipped bits by estimating the positions of these flipped bits in the detection process. In this paper, we apply a selected flipping operation with a small L to this estimation scheme so that the number of flipped bits is not too great to estimate.

A. Flipped Bits Estimation for Side-Information Free Techniques

Remember that in Section II-B, for the side-information free technique, I' iterations are implemented for each of the L component detectors and in case that the l' th component detector has the largest LLRSUM at the I' th iteration then the rest of the $I - I'$ iterations are implemented for only the l' th component detector, where each iteration is implemented for the exchange of information between the MAP (channel) detector (equalizer) and the LDPC decoder as indicated in Section II-A. Note that the LDPC decoder is used to remove both the additive noise and the flipped bits which are used to meet the RLL constraint. In

case that the number of flipped bits is great, the LDPC decoder alone can not correct these flipped bits. We need to employ the correlation property within the RLL sequences to enhance the error-correcting capability of the LDPC code.

After we decide the single component detector which employs the candidate $\bar{q}_{l'}$ and will be chosen for the j th iteration, $I' < j \leq I$, we will execute the decoding and estimation operations as shown in Fig. 4. In each iteration, we have the soft output of the LDPC decoder, i.e., the *a posteriori* LLR value for the bit v_i , denoted $L_D(v_i)$, $0 \leq i \leq n - 1$. For each component of $\bar{q}_{l'}$ equal to 1, we reverse the polarity of corresponding component of the $L_D(v_i)$ to remove the effect of selective flipping and obtain $L_D(y_i)$. Make hard decision on $L_D(y_i)$. Then, we have $\bar{v}^* \oplus \bar{q}_{l'}$ which imitates the procedure of selective flipping, where $\bar{v}^* = (v_0^*, v_1^*, \dots, v_{n-1}^*)$ is an estimation of \bar{v} obtained by making hard decision on $L_D(v_i)$. Moreover, we modify $\bar{v}^* \oplus \bar{q}_{l'}$ into \bar{y}^* which satisfies the RLL constraint by flipping some bits of $\bar{v}^* \oplus \bar{q}_{l'}$. The nonzero bits in $\bar{v}^* \oplus \bar{q}_{l'} \oplus \bar{y}^*$ represent the estimation of the flipped bits for satisfying the RLL constraint. The nonzero bits in $\bar{v}^* \oplus \bar{q}_{l'} \oplus \bar{y}^*$ will be used to change the polarity of $L_a(y_i)$ and $L_e(y_i)$ in Algorithm SI. The algorithm for detection and decoding along with the estimation of flipped bits is summarized as follows.

ALGORITHM NSIEST

- Step 1) We follow Algorithm NSI until the we identify the l' th component detector as the one which has the largest LLRSUM at the end of the I' th iteration. We now consider only the l' th component detector. Set $U_o = I' + 1$.
- Step 2) For each component of $\bar{q}_{l'}$ equals to 1, we reverse the polarity of corresponding component of the $L_D(v_i)$ to obtain $L_D(y_i)$. Make hard decision on $L_D(y_i)$. Then, we have $\bar{v}^* \oplus \bar{q}_{l'}$. Modify $\bar{v}^* \oplus \bar{q}_{l'}$ into \bar{y}^* which satisfies the RLL constraint by flipping some bits of $\bar{v}^* \oplus \bar{q}_{l'}$.
- Step 3) Change the polarity of $L_a(y_i)$ if the i th component of $\bar{v}^* \oplus \bar{q}_{l'} \oplus \bar{y}^*$ is 1. We have $L'_a(y_i)$. Then, repeat Step 1 of Algorithm SI except that $L_a(y_i)$ is replaced by $L'_a(y_i)$.
- Step 4) Change the polarity of $L_e(y_i)$ if the i th component of $\bar{v}^* \oplus \bar{q}_{l'} \oplus \bar{y}^*$ is 1. We have $L'_e(y_i)$. Then, repeat Steps 2 and 3 of Algorithm SI except that $L_e(y_i)$ is replaced by $L'_e(y_i)$ and \bar{q} is replaced by $\bar{q}_{l'}$.
- Step 5) If $U_o < I$, then increase U_o by 1 and go back to Step 2. Otherwise, make hard decision on $L_D(v_i)$, $i = 0, 1, \dots, n - 1$, to obtain the final estimate of \bar{v} .

The computational complexity of Algorithm NSIEST is about $(L'I' + (I - I'))/I$ times of that of Algorithm SI. If using the same I' , the computational complexity of Algorithm NSIEST is almost the same as that of Algorithm NSI since the additional operations of finding $\bar{v}^* \oplus \bar{q}_{l'} \oplus \bar{y}^*$ and bit flipping contribute to insignificant computational complexity increase. However, the latency will be increased.

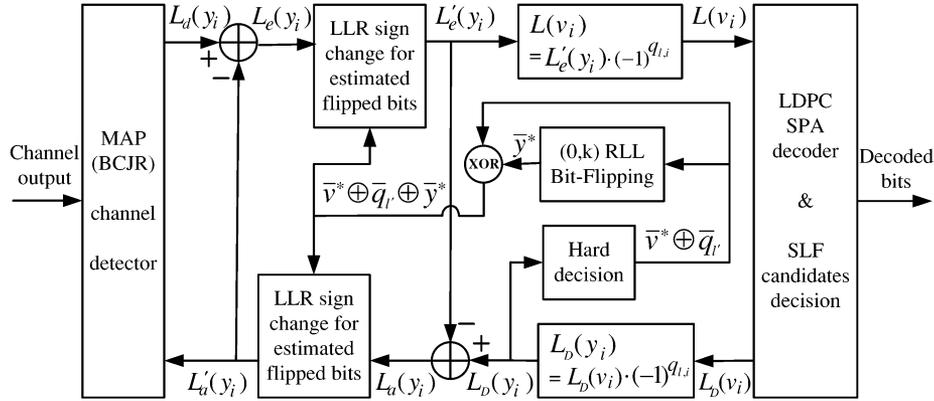


Fig. 4. Iterative decoding and flipped bits estimation for algorithm NSIEST.

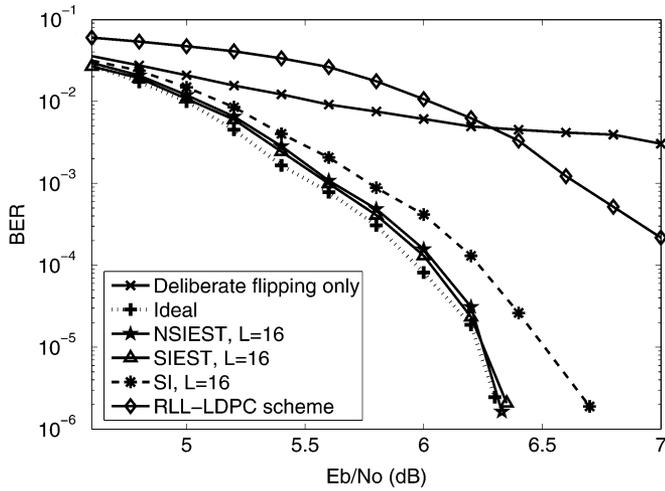


Fig. 5. Additional simulation results of (999, 888) LDPC coded EPR4 channel with (0,7) constraint ($U_i = 1, U_o = 15$).

Example 4: We modify Example 3 by using Algorithm NSIEST for detection and decoding instead of using Algorithm NSI. Now we use the (999, 888) LDPC code from [14]. Other parameters are $(d, k) = (0, 7), L = 16, I = 15$ and $I' = 1$. The computational complexity of using Algorithm NSIEST is only $(16 \times 1 + (15 - 1)) / 15 = 2$ times of that of using Algorithm SI. From Fig. 5, we see that the simulation results indicate that the BER performance of using Algorithm NSIEST is very close to the ideal case. □

B. Flipped Bits Estimation Along With Side Information

Either Algorithm NSI or Algorithm NSIEST is designed based on the idea of waiving the need of side information so that there is no rate loss resulting from using side information. The price is the increased computational complexity. The concept of estimation can be applied to the selective flipping technique with side information to reduce the number of candidates needed for selective flipping, i.e., L , and hence the number of bits for side information. In this way, there will be almost no penalty for computational complexity increase, while the number of needed side-information bits can be kept small.

Assuming that the side information \bar{s} is correctly recovered, we can then find the selected candidate \bar{q} . We have the following algorithm for iterations based on the estimation of flipped bits.

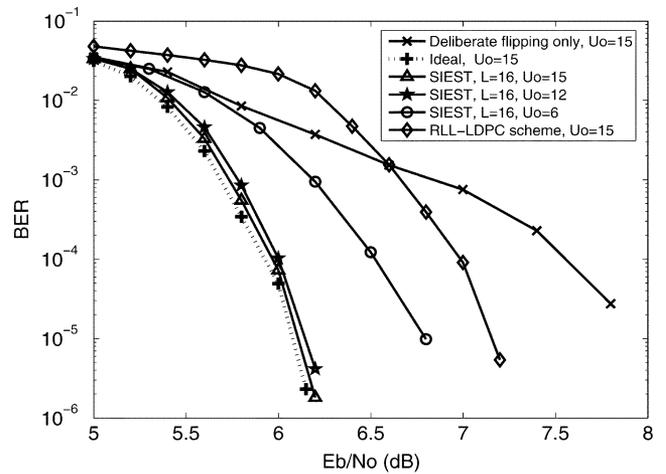


Fig. 6. Simulation results of (4608, 4096) LDPC coded EPR4 channel with (0,7) constraint ($U_i = 1$).

ALGORITHM SIEST

- Step 1) Set $U_o = 1$.
- Step 2) Repeat Step 2 to Step 5 of Algorithm NSIEST by using \bar{q} to replace $\bar{q}_{l,i}$.

Example 5: We modify Example 1 by using Algorithm SIEST to replace Algorithm SI. We consider the case of $L = 16$. From Fig. 5, we see that the obtained BER performance is close to the ideal case. From Fig. 2, we find that the BER performance of the ideal case is similar to that obtained by using Algorithm SI with $L = 16$. In case of using identical L , the computational complexity of Algorithm SIEST is almost the same as that of Algorithm SI. □

In the previous examples, we consider the (999, 888) LDPC code constructed by using the method in [14]. In the following example, we demonstrate Algorithm SIEST can work well for a longer LDPC code constructed by using the progressive edge growth (PEG) method in [20], [21].

Example 6: In this example, we construct an LDPC coded system using SLF to meet the (0,7) constraint by replacing the (999, 888) LDPC code with a (4608, 4096) PEG-LDPC code [20], [21]. Fig. 6 shows the BER performance of such a LDPC

coded system using Algorithm SIEST with $L = 16$ over an EPR4 channel. Also included in Fig. 6 are the BER performances for the ideal case and the case of using deliberate flipping proposed in [16]. From Fig. 6, we see that the simulation results indicate that the BER performance using Algorithm SIEST is very close to the ideal case. For comparison, we also construct a RLL-LDPC concatenation scheme [5] by using a (4352, 4096) RLL code and a (4608, 4352) PEG-LDPC code in a way similar to Example 2. The overall information rate of this RLL-LDPC concatenation scheme is $4096/4608 = 0.889$ which is the same as that of the (4608, 4096) PEG-LDPC code. From Fig. 6, we can see that the BER performance obtained by using RLL-LDPC concatenation scheme is inferior to that obtained by using Algorithm SIEST with $L = 16$. \square

C. Convergence Analysis for Algorithms NSIEST and SIEST

In this paper, we use the technique of EXIT (extrinsic information transfer) chart [22] to examine the convergence behavior of Algorithms NSIEST and SIEST. We assume that the detector is chosen correctly in either NSIEST and SIEST. We consider information exchange between the LDPC decoder and the MAP equalizer (detector) including the flipped bits estimator.

For the MAP equalizer including the flipped bits estimator, we investigate the EXIT characteristic T_E of the input mutual information $I_{E,in} = I_E(L_D(v_i), v_i)$ and the output mutual information $I_{E,out} = I_E(L(v_i), v_i)$. Since the channel output is also an input to the MAP equalizer, we calculate T_E based on a channel with a certain level of additive noise and a certain level of flipping errors which are dependent on E_b/N_0 , k and L , i.e., the number of candidates in the SLF technique. Hence, T_E is a function of E_b/N_0 , k and L , and we write $I_{E,out} = T_E(I_{E,in}, E_b/N_0, k, L)$.

For the LDPC decoder, we investigate the EXIT characteristic T_D of the input mutual information $I_{D,in} = I_D(L(v_i), v_i)$ and the output mutual information $I_{D,out} = I_D(L_D(v_i), v_i) = T_D(I_{D,in})$. Note that T_D is not a function of E_b/N_0 or k or L .

Fig. 7 shows T_D and T_E under various conditions. From Fig. 7, we find that the LDPC coded system using Algorithm NSIEST (or SIEST) with $L = 16$ can converge to a low BER at $E_b/N_0 = 6.5$ dB under (0, 7) constraint since T_D and T_E intersect at a value of mutual information equal to 1. In addition, we find that the LDPC coded system using Algorithm NSIEST (or SIEST) with $L = 256$ can not converge to a low BER at $E_b/N_0 = 6.5$ dB under (0, 3) constraint since T_D and T_E intersect at a low value of mutual information.

Although EXIT chart analysis is most effective for long codes, the EXIT chart for our examples using LDPC codes of lengths 999 and 4608 respectively still reveals valuable information. For example, we can investigate the speed of convergence by observing the trajectory in Fig. 8. We find that the LDPC coded system using Algorithm NSIEST (or SIEST) can converge to a value of mutual information equal to 1 at the 12th iteration. This observation of convergence speed is verified by the BER results shown in Fig. 6. Both the (4608, 4096) PEG LDPC code and the (999, 888) LDPC code used in this paper have the same rate of 0.888. The longer (4608, 4096) PEG LDPC code has better BER performance than the

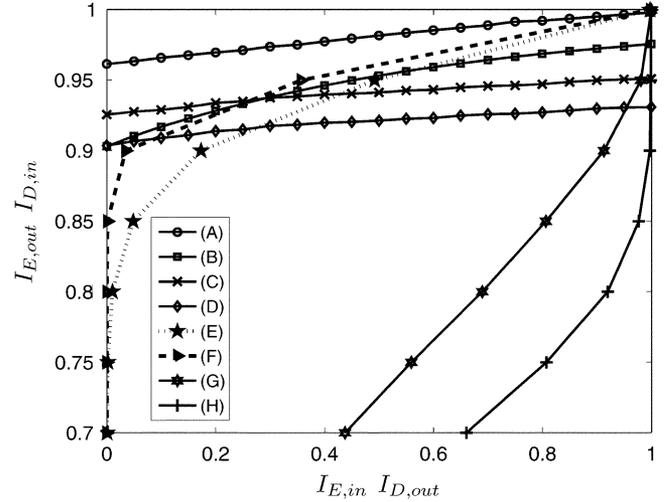


Fig. 7. EXIT curves of MAP detector including flipped bits estimator and LDPC decoder. (A) Detector, $(E_b)/(N_0) = 6.5$ dB, (0,7) constraint, $L = 16$; (B) Detector, $(E_b)/(N_0) = 5.0$ dB, (0,7) constraint, $L = 16$; (C) Detector, $(E_b)/(N_0) = 6.5$ dB, (0,3) constraint, $L = 256$; (D) Detector, $(E_b)/(N_0) = 5.0$ dB, (0,3) constraint, $L = 256$; (E) (999, 888) LDPC Decoder, $U_i = 1$; (F) (4608, 4096) LDPC Decoder, $U_i = 1$; (G) (4096, 2048) LDPC Decoder, $U_i = 1$; (H) (4096, 2048) LDPC Decoder, $U_i = 2$.

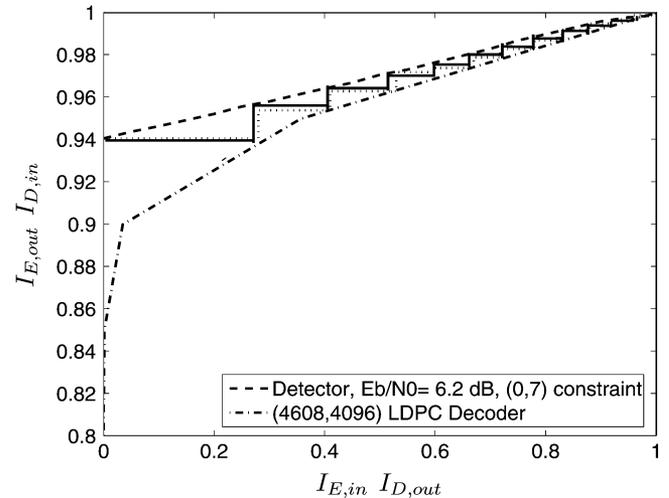


Fig. 8. Trajectories of iterative processes for MAP detector including flipped bits estimator and LDPC decoder (SIEST with $L = 16$ and $U_i = 1$).

(999, 888) LDPC code for E_b/N_0 greater than 6 dB. However, at $E_b/N_0 = 5$ dB under the (0, 7) constraint, the BER performance of the (4608, 4096) PEG LDPC code is worse. This result can be verified by observing Fig. 7, which shows that using the (4608, 4096) PEG LDPC code, the T_D and T_E intersect at a smaller value of mutual information as compared to using the (999, 888) LDPC code.

IV. FLIPPED BITS ESTIMATION FOR STRICT RLL CONSTRAINTS

For a very strict RLL constraint, such as $(d, k) = (0, 3)$, the number of flipped bits will be great even if the number of candidates in the SLF technique is very large. To correct such a large number of flipped bits, the rate of LDPC code should be low enough.

Example 7: Consider an LDPC coded system under the (0,3) constraint. We use SIEST with $L = 256$ and $U_o = 15$. Let

$E_b/N_0 = 6.5$ dB. Suppose that a rate 0.5 (4096, 2048) LDPC code [14] is used. Two T_D curves plotted for $U_i = 1$ and $U_i = 2$ can respectively be found in Fig. 7. For $U_i = 1$, it is clear that T_D and T_E intersect at a value of mutual information smaller than 1. If we increase the number of inner iterations of the LDPC code to $U_i = 2$, T_D and T_E intersect at a value of mutual information equal to 1. Simulation shows that for the $U_i = 1$ and $U_i = 2$ cases, the associated BER are about 10^{-2} and about 10^{-5} respectively. \square

If we want to avoid using the low rate LDPC code and using SLF with a large number of candidates, the powerful method proposed in [5] which has been described at the end of Section II-A can be used.

Example 8: Consider the case of $(d, k) = (0, 3)$. According to Table I of [5], we can design a (9,8) code C_3 meeting the (0,3) constraint, for which 1 of the 9 bit positions in each codeword can be used as the unconstrained bit. We concatenate 111 codewords of C_3 to form a (999, 888) RLL code C_4 , meeting the (0,3) constraint, for which 111 bit positions in each codeword can be used as unconstrained bits and the other $(999 - 111) = 888$ bit positions can be used as constrained bits. In a way similar to Example 2, we can have a LDPC codeword of length 999 bits which meets the (0,3) constraint and the overall information rate is $777/999 \approx 0.778$. \square

To increase the information rate above the method proposed in [5], we can encode only a portion of the message bits into a RLL sequence. After the RLL and LDPC encoding, there are some code bits which may not meet the RLL constraint and can be processed by the selective flipping technique and the flipped bits estimation technique. An example which improves the information rate of Example 8 is given as follows.

Example 9: Consider the case of $(d, k) = (0, 3)$. According to Table I of [5], we can design a (10,9) code C_5 meeting the (0,3) constraint, for which 1 of the 10 bit positions in each codeword can be used as the unconstrained bit. We concatenate 98 codewords of C_5 to form a (980, 882) RLL code C_6 , meeting the (0,3) constraint, for which 98 bit positions in each codeword can be used as unconstrained bits and the other $(980 - 98) = 882$ bit positions are used as constrained bits. At the beginning of encoding, we encode an 882-bit sequence which contains 784 message bits and 98 zero bits into a 980-bit codeword of C_6 . The 98 zero bits are located at the unconstrained bit positions. The other 882 constrained bits of a codeword of C_6 together with additional 6 message bits are encoded into a codeword of the (999, 888) systematic LDPC codeword, where 98 out of the 111 parity bits will replace 98 zero bits in the unconstrained bit positions. The resultant code, denoted C_7 , has an overall information rate of $790/999 \approx 0.790$. In C_7 , each codeword is still an LDPC codeword of length 999 bits for which 980 bits meets the (0,3) constraint and the other 19 bits may not. We can deliberately flip some bits of these 19 bits to meet the (0,3) constraint. At the reading side, we can apply the conventional message passing between LDPC code and the MAP detector, which is equivalent to Algorithm SI with $L = 1$. The associated BER performances as indicated by “ C_7 , deliberate flipping only” are shown in Fig. 9. We can also apply NSIEST to C_7 with deliberate flipping. From Fig. 9, we can see that the BER performances represented by “ C_7 , NSIEST and $L = 1$ ”, i.e., pure estimation without using SLF technique, are clearly superior to those represented by “ C_7 , deliberate flipping only.” In Fig. 9,

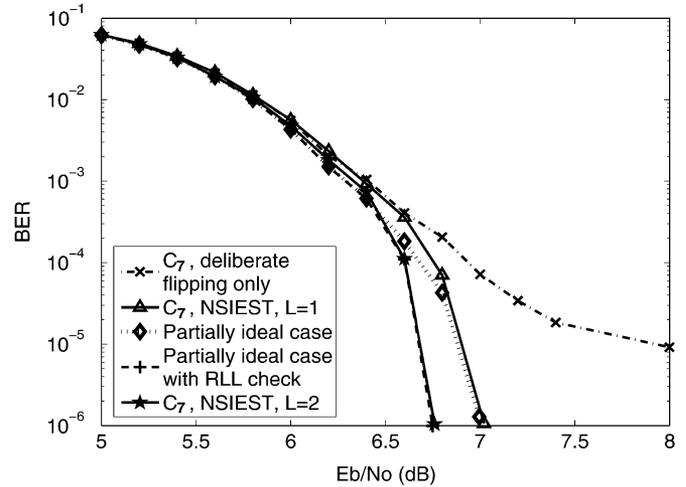


Fig. 9. Simulation results of (999, 888) LDPC coded EPR4 channel with (0,3) constraint ($U_i = 1, U_o = 15$).

we also observe that C_7 with NSIEST and $L = 2$ provides further improvement in the BER performances. \square

As a comparison, we may consider a partially ideal case for which in the storage side, C_7 without flipping any bit is used, and in the reading side, conventional message passing between LDPC code and the MAP detector, equivalently Algorithm SI with $L = 1$, is used. Although no bit flipping is needed in this partially ideal case, we see from Fig. 9 that the associated BER performances are inferior to those obtained by C_7 with NSIEST and $L = 2$. A major reason for this phenomenon is that in C_7 with NSIEST and $L = 2$, the correlation property imposed by the (0,3) constraint is efficiently exploited to remove errors caused by additive noise and deliberate flipping. To examine this argument, we modify the detection of the partially ideal case by checking the 980 bit positions which are encoded to satisfy the (0,3) constraint to see whether they meet the (0,3) constraint or not. Fig. 9 shows that the associated BER performances will be very close to those obtained by C_7 with NSIEST and $L = 2$.

BER performances of the ideal case, for which LDPC coded PR channel without RLL constraint is considered, have been shown in Figs. 2 and 5. For the ideal case, the BER is 10^{-6} at E_b/N_0 of 6.3 dB. From Fig. 9, we see that C_7 with NSIEST and $L = 2$, the BER is 10^{-6} at E_b/N_0 of 6.75 dB. The loss of E_b/N_0 for C_7 with NSIEST and $L = 2$ as compared to the ideal case at BER = 10^{-6} is 0.45 dB, which is resultant from the rate loss. Since the information rate of C_7 is 0.790 and the information rate of the ideal case is 0.889, the asymptotic rate loss at very high E_b/N_0 will approach 0.51 dB. In this sense, NSIEST with a sufficiently large L is very efficient for ECC with strict RLL constraint by properly using the technique of [5].

V. CONCLUSION

We propose two techniques for the LDPC coded partial response channel with RLL constraint. The first is a modification of the selective flipping technique so that side information is not needed. The advantage of this modification is the increased information rate and the disadvantage is the increased computational complexity as compared to the selective flipping technique using side information. The second is based on the esti-

mation of flipped bits for the selective flipping technique. This technique can increase the information rate or reduce the BER without the penalty of increasing computational complexity as compared to the selective flipping technique either using side information or not using side information. In case of very strict RLL constraint, the proposed techniques can be incorporated into a technique in [5] to increase the information rate without performance degradation.

ACKNOWLEDGMENT

The authors are grateful to the anonymous reviewers who provided valuable comments and suggestions which significantly enhance the quality of this paper. This work was supported by National Science Council of the R.O.C. under Grant NSC 96-2219-E-002-012.

REFERENCES

- [1] B. H. Marcus, P. H. Siegel, and J. K. Wolf, "Finite-state modulation codes for data storage," *IEEE J. Sel. Areas Commun.*, vol. 10, no. 1, pp. 5–37, Jan. 1992.
- [2] K. A. S. Immink, *Coding Techniques for Digital Recorders*. Englewood Cliffs, NJ: Prentice Hall, 1991.
- [3] P. Lee and J. K. Wolf, "A general error-correcting code construction for run-length limited binary channels," *IEEE Trans. Inf. Theory*, vol. 35, no. 6, pp. 1330–1335, Nov. 1989.
- [4] H. H. Tang and M. C. Lin, "A class of multilevel run-length limited trellis codes," *Electron. Lett.*, vol. 35, no. 25, pp. 2192–2193, Dec. 9, 1999.
- [5] A. J. van Wijngaarden and K. A. S. Immink, "Maximum runlength limited codes with error control capabilities," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 4, pp. 602–611, Apr. 2001.
- [6] Y. Han and W. E. Ryan, "Concatenating a structured LDPC code and a constrained code to preserve soft-decoding, structure, and burst correction," *IEEE Trans. Magn.*, vol. 42, no. 10, pp. 2558–2560, Oct. 2006.
- [7] J. Lu and K. G. Boyer, "Novel RLL-ECC concatenation scheme for high-density magnetic recording," *IEEE Trans. Magn.*, vol. 43, no. 6, pp. 2271–2273, Jun. 2007.
- [8] B. M. Kurkoski, P. H. Siegel, and J. K. Wolf, "Joint message-passing decoding of LDPC codes and partial-response channels," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1410–1422, Jun. 2002.
- [9] J. L. Fan and J. M. Cioffi, "Constrained coding techniques for soft iterative decoders," in *Proc. IEEE GLOBECOM*, Rio De Janeiro, Brazil, 1999, pp. 723–727.
- [10] Z. Li and B. V. K. V. Kumar, "Low-density parity-check codes with run length limited (RLL) constraints," *IEEE Trans. Magn.*, vol. 42, no. 2, pp. 344–349, Feb. 2006.
- [11] K. Anim-Appiah and S. W. McLaughlin, "Turbo codes cascaded with high-rate block codes for $(0, k)$ -constraint channels," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 4, pp. 677–685, Apr. 2001.
- [12] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error correcting coding and decoding: Turbo codes," in *Proc. IEEE Int. Conf. Commun. (ICC 93)*, Geneva, Switzerland, May 1993, pp. 1064–1070.
- [13] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.
- [14] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1997.
- [15] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: A rediscovery and new results," *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2711–2736, Nov. 2001.
- [16] B. Vasic and K. Pedagani, "Run-length-limited low-density parity check codes based on deliberate error insertion," *IEEE Trans. Magn.*, vol. 40, no. 3, pp. 1738–1743, May 2004.
- [17] A. Kunisa, "Runlength violation of weakly constrained code," *IEEE Trans. Commun.*, vol. 50, no. 1, pp. 1–6, Jan. 2002.
- [18] I. J. Fair, W. D. Gover, W. A. Krzymien, and R. I. MacDonald, "Guided scrambling: A new line coding technique for high bit rate fiber optic transmission systems," *IEEE Trans. Commun.*, vol. 39, pp. 289–297, Feb. 1991.
- [19] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes, Appendix A*. Amsterdam, The Netherlands: North-Holland, 1983.
- [20] X. Hu, E. Eleftheriou, and D. Arnold, "Progressive edge-growth Tanner graphs," in *Proc. IEEE Global Telecommunications Conf. 2001 (GLOBECOM'01)*, Nov. 25–29, 2001, vol. 2, pp. 995V–1001.
- [21] X.-Yu. Hu, "Regular and irregular progressive edge-growth Tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.
- [22] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 40, pp. 1727–1737, Oct. 2001.

Manuscript received November 20, 2007; revised April 23, 2008. Published August 20, 2008 (projected). Corresponding author: M. C. Lin (e-mail: mcln@cc.ee.ntu.edu.tw).

Hsin-Yi Chen was born in Chiayi County, Taiwan, R.O.C., on November 6, 1969. He is currently working toward the Ph.D. degree in graduate institute of communication engineering, National Taiwan University, Taipei, Taiwan, R.O.C.

Since 1998, he has served as an Assistant Researcher in Telecommunication Laboratories of Chinghua Telecom Co., Ltd., Taipei, Taiwan, Republic of China. His research interests are in the area of coding theory and constrained coding for recording systems.

Mao-Chao Lin was born in Taipei, Taiwan, R.O.C., on December 24, 1954. He received the Bachelor's and Master's degree, both in electrical engineering, from National Taiwan University in 1977 and 1979, respectively, and the Ph.D. degree in electrical engineering from the University of Hawaii in 1986.

From 1979 to 1982, he was an Assistant Scientist of Chung-Shan Institute of Science and Technology at Lung-Tan, Taiwan. He is currently a Professor in the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C. His research interests are in the area of coding theory and its applications.

Yeong-Luh Ueng (M'05) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1997. At the same university, he received the M.S. and Ph.D. degrees in communication engineering from the Graduate Institute of Communication Engineering in 1999 and 2001, respectively.

From 2001 to 2005, he was with a private company in Taiwan and focused on the design and development of various wireless chips including RF transceiver chips and Bluetooth and PHS baseband chips. In December 2005, he joined the faculty of National Tsing-Hua University, Hsinchu, Taiwan, where he is currently an Assistant Professor in the Department of Electrical Engineering and the Institute of Communications Engineering. His research interests include coding theory, wireless communications, and communication IC.

Dr. Ueng was elected an honorary member of the Phi Tau Phi Scholastic Honor Society.