On Soft Iterative Decoding For Ternary Recording Systems With RLL Constraints

Shih-Kai Lee
Department of Communications Engineering
Yuan Ze Univ.
Taoyuan, Taiwan, R.O.C.
Email: sklee@saturn.yzu.edu.tw

Hsin-Yi Chen*
Telecommunication Laboratories of
Chunghwa Telecom Co., Ltd.
Taipei, Taiwan, R.O.C.
Email: yichen@cht.com.tw

Mao-Chao Lin, Tien-Hui Chen
Hong-Fu Chou, Yu-Hsien Ku
Institute of Communications Engineering
National Taiwan Univ.
Taipei, Taiwan, R.O.C.
Email: mclin@cc.ee.ntu.edu.tw

***Abstract*—In this paper, we investigate the soft iterative decoding technique for ternary recoding systems with run-length-limited (RLL) constraints. We employ a simple binary-to-ternary RLL encoder following the LDPC (low density parity check) encoder. In the decoder, the iteratively passing of soft information between the LDPC decoder and a detector is used, where the detector is constructed for a combination of the RLL encoder, PLM (pulse length modulation) precoder and the partial response channel. We provide two different decoding algorithms. For one of the decoding algorithm, we are able to obtain bit-error-rate performance which is inferior to the comparable system without considering the RLL constraint for the high sign-to-noise ratio (SNR) regime and is better for the low-to-moderate SNR regime.**

I. Introduction

In recording systems, run-length-limited (RLL) coding [1]-[3] is frequently used to alleviate the undesired inter-symbol interference (ISI) and to facilitate synchronization. An RLL sequence is a sequence with the run length of consecutive 0's between two consecutive nonzero symbols being under a given constraint. In particular, a (d, k) sequence [4]-[6] is an RLL sequence for which the number of consecutive 0's between two consecutive nonzero symbols is at least d and at most k.

By now, there have been many research works related to binary RLL coding [7]-[14]. In this paper, we investigate the ternary RLL coding assisted with LDPC coding.

Recently, three-level RLL coded recording experiments have been reported [15], [16]. The use of ternary-RLL coding can increase the recording density by 50% as compared to the binary RLL recording [15]. In this paper, we focus on ternary $(0, k)$ RLL constrained recording systems. We consider the recording system comprising LDPC coding, interleaving, RLL coding, precoding using pulse length modulation (PLM), partial response (PR) channel corrupted by additive white Gaussian noise (AWGN), maximum a posteriori (MAP) decoding for the precoded PR channel, soft mapping/demapping, interleaver/deinterleaver, and channel decoding.

Since error correction coding (ECC) is a powerful tool for enhancing the reliability, the integration of ECC and RLL in the recording system is an important issue. Usually using RLL coding followed by an ECC, referred as the RLL-ECC concatenation, will destroy the RLL constraint. In [11], The RLL-ECC concatenation is implemented by a more strict RLL coding for the message part of the ECC and the parity part of the ECC is uniformly inserted back to the message part to result in the RLL sequence with less strict constraint. Using ECC followed by the RLL coding, referred as the ECC-RLL concatenation, will usually affect either the error-correcting capability of the ECC or complicate the decoding operation. In [12], a modified ECC-RLL version is proposed by flipping the bits in ECC codeword to satisfy the RLL constraint and the flipped bits are removed by the error-correcting capability of the ECC code. In [13], a multiple-candidate technique is used to reduce the number of flipped bits.

In this paper, we consider the ECC-RLL concatenation and employ the iterative passing of the soft information between the ECC decoder and the RLL detector. The soft iterative decoding for the binary case has been discussed in [17]. It will be interesting to investigate the performance for the ternary recording system using the soft iterative decoding. In this paper, we propose two soft iterative decoding algorithms for the ECC-RLL concatenation recording systems. The difference of the two proposed algorithms lies in the trellis representations of the concatenation of the RLL encoder, PLM (pulse length modulation) precoder, the partial response channel. We will compare the decoding performance of the proposed decoding algorithms for the ECC-RLL systems to an efficient RLL-ECC algorithm using the techniquein [11].

The rest of this paper is organized as follows. In section II, we describe the basic LDPC coded recording systems. The proposed soft decoding algorithms are given in section III. Simulation results and comparison are shown in section IV. Conclusions are provided in section V.

II. Basic LDPC Coded Recording Systems

A. PLM Precoded Partial Response Channels

In the recording, a PLM precoder [18] followed a signal mapper is used. An M-ary PLM precoder takes an M-ary symbol sequence $\{x_i\}$ as input and an M-ary symbol sequence

[*] Corresponding author: Hsin-Yi Chen is with the Telecommunication Laboratories of Chunghwa Telecom Co., Ltd. 5F, No. 11, Lane 74, Sec. 4, Hsin-Yi Rd., Taipei, Taiwan 10682, R.O.C. (e-mail: yichen@cht.com.tw).

This work was supported by National Science Council of R.O.C. under grant NSC 98-2221-E-155-022-.

978–1–4244–6017–5/10/$26.00 © 2010 IEEE

$\{y_i\}$ as output, where the relation between $\{x_i\}$ and $\{y_i\}$ the mapping is given by

$$y_i = x_i \oplus y_{i-1} \quad (mod\ M). \tag{1}$$

The signal mapper maps the M-ary symbol into an M-level pulse amplitude modulation (PAM) signal so that the ternary symbol in $\{0, 1, 2\}$ is mapped to a symbol in $\{-1, 0, 1\}$. The channel between the PLM precoder and the reading output of the recording system can be represented by a partial response (PR) channel. In this paper, We use the PR channel represented by $1+D+D^2+D^3$, denoted as PR(1,1,1,1), for demonstration. For simplicity, we assume that the output of the PR channel will be corrupted by additive white Gaussian noise (AWGN) only, although in the optical recording channel, noise model should be more complicated.

Fig. 1 shows the equivalent form of the concatenation of the PLM precoder, signal mapper and the partial response channel, which can be represented by an M^D-state PSPR (PLM-Signal mapper-Partial response channel) trellis. The channel detector at the reading side can apply the Viterbi algorithm or the Max-Log BCJR algorithm to PSPR trellis to retrieve the hard or soft information respectively. Since we consider the PR(1,1,1,1) partial response in the paper, the number of states in the PSPR trellis will be 3^3 = 27.

Suppose that we place an LDPC encoder and an interleaver in front of the PLM precoder. We can iteratively passing the soft information between the BCJR detector of the PSPR trellis and the LDPC decoder to achieve very reliable decoding results. Usually, we will conduct U_i inner iterations within the LDPC decoder and U_o outer iterations within the loop comprising the LDPC decoder and the BCJR detector of the PSPR trellis.

In case that RLL constraint is considered and the ECC-RLL concatenation is used, we will place an RLL encoder between the interleaver and the PLM precoder as shown in Fig. 2. In the following, we will show a simple RLL encoder.

B. Ternary (0,3) Code Construction

The capacity of an M-ary RLL (d, k) coding [19], denoted by $C_{(M,d,k)}$ bits/symbol, is given by

$$C_{(M,d,k)} = \log_2 \lambda, \tag{2}$$

where λ is the largest real root of the characteristic equation

$$Z^{k+2} - Z^{k+1} - (M-1)Z^{k-d+1} + M - 1 = 0. \tag{3}$$

The code efficiency of an M-ary RLL (d, k) code is expressed as $\eta = R/C_{(M,d,k)}$. According to (2) and (3), the ternary $(0, 3)$ RLL coding has capacity $C_{(3,0,3)} \approx 1.5726$ bits/symbol.

In practical applications, we need to encode every p message bits to an RLL codeword of q ternary symbols to form a ternary RLL block code with code rate of $R = p/q$ $bits/symbols$, where the concatenation of any two q-symbol sequences should still meet the RLL constraints. In Tables I, we show a ternary $(d, k) = (0, 3)$ RLL code with p = 4 and q =3. The associated code rate is R = p/q = 4/3 bits/symbol, the code efficiency is $\eta = 84.7852\%$. Using a larger p, we can construct ternary (0,3) code with better code efficiency. However, the complexity of decoding will be increased.

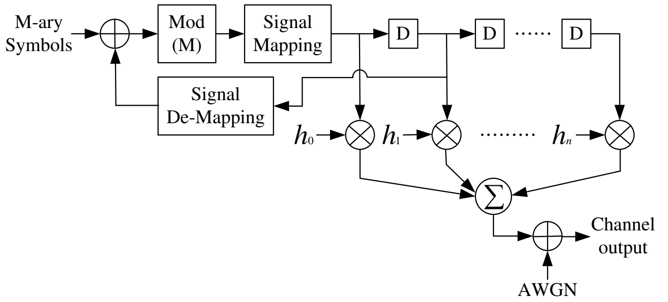

Fig. 1. Equivalent form of the concatenation of the pulse length modulation precoder, the signal mapper and the partial response channel.

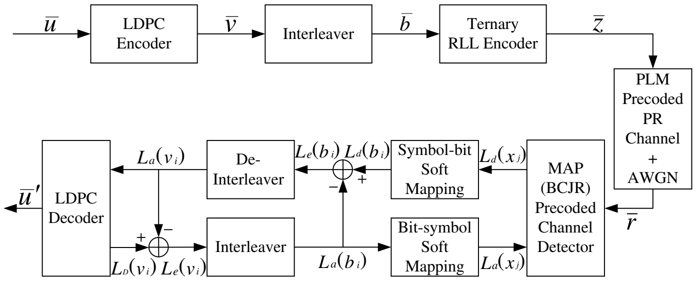

Fig. 2. Schematic diagram of a ternary RLL recording systems over the precoded partial response channel with iterative decoding.

III. Soft Decoding for an ECC-RLL Scheme

The LDPC coded RLL recording system shown in Fig. 2 can be decoded efficiently by passing the soft information between the RLL-PSPR detector and the LDPC decoder, where the RLL-PSPR detector is the detector for the concatenation of the RLL encoder, the PLM decoder, the signal mapper and the partial response channel. However, the trellis used for the RLL-PSPR detector,i.e, the trellis representing the concatenation of the RLL encoder, the PLM decoder, the signal mapper and the partial response channel is essential in the decoding. In the following, we propose two different RLL-PSPR detectors, which respectively provide different performances.

A. Codeword-Branch Trellis

Now we consider the trellis with codeword branches, which represents the the concatenation of the RLL encoder, the PLM decoder, the signal mapper and the partial response channel. Such a trellis can also be referred as the codeword-PSPR trellis. There are $\rho = M^D$ states, L = 2^p outgoing branches for each state, where D is the number delay of precoded PR channel, L is number of possible RLL codewords. However, for different states, the number of incoming branches may be different. Let $s^{'}$ be the present state and s be the next state. The branch connecting state $s^{'}$ and state s is represented by a q-symbol RLL codeword (corresponding to the p-bit input $[b_0, b_1, \cdots, b_{p-1}]$), denoted as $\overline{X}_\ell$, ℓ = 0, 1, $\cdots$, $L-1$ and the

TABLE I

A TERNARY $(0,3)$ RLL CODE WITH RATE OF 4/3 BITS/SYMBOL.

Binary	Ternary
$b_3b_2b_1b_0$	$x_2x_1x_0$
0 0 0 0	0 0 1
0 0 0 1	0 0 2
0 0 1 0	0 1 0
0 0 1 1	0 2 0
0 1 0 0	0 1 1
0 1 0 1	0 1 2
0 1 1 0	0 2 1
0 1 1 1	0 2 2
1 0 0 0	1 0 1
1 0 0 1	1 0 2
1 0 1 0	1 1 0
1 0 1 1	1 2 0
1 1 0 0	2 0 1
1 1 0 1	2 0 2
1 1 1 0	2 1 0
1 1 1 1	2 2 0

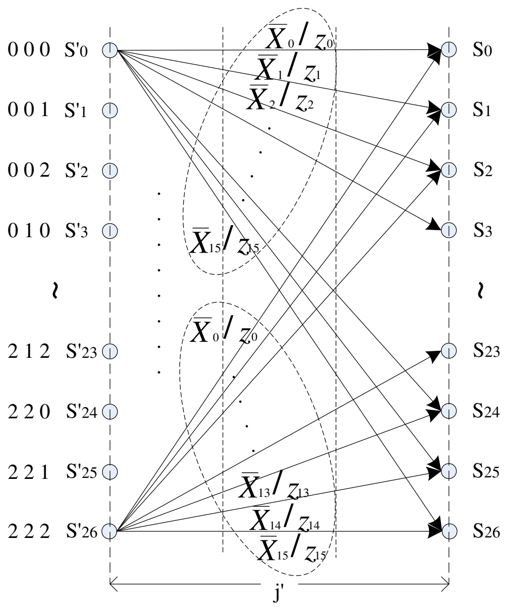

Fig. 3. Codeword-PSPR trellis.

associated output of the PR channel, denoted as $\overline{z}_\ell$. The RLL-PSPR trellis for M = 3 and D = 3 is shown in Fig. 3. Based on $\overline{r}_\ell$ = $(r_{\ell,0}, \cdots, r_{\ell,q-1})$, which is the error-corrupted version of $\overline{z}_\ell$, the operation of iterative passing of soft information between the LDPC decoder and the trellis MAP detector is described as following.

ALGORITHM Using the Codeword-PSPR Trellis:

Set the index of outer iteration U_o = 1 and the $a\ priori\ LLR$ (log-likelihood ratio) value of the RLL codeword $\overline{X}_\ell$, $L_a(\overline{X}_\ell)$ to $\ln(\frac{1}{L})$.

Step 1 The MAP detector for the RLL-PSPR trellis takes $\overline{r}_\ell$ and $L_a(\overline{X}_\ell)$, as input to calculate the metric of the q-symbol RLL codeword $\overline{X}_\ell$ representing the branch from state $s^{'}$ to state s, $\gamma_{t,\ell}(s^{'},s)$. The $\gamma_{t,\ell}(s^{'},s)$ is calculated by

$$\gamma_{t,\ell}(s^{'},s) = L_a(\overline{X}_\ell) + \sum_{i=0}^{q-1} \ln(e^{-\frac{E_b}{N_0}|r_{\ell,i}-z_{\ell,i}|^2}) \quad (4)$$

where $t \in \{0, 1, \cdots, B\}$ is the index of time and E_b is the average bit energy and N_0 is the one-sided power spectral density of the AWGN. Note that an n-bit binary LDPC codeword is encoded into a sequence of B M-ary RLL codewords. Let max$^*(y,z)$ = max(y,z) + $\ln(1+e^{-|y-z|})$. We calculate α_t in the forward recursion and β_t in the backward recursion, where

$$\alpha_t(s) = \begin{cases} 0, & \text{if } t = 0,\ s^{'} = 0; \\ -\infty, & \text{if } t = 0,\ s^{'} \neq 0; \\ \max^*[\gamma_{t,\ell}(s^{'},s) & \\ +\alpha_{t-1}(s^{'})], & \text{if } 1 \leq t \leq B. \end{cases} \quad (5)$$

and

$$\beta_t(s^{'}) = \begin{cases} \ln\frac{1}{\rho}, & \text{if } t = \text{B}; \\ \max^*[\gamma_{j^{'},t}(s,s^{'}) & \\ +\beta_{j^{'}+1}(s)], & \text{if } 0 \leq t \leq B-1. \end{cases} \quad (6)$$

The $a\ posteriori$ LLR of $\overline{X}_\ell$ from $s^{'}$ to s is $L_d(\overline{X}_\ell)$ = $\max^*_{\overline{X}_\ell}[\gamma_{t,\ell}(s^{'},s)$ + $\alpha_{t-1}(s)$ + $\beta_{t+1}(s^{'})]$. The $a\ posteriori\ LLR$ of b_i is $L_d(b_i)$ = $\max^*_{\overline{X}_\ell \in (b_i=1)} L_d(\overline{X}_\ell)$ - $\max^*_{\overline{X}_t \in (b_i=0)} L_d(\overline{X}_\ell)$, i = 0, 1, $\cdots$, $p-1$. In each iteration, the extrinsic LLR of b_i is obtained by $L_e(b_i)$ = $L_d(b_i)$ - $L_a(b_i)$.

Step 2 After the de-interleaver processing, we have $L_a(v_i)$ = $\prod^{-1}(L_e(b_i))$. Then, use $(L_a(v_0), L_a(v_1), \cdots, L_a(v_{n-1}))$ as input to the decoder of the LDPC code C_{ldpc} with code length n = Bq/p. 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. The extrinsic value for the bit v_i from the LDPC decoder is obtained by $L_e(v_i)$=$L_D(v_i)$ - $L_a(v_i)$. After interleaver processing, $L_a(b_i)$ = $\prod(L_e(v_i))$, the resultant sequence is $(L_a(b_0), L_a(b_1), \cdots, L_a(b_{n-1}))$ which will be used in the next iteration.

Step 3 The $a\ priori$ bit LLR $L_a(b_i)$ must be converted into the LLR of the RLL codeword $\bar{X}_\ell$, $L_a(\bar{X}_\ell)$ as input of MAP detector. Let $Pr(b_i = 0)$=$\frac{1}{1+e^{L_a(b_i)}}$ and $Pr(b_i = 1)$=$\frac{e^{L_a(b_i)}}{1+e^{L_a(b_i)}}$. Then, the probability of RLL codeword $\overline{X}_\ell$ is, $Pr(\overline{X}_\ell)$ = $Pr(b_{\ell,0}) \cdot Pr(b_{\ell,1}) \cdot \ldots \cdot Pr(b_{\ell,p-1})$, where $\ell \in \{0, 1, \ldots, L-1\}$. Finally, the $L_a(\overline{X}_\ell) = \ln(Pr(\overline{X}_\ell))$.

Step 4 Increase U_o by 1. If the index of iteration U_o = I, then the hard decision of $L_D(v_i)$, $i = 0, 1, \cdots, n-1$, is used as the final estimate of $\bar{v}$, the codeword of C_{ldpc}. Otherwise, go back to Step 1.

B. Symbol-Branch Trellis

Now we consider the trellis with symbol branches, which represents the concatenation of the PLM decoder, the signal mapper and the partial response channel. Note that the RLL encoder is not included. Such a trellis can be referred as the symbol-PSPR trellis. There are $\rho = M^D$ states, M outgoing branches and M incoming branches for each state. Let $s^{'}$ be the present state and s be the next state. The branch connecting state $s^{'}$ and state s is represented by a an M-ary symbol (corresponding to M-ary input), denoted as X, X = 0, 1, $\cdots$, $M-1$ and the associated output of the PR channel, denoted as z. The PSPR trellis for M = 3 and D = 3 is shown in Fig. 4. Based on r, the error-corrupted version of X, the operation of iterative passing of soft information between the LDPC decoder and the MAP detector for the RLL-PSPR concatenation is described as following.

ALGORITHM Using the Symbol-PSPR Trellis :

Set the index of outer iteration U_o = 1 and the *a priori* LLR value of the RLL symbol x_j, $L_a(x_j)$ to $\ln(\frac{1}{M})$.

Step 1 The MAP detector for the PSPR trellis takes r_j and $L_a(x_j)$, as input to calculate the symbol branch metrics, $\gamma_j(s^{'}, s)$,

$$\gamma_j(s^{'}, s) = L_a(x_j) + \ln(e^{-\frac{E_b}{N_0}|r_j - z_j|^2}), \qquad (7)$$

where $j \in \{0,\ 1,\ \ldots,\ H)$ is the time index. Note that an n-bit binary LDPC codeword is encoded into a sequence of H M-ary symbols. The $L_a(x_j)$ is the *a prior* symbol LLR. For the forward recursion,

$$\alpha_j(s) = \begin{cases} 0, & \text{if } j = 0,\ s^{'} = 0; \\ -\infty, & \text{if } j = 0,\ s^{'} \neq 0; \\ \max{}^{*}[\gamma_j(s^{'}, s) & \\ +\alpha_{j-1}(s^{'})], & \text{if } 1 \leq j \leq H. \end{cases} \qquad (8)$$

For backward recursions,

$$\beta_j(s^{'}) = \begin{cases} \ln\frac{1}{\rho}, & \text{if } j = H; \\ \max{}^{*}[\gamma_j(s, s^{'}) & \\ +\beta_{j+1}(s)], & \text{if } 0 \leq j \leq H-1. \end{cases} \qquad (9)$$

The *a posteriori* value of x_j is, $L_d(x_j, g)$ = $\max^{*}_{x_j \in g}[\gamma_j(s^{'}, s) + \alpha_{j-1}(s) + \beta_{j+1}(s^{'})]$, $j \in (0, 1, \ldots, H-1)$, $g \in \{0, 1, \ldots, M-1\}$.

Step 2 The soft-value of the symbol, $L_d(x_j, g)$, need to be converted into bit Log-Likehood Ratio (LLR). According to the p-bit to q-symbol RLL encoding table, the bit LLR for each b_i is, $L_d(b_i) = L_d(b_i = 1) - L_d(b_i = 0)$, $i = 0, 1, \ldots, p-1$, where $L_d(b_i = 1)$=$\max^{*}_{\overline{X}_\ell \in (b_i=1)}[\sum_{m=j^{'}q}^{(j^{'}+1)q-1} L_d(x_m, g)]$, $L_d(b_i = 0)$=$\max^{*}_{\overline{X}_\ell \in (b_i=0)}[\sum_{m=j^{'}q}^{(j^{'}+1)q-1} L_d(x_m, g)]$, $j^{'} \in (0, 1, \ldots, B-1)$, $g \in \{0, 1, \ldots,$M-1$\}$.

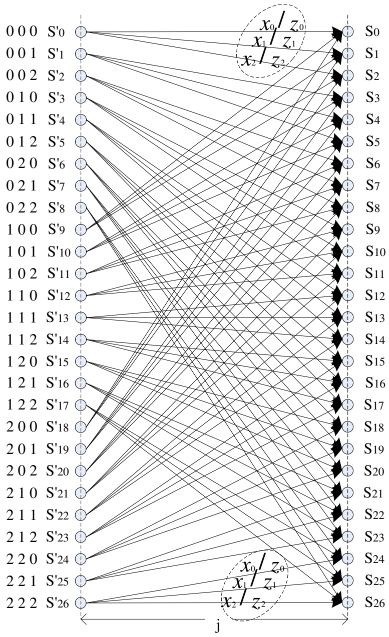

Fig. 4. Symbol-PSPR trellis.

Step 3 After the de-interleaver processing, we have $L_a(v_i)$ = $\prod^{-1}(L_e(b_i))$. Then, use $(L_a(v_0), L_a(v_1),$ $\cdots, L_a(v_{n-1}))$ as input to the decoder of the LDPC code C_{ldpc}. After U_i iterations within the LDPC decoder, the LDPC decoder generates the *a posteriori LLR* for v_i, denoted $L_D(v_i)$ as output. The extrinsic value for the bit v_i from the LDPC decoder is obtained by $L_e(v_i)$=$L_D(v_i)$-$L_a(v_i)$. After the interleaver processing, we have $L_a(b_i)$ = $\prod(L_a(v_i))$. The resultant sequence is $(L_a(b_0), L_a(b_1), \cdots, L_a(b_{n-1}))$, which will be used in the next iteration.

Step 4 The bit-to-symbol soft mapper convertes the bit LLR to *a prior* bit probability, by $Pr(b_i$ = $0)$=$\frac{1}{1+e^{L_a(b_i)}}$ and $Pr(b_i$ = $1)$=$\frac{e^{L_a(b_i)}}{1+e^{L_a(b_i)}}$. The binary p-bit codeword probability of $\bar{X}_\ell$ is, $Pr(\overline{X}_\ell)$ = $Pr(b_{\ell,0}) \cdot Pr(b_{\ell,1}) \cdot \ldots \cdot Pr(b_{\ell,p-1})$, where $\ell \in \{0, 1, \ldots, L-1\}$. The symbol probability of x_j is, $Pr(x_{j,i})$=$\sum_{\overline{X}_\ell \in (x_j = g)}\{Pr(\overline{X}_\ell)\}$, where $\ell \in \{0, 1, .., L-1\}$, g = $\{0, 1, \cdots, M-1\}$. Then, the *a prior* symbol soft value is $L_a(x_j) = \ln(Pr(x_j))$.

Step 5 Increase U_o by 1. If the index of iteration U_o = I, then the hard decision of $L_D(v_i)$, $i = 0, 1, \cdots, n-1$, is used as the final estimate of $\bar{v}$. Otherwise, go back to Step 1.

IV. Specific Constructions and Simulation Results

In this section, we show specific LDPC coded ternary RLL recording systems decoded based on the codeword-PSPR trellis and symbol-PSPR trellis respectively together with the simulation results for bit error rates (BER). Here, we consider

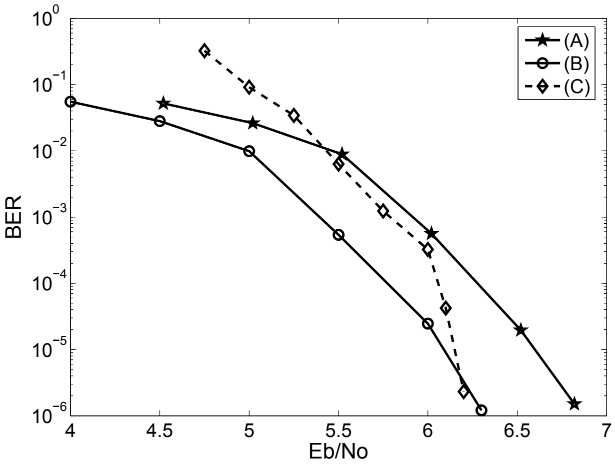

Fig. 5. Simulation results of a ternary coded systems over the PR(1,1,1,1) channel. $U_i = 1$, $U_o = 20$. (A) Using the codeword-PSPR trellis; (B) Using the symbol-PSPR trellis; (C) RLL-free system.

the PR (1,1,1,1) channel. The RLL constraint is the $(d, k) = (0, 3)$ constraint. The RLL encoder is obtained based on Tables I. Thus, the coding rate o the RLL code is $R_{rll} = p/q$ = 4/3 bits/symbol. We follow the Progressive edge-growth (PEG) LDPC construction method [20] to find a (4608,4096) binary LDPC code with coding rate of R_{ecc} = 0.8889. Hence, the overall coding rate is $R = R_{ecc}R_{rll}$ = (4/3)·0.8889 = 1.1852 bits/symbol. One inner iteration within the LDPC decoder and 20 iterations between the MAP detector and the LDPC decoder are employed. Thus, U_i = 1 and U_o = 20. From Fig. 5, we see that using the symbol-PSPR trellis will outperform using the codeword-PSPR trellis. This phenomenon probably implies that extracting the LLR bit information from the RLL structure alone is more accurate than extracting the LLR bit information from the trellis which integrates the RLL coding and the PSPR concatenation.

For comparison, we also show the BER performance of the LDPC coded ternary system over the concatenation of the PLM precoder, signal mapper and the PR response channel of the same parameters without RLL constraint. We use a rate of 11/7 bits/symbol signal mapper to convert the binary sequence into the ternary sequence. Although there is no RLL constraint in the signal mapper, the rate of 11/7 $\approx$ 1.5714 is still a bit lower than the capacity of the ternary (0,3) coding which is 1.5726. In order that the overall system rate to be close to 1.1852 bits/symbol, a (3456,2608) ternary PEG LDPC code is used. From Fig. 5, we note that compared to the ternary system without RLL constraint, the proposed ternary ECC-RLL concatenation using the PSPR trellis can achieve BER performance which is better for the low to medium signal-to-noise ratio (SNR) regime and is worse for the high SNR regime.

V. Conclusion

We have investigated the LDPC coded ternary recording systems with (0,3) constraint through a ECC-RLL concatenation structure. Two decoding methods for the iterative passing of soft information between the LDPC decoder and the MAP detector for the concatenation of the combination of the RLL encoder, PLM precoder and the partial response channel are proposed. The first is based on the trellis with codeword branches representing the the combination of the RLL encoder, PLM precoder and the partial response channel. The second is based on the trellis with symbol branches representing the combination of PLM precoder and the partial response channel. Simulation results shows that the second provides better performance. Comparison with the system without considering the RLL constraint is also provided.

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] K. A. S. Immink, *Codes for Mass Data Storage Systems.* Shannon Foundation Publishers, 2004.

[4] C. French, G. Dixon, and J. Wolf, ”Results involving d,k-constrained M-ary codes,” *IEEE Trans. Magn.,* MAG-23, No. 5, pp. 3678-80, 1987.

[5] D.T. Tang and L.R. Bahl, ”Block Codes for a Class of Constrained Noiseless Channels,” Information and Control, vol. 17, pp. 436-461, 1970.

[6] H.-Y. Chen, H.-F. Chou, M.-C. Lin, and S.-K. Lee, ”Capacity Approaching Run-Length-Limited Codes for Multilevel Recording Systems,” *IEEE Trans. Magn.,* vol. 46, no. 1, pp. 95V104, Jan. 2010.

[7] 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.

[8] A. Kunisa, ”Runlength violation of weakly cnstrained code,” *IEEE Trans. Commun.,* vol. 50, no. 1, pp. 1-6, Jan. 2002.

[9] I. J. Fair,W. D. Gover, W. A. Krzymien, and R. I. MacDonald, ”Guided scrambling: A new line coding tecnique for high bit rate fiber optic transmission systems,” *IEEE Trans. Commun.,* vol. 39, pp. 289-297, Feb. 1991.

[10] 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.

[11] 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.

[12] 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. 1738V1743, May 2004.

[13] 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.

[14] Hsin-Yi Chen, Mao-Chao Lin, and Yeong-Luh Ueng, ”Low-Density Parity-Check Coded Recording Systems With Run-Length-Limited Constraints,” *IEEE Trans. Magn.,* vol. 44, no. 9, pp. 2235-2242, Sept. 2008.

[15] F. H. Lo, J. W. Kuo, N. H. Tseng, J. J. Ju, and D. Howe, ”Recording of multi-level run-length-limited modulation signals on compact disc/digital versatile disc rewritable discs,” Jpn. J. Appl. Phys., vol. 43, no. 7B, pp. 4852-4855, Jul. 2004.

[16] H. Hu, L. F. Pan, G. S. Qi, H. Hu and D. Y. Xu, ”Study of multi-level run-length limited photo-chromic storage,” Acta Physica Sinica, vol. 55, pp. 1759-1763, April 2006.

[17] John L. Fan, *Constrained Coding and Soft Iterative Decoding.* ISBN 079237455X, 9780792374558, Springer, 2001.

[18] Dennis, ”Rll modulation for multi-level recording,” Optical Data Storage Center,University of Arizona, 1999.

[19] CA. French, G.S. Dixon, and J. Wolf, ”Results involving (d,k)- constrained M-ary codes,” *IEEE Trans. Magn.,* vol. 23, no. 5, pp. 3678-3680, Sept. 1987.

[20] X. Hu, E. Eleftheriou, and D. Arnold, ”Progressive edge-growth Tanner graphs,” Global Telecommunications Conference, 2001. GLOBECOM ’01. IEEE , Volume: 2 ,pp. 995-1001, 25-29 Nov. 2001.