Transactions Letters

Combined Puncturing and Path Pruning for Convolutional Codes and the Application to Unequal Error Protection

Chung-Hsuan Wang and Yi-Hsin Lin

Abstract—**In this paper, puncturing and path pruning are combined for convolutional codes to construct a new coding scheme for unequal error protection (UEP), called the hybrid punctured and path-pruned convolutional codes. From an algebraic viewpoint, we show that the hybrid codes not only inherit all the advantages of the conventional rate-compatible punctured convolutional codes and path-compatible pruned convolutional codes but also can provide more flexible choices of protection capability for UEP. In addition, a data-multiplexing scheme originally proposed for path-pruned codes which can guarantee smooth transition between rates without additional zero-padding for frame termination is proven applicable to the hybrid codes to improve the system throughput.**

Index Terms—**Convolutional codes, path pruning, puncturing, unequal error protection.**

I. Introduction

CHANNEL coding for unequal error protection (UEP) has recently attracted significant research interests, since it can make the best use of the channel bandwidth for many practical applications, e.g., broadcast systems, packet-switching networks, and visual/speech communication systems, etc. Among convolutional codes, there exist two UEP schemes: the rate-compatible punctured convolutional (RCPC) codes [1] and the path-compatible pruned convolutional (PCPC) codes [2]. RCPC codes achieve UEP by puncturing off different amounts of coded bits of the parent code*. For PCPC codes, UEP is accomplished by pruning away different amounts of the low-weight paths from the trellis of the parent code.

Both approaches provide remarkable UEP performance but have many complementary properties in nature. For instance, compared with the parent code, RCPC codes are of higher rates and poorer bit error rate (BER) performance but PCPC codes are of lower rates and better BER performance. RCPC codes can easily provide flexible choices of code rate but suffer from large number of the long error events which implies the decoder will require a longer path memory [1]. PCPC codes, on the other hand, take the advantages of sparse distance spectra and shorter error events, but only few of rates are available if the (n, k) parent codes of small k are chosen [2]. By comparing the code tables searched in [1]–[4], PCPC codes are observed to have better UEP capabilities than RCPC codes for some choices of rate and memory. Yet in other cases, RCPC codes can have better UEP performance than PCPC codes. In this paper, puncturing and path pruning are combined for convolutional codes to construct a new UEP scheme, called the hybrid punctured and path-pruned convolutional codes, which can not only gain the advantages of both schemes but also mitigate the respective drawbacks.

Paper approved by A. K. Khandani, the Editor for Coding and Information Theory of the IEEE Communications Society. Manuscript received October 2, 2006; revised April 13, 2007. This work was supported by the National Science Council of the Republic of China under Grant NSC 96–2219–E–009–011 and Aim for the Top University Plan. It was presented in part at the IEEE International Symposium on Information Theory, Yokohama, Japan, July 2003.

The authors are with the Department of Communication Engineering, National Chiao Tung University, Hsinchu 30010, Taiwan (e-mail: chwang@mail.nctu.edu.tw).

Digital Object Identifier 10.1109/TCOMM.2008.060393.

*For convenience, the ordinary convolutional code to be punctured or pruned is called the parent code, and the resulting RCPC codes or PCPC codes are called the children codes.

II. Combined Puncturing and Path Pruning

To consider both of the effects of puncturing and path pruning, we first derive an algebraic formulation of the hybrid codes†. Recall the puncturing and pruning processes in [1] and [2], respectively. Consider an (n, k) parent code C with generator matrix $G(D)$. A puncturing table $\mathcal{A}$ with period p is defined as an $n \times p$ matrix with the (u, v)th entry $a_{u,v}$ taking value from $\{0, 1\}$ $\forall$ $0 \le u < n$, $0 \le v < p$. Puncturing C by $\mathcal{A}$ can generate the punctured code of the following equivalent generator matrix:

$$G(D)^{[p]} \cdot \mathcal{P} \tag{1}$$

where $G(D)^{[p]}$ denotes the equivalent $kp \times np$ generator matrix of C by depth-p blocking [5] and $\mathcal{P}$ is the puncturing matrix corresponding to $\mathcal{A}$ obtained from the $pn \times pn$ identity matrix by deleting its $(j+1)$th column $\forall$ $j \in \{n*v+u | \forall$ $a_{u,v} = 0\}$. For pruning with $\hat{k}$ active inputs, a path locating matrix $\Theta(D)$ is defined as a $\hat{k} \times k$ polynomial matrix with $1 \le \hat{k} < k$. Pruning C by cascading $\Theta(D)$ ahead of $G(D)$ can obtain the path-pruned code with generator matrix

$$\Theta(D) \cdot G(D). \tag{2}$$

Suppose C is first pruned by $\Theta(D)$ and then punctured by $\mathcal{A}$. By (1) and (2), the resulting hybrid code is now of the following generator matrix:

$$(\Theta(D) \cdot G(D))^{[p]} \cdot \mathcal{P}. \tag{3}$$

†In the rest of this paper, the hybrid punctured and path-pruned convolutional codes are called the hybrid codes for convenience.

To further manipulate (3), consider Lemma 1. (Please see Appendix for the detailed proof.)

Lemma 1: Consider an $(n,\ k)$ convolutional code with generator $G(D)$ which is pruned by the path locating matrix $\Theta(D)$ to obtain the child code $\hat{C}$. The generator matrix of the equivalent $(np,\ kp)$ code constructed by blocking $\hat{C}$ with depth-p is of the form

$$\Theta(D)^{[p]} \cdot G(D)^{[p]}$$

where $\Theta(D)^{[p]}$ and $G(D)^{[p]}$ denote the depth-p blocked versions of $\Theta(D)$ and $G(D)$, respectively.

By Lemma 1, it implies that $(\Theta(D) \cdot G(D))^{[p]} = \Theta(D)^{[p]} \cdot G(D)^{[p]}$, and hence the generator matrix in (3) can be deduced to

$$\Theta(D)^{[p]} \cdot (G(D)^{[p]} \cdot \mathcal{P}). \tag{4}$$

However, if C is first punctured by $\mathcal{A}$ and then pruned by $\Theta(D)^{\ddagger}$, by (1) and (2), the generator matrix of the resulting hybrid code is just of the form in (4). Therefore, we have the following important result on combining puncturing and path pruning.

Theorem 1: Given a parent code for combined puncturing and path pruning, the same hybrid code will be obtained no matter whether we first do path pruning or puncturing.

Example 1: Given a convolutional code C with generator matrix

$$G(D) = \begin{bmatrix} D & 1+D & 1+D \\ D^2+D+1 & D & 0 \end{bmatrix}$$

let C be first pruned by a path locating matrix $\Theta(D) = [D^2+1\ \ 1]$ and then punctured by the following puncturing table $\mathcal{A}$ with period 2:

$$\mathcal{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \text{ and } \mathcal{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where $\mathcal{P}$ is obtained from the 6×6 identity matrix by deleting its second column since $a_{1,0}$ is the only zero-entry in $\mathcal{A}$. Blocking $\Theta(D) \cdot G(D)$ with depth-2, we have

$$(\Theta(D) \cdot G(D))^{[2]} = \begin{bmatrix} D+1 & D+1 & D+1 & D & D & D+1 \\ D^2 & D^2 & D^2+D & D+1 & D+1 & D+1 \end{bmatrix}.$$

Therefore, by (3), the resulting hybrid code is of generator matrix

$$(\Theta(D) \cdot G(D))^{[2]} \cdot \mathcal{P} = \begin{bmatrix} D+1 & D+1 & D & D & D+1 \\ D^2 & D^2+D & D+1 & D+1 & D+1 \end{bmatrix}. \tag{5}$$

On the other hand, suppose C is first punctured and then path-pruned by the same $\mathcal{A}$ and $\Theta(D)$. By depth-2 blocking, we have

$$\Theta(D)^{[2]} = \begin{bmatrix} D+1 & 1 & 0 & 0 \\ 0 & 0 & D+1 & 1 \end{bmatrix}$$

and

$$G(D)^{[2]} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ D+1 & 0 & 0 & 1 & 1 & 0 \\ D & D & D & 0 & 1 & 1 \\ D & D & 0 & D+1 & 0 & 0 \end{bmatrix}.$$

‡To match the dimension after depth-p blocking, $G(D)^{[p]} \cdot \mathcal{P}$ is cascaded with $\Theta(D)^{[p]}$, instead of $\Theta(D)$, in (4).

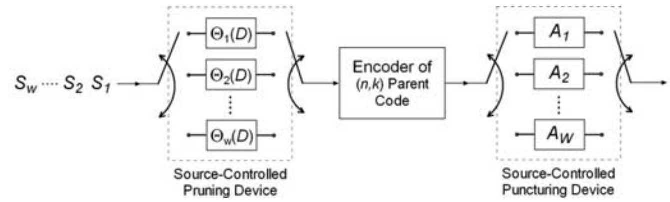

Fig. 1. Source-controlled scenario of combined path pruning and puncturing for UEP.

By (4), we have

$$\Theta(D)^{[2]} \cdot (G(D)^{[2]} \cdot \mathcal{P}) = \begin{bmatrix} D+1 & D+1 & D & D & D+1 \\ D^2 & D^2+D & D+1 & D+1 & D+1 \end{bmatrix}. \tag{6}$$

By comparing (5) and (6), the same hybrid code is obtained no matter whether we first do puncturing or path pruning. ■

III. Application of the Hybrid Codes to UEP

Given an $(n,\ k)$ parent code, puncturing generates only children codes of higher rates and worse protection capabilities by (1). Path pruning contrarily generates children codes of lower rates and better protection capabilities by (2). However, if the parent code is to be path-pruned and punctured (with period p) simultaneously by (3) or (4), we can obtain the hybrid codes of more available rates, i.e., $l_1 p/(l_1 p + l_2)$ $\forall\ 1 \le l_1 \le k$ and $1 \le l_2 \le (n - l_1)p$ with values ranging from $\frac{1}{n}$ to $\frac{kp}{kp+1}$, which can provide both of better and worse BER performances than the parent code. Morevoer, RCPC and PCPC codes can be decoded by a single Viterbi decoder for their parent codes as long as the add-compare-select units are modified to accommodate the puncturing effect for metric computation [1] and to identify the valid state-transition after path pruning [2], respectively. With both of the above modifications, the hybrid codes can be shown to preserve the desirable single-decoder property by Theorem 1. Owing to these advantages, the hybrid codes are well suitable for the application to UEP.

Consider W groups of source data S_l's, each of the required BER $P_{b,l}$; assume $P_{b,1} \ge P_{b,2} \ge \cdots \ge P_{b,W}$ without loss of generality. To provide UEP for S_l's by the hybrid scheme, we first choose a proper parent code together with W pairs of the puncturing tables and path locating matrices $(\mathcal{A}_l, \Theta_l(D))$'s to generate a family of children codes $\hat{C}_l$'s, each of free distance $d_f(\hat{C}_l)$ to satisfy $P_{b,l}$. $(A(l)$'s and $\Theta_l(D)$'s should be carefully selected to avoid catastrophic encoders for $\hat{C}_l$'s and to satisfy the single-decoder property, respectively.) Based on the source-controlled scenario in Fig. 1, $\mathcal{A}_l$ and $\Theta_l(D)$ are dynamically switched for puncturing and path pruning as S_l is fed to the encoder, thus fulfilling the desire for UEP. However, similar to the pure RCPC and PCPC schemes, the dynamic switching of $(\mathcal{A}_l, \Theta_l(D))$'s may cause unpredictable degradation of the UEP performance designed for hybrid codes. Conventionally, puncturing and path pruning are regulated by the rate-compatible criterion [1] and the path-compatible criterion [2], respectively, to guarantee the UEP performance. Let $\mathcal{P}_l$ denote the puncturing matrix associated with $\mathcal{A}_l$ for all l. In terms of $\mathcal{P}_l$'s and $\Theta_l(D)$'s, the rate-compatible criterion and path-compatible criterion can be

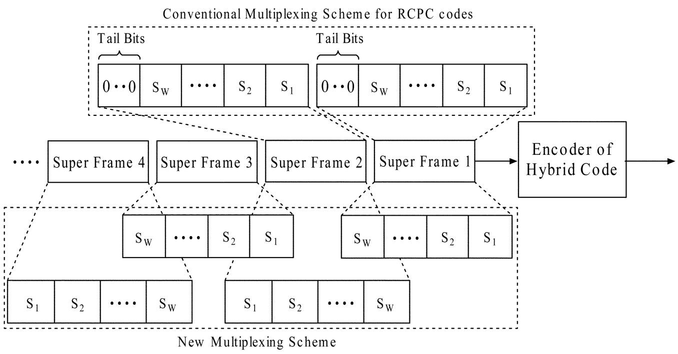

Fig. 2. Multiplexing schemes for UEP.

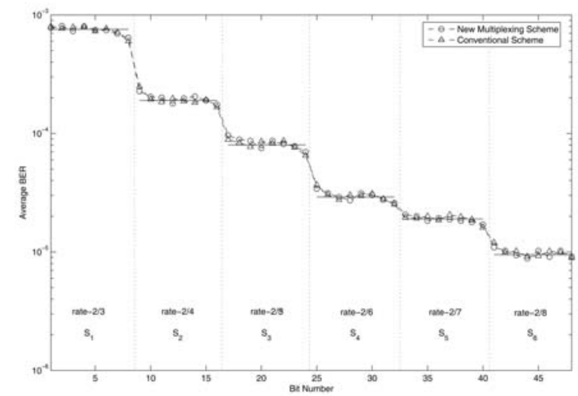

Fig. 3. Average BER of source bits in a super frame for different multiplexing schemes on additive white Gaussian noise channels at signal-to-noise ratio 4 dB with binary phase-shift keying modulation, where six groups of data S_i's $\forall\ 1 \le i \le 6$ (each containing 8 bits per super frame) are protected by the code family in Table I. (The solid lines denote the designed BER of the children codes.)

expressed as

$$\mathcal{P}_l \subseteq \mathcal{P}_{l+1} \text{ and } \Theta_{l+1}(D) = \theta_l(D) \cdot \Theta_l(D),\ \forall\ 1 \le l < W \quad (7)$$

where $\mathcal{P}_l \subseteq \mathcal{P}_{l+1}$ means that all columns of $\mathcal{P}_l$ are contained in $\mathcal{P}_{l+1}$ and $\theta_l(D)$'s are some polynomial matrices. Both of the criteria can then be directly applied to the hybrid codes to mitigate the undesired performance degradation, since puncturing and path pruning can be independently conducted by Theorem 1. Under the above restrictions, we have Theorem 2 to assure the UEP performance of hybrid codes.

Theorem 2: Consider a parent code C which is processed by W pairs of puncturing tables and path locating matrices $(\mathcal{A}_l, \Theta_l(D))$'s satisfying the rate-compatible criterion and the path-compatible criterion in (7) to generate a family of children codes $\hat{C}_l$'s, each of free distance $d_f(\hat{C}_l)$. Suppose we have $d_f(\hat{C}_l) \le d_f(\hat{C}_{l+1})$, $\forall\ 1 \le l < W$. Then, all codewords across the switching boundaries between ϕ pairs of $(\mathcal{A}_{l_1}, \Theta_{l_1}(D))$, $(\mathcal{A}_{l_2}, \Theta_{l_2}(D))$, $\cdots$, $(\mathcal{A}_{l_\phi}, \Theta_{l_\phi}(D))$ with $1 \le l_i \le W\ \forall\ 1 \le i \le \phi$ for combined puncturing and path pruning will have a distance $\min_{1\le i\le\phi} d_f(\hat{C}_{l_i})$ at least.

Moreover, in [1], S_l's are suggested to be grouped into super frames before encoding as the conventional multiplexing scheme depicted in Fig. 2. In this scheme, S_l is followed by $S_{l+1}\ \forall\ 1 \le l < W$ to achieve the minimum distance loss guaranteed by the rate-compatible criterion; extra (all-zero) tail bits are inserted at the end of every super frame to avoid the abrupt switching from S_W to S_1. However, by Theorem 2, the distance between any two codewords of the hybrid code can be shown to be lower bounded by $d_f(\hat{C}_l)$ no matter the puncturing tables and path locating matrices are switched either from $(\mathcal{A}_l, \Theta_l(D))$ to $(\mathcal{A}_{l+1}, \Theta_{l+1}(D))$ or from $(\mathcal{A}_{l+1}, \Theta_{l+1}(D))$ to $(\mathcal{A}_l, \Theta_l(D))$. Suppose S_l's are multiplexed by the new scheme in Fig. 2 which is originally proposed for pure PCPC codes [2], where no tail bits are required but S_l's are multiplexed in a reverse order for alternate super frames. Accordingly, the puncturing tables and path locating matrices are restricted to switch either from $(\mathcal{A}_l, \Theta_l(D))$ to $(\mathcal{A}_{l+1}, \Theta_{l+1}(D))$ or from $(\mathcal{A}_{l+1}, \Theta_{l+1}(D))$ to $(\mathcal{A}_l, \Theta_l(D))$, $\forall\ 1 \le l < W$; the same distance loss as the conventional multiplexing scheme can thus be obtained even without additional overheads.

For performance verification, both of the multiplexing schemes are simulated with the same family of hybrid codes in Table I. In this family, the parent code (of memory 5) with depth-2 blocking is properly punctured and path pruned to provide 6-level UEP under the rate-compatible and path-compatible criteria. Let every super frame contain 6 groups of source data, each consists of 8 bits. Extra 5 zero bits are required in the conventional scheme for every super frame. The BER curves in Fig. 3 indicate that the hybrid codes with dynamic switching of $(\mathcal{A}_l, \Theta_l(D))$'s can achieve the designed UEP performance as expected in Theorem 2. The new multiplexing scheme is also observed to provide almost the same UEP performance as the conventional one even without the extra overheads for frame termination.

TABLE I
THE FAMILY OF HYBRID CODES FOR SIMULATION IN FIG. 3 ($G(D)$: GENERATOR MATRIX OF PARENT CODE, r_c: CODE RATE OF CHILD CODE, $\Theta(D)$: PATH LOCATING MATRIX, $\mathcal{A}$: PUNCTURING TABLE)

$G(D) = (D^5+D^3+D+1\ \ D^5+D^4+D^3+D^2+1)$	
$\Theta(D) = (D^3+D^2+1\ \ D^2+1)$	
r_c	$\mathcal{A}$
$\frac{2}{8}$	$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}^{\mathcal{T}}$
$\frac{2}{7}$	$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}^{\mathcal{T}}$
$\frac{2}{6}$	$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}^{\mathcal{T}}$
$\frac{2}{5}$	$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}^{\mathcal{T}}$
$\frac{2}{4}$	$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}^{\mathcal{T}}$
$\frac{2}{3}$	$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}^{\mathcal{T}}$

$\mathcal{T}$ denotes the operation of taking transpose.

IV. CONCLUSIONS

In this paper, puncturing and path pruning are combined for convolutional codes to construct a new UEP scheme. Compared with the pure RCPC and PCPC codes, the hybrid codes not only inherit all the advantages of both schemes but also can mitigate the respective drawbacks and provide

$$\Phi_1(D)^{[p]} + \Phi_2(D)^{[p]} = \begin{pmatrix} \Phi_{1,0}+\Phi_{2,0} & \Phi_{1,1}+\Phi_{2,1} & \cdots & \Phi_{1,p-1}+\Phi_{2,p-1} \\ & \Phi_{1,0}+\Phi_{2,0} & \cdots & \Phi_{1,p-2}+\Phi_{2,p-2} \\ & & \ddots & \vdots \\ & & & \Phi_{1,0}+\Phi_{2,0} \end{pmatrix}$$
$$+ \sum_{l=1}^{\max(d_1,\,d_2)} D^l \begin{pmatrix} \Phi_{1,lp}+\Phi_{2,lp} & \Phi_{1,lp+1}+\Phi_{2,lp+1} & \cdots & \Phi_{1,(l+1)p-1}+\Phi_{2,(l+1)p-1} \\ \Phi_{1,lp-1}+\Phi_{2,lp-1} & \Phi_{1,lp}+\Phi_{2,lp} & \cdots & \Phi_{1,(l+1)p-2}+\Phi_{2,(l+1)p-2} \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{1,(l-1)p+1}+\Phi_{2,(l-1)p+1} & \Phi_{1,(l-1)p+2}+\Phi_{2,(l-1)p+2} & \cdots & \Phi_{1,lp}+\Phi_{2,lp} \end{pmatrix}$$

more flexible choices of UEP capability. From an algebraic viewpoint, we show that the hybrid codes can be constructed in spite of the order of puncturing and path pruning; it has also been proven that the conventional rate-compatible and path-compatible criteria can be directly applied to the hybrid codes to assure the designed performance for UEP. In addition, the efficient data-multiplexing scheme originally presented for PCPC codes is shown applicable to the hybrid codes to improve the system throughput.

Appendix Proof of Lemma 1

To prove Lemma 1, we first show the following properties for code blocking:

Property 1. $(\Phi_1(D)+\Phi_2(D))^{[p]} = \Phi_1(D)^{[p]} + \Phi_2(D)^{[p]}$

Property 2. $(\Psi_1 \cdot \Psi_2 D^{\eta_1+\eta_2})^{[p]} = (\Psi_1 D^{\eta_1})^{[p]} \cdot (\Psi_2 D^{\eta_2})^{[p]}$

where $\Phi_1(D)$, $\Phi_2(D)$ denote two arbitrary additive polynomial matrices, Ψ_1, Ψ_2 stand for two arbitrary multiplicative binary matrices, and η_1, η_2 are non-negative integers. Decompose $\Phi_1(D)$ and $\Phi_2(D)$ as $\Phi_i(D) = \sum_{l=0}^{d_i} \Phi_{i,l} D^l \ \forall\ i =$ 1 and 2, where $\Phi_{i,l}$'s are some binary matrices and d_i is the maximum degree of $\Phi_i(D)$. By code blocking, for i =1 and 2, we have

$$\Phi_i(D)^{[p]} = \begin{pmatrix} \Phi_{i,0} & \Phi_{i,1} & \cdots & \Phi_{i,p-1} \\ & \Phi_{i,0} & \cdots & \Phi_{i,p-2} \\ & & \ddots & \vdots \\ & & & \Phi_{i,0} \end{pmatrix} + \sum_{l=1}^{d_i} D^l \begin{pmatrix} \Phi_{i,lp} & \Phi_{i,lp+1} & \cdots & \Phi_{i,(l+1)p-1} \\ \Phi_{i,lp-1} & \Phi_{i,lp} & \cdots & \Phi_{i,(l+1)p-2} \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{i,(l-1)p+1} & \Phi_{i,(l-1)p+2} & \cdots & \Phi_{i,lp} \end{pmatrix} \quad \text{(A-1)}$$

where $\Phi_{i,l}$ turns to be a zero matrix if $l > d_i$ and all zero matrices in the blank area are neglected for convenience. By (A-1), for i = 1 and 2, it implies that the value of $\Phi_1(D)^{[p]} + \Phi_2(D)^{[p]}$ is as shown at the top of the page, which is the same as the direct decomposition of $(\Phi_1(D)+\Phi_2(D))^{[p]}$ by (A-1), hence completing the proof of Property 1.

Next, by (A-1), we have

$$(\Psi_i D^{\eta_i})^{[p]} = D^{\phi_i} \begin{pmatrix} \underbrace{0 \cdots 0}_{\eta_i \bmod p} & \Psi_i & & \\ & & \ddots & \\ & & & \Psi_i \\ & & \eta_i \bmod p & \left\{ \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \right. \end{pmatrix} + D^{\phi_i+1} \begin{pmatrix} \left. \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \right\} p-(\eta_i \bmod p) & & \\ \Psi_i & & \\ & \ddots & p-(\eta_i \bmod p) \\ & & \Psi_i \ \ \overbrace{0 \cdots 0} \end{pmatrix} \quad \text{(A-2)}$$

where 0 denotes the zero matrix with the same size as Ψ_i and $\phi_i = \lceil \frac{\eta_i+1}{p} \rceil - 1$ for i = 1 and 2. By (A-2), multiplying $(\Psi_1 D^{\eta_1})^{[p]}$ by $(\Psi_2 D^{\eta_2})^{[p]}$ directly results in

$$(\Psi_1 D^{\eta_1})^{[p]} \cdot (\Psi_2 D^{\eta_2})^{[p]} = D^{\phi_3} \begin{pmatrix} 0 & \cdots & 0 & \Psi_1 \cdot \Psi_2 & & \\ & & & & \ddots & \\ & & & & & \Psi_1 \cdot \Psi_2 \\ & & & & & 0 \\ & & & & & \vdots \\ & & & & & 0 \end{pmatrix} + D^{\phi_3+1} \begin{pmatrix} 0 & & & & & \\ \vdots & & & & & \\ 0 & & & & & \\ \Psi_1 \cdot \Psi_2 & & & & & \\ & \ddots & & & & \\ & & \Psi_1 \cdot \Psi_2 & 0 & \cdots & 0 \end{pmatrix} \quad \text{(A-3)}$$

where

$$\phi_3 = \begin{cases} \lceil \frac{\eta_1+1}{p} \rceil + \lceil \frac{\eta_2+1}{p} \rceil - 2, & \text{if } 1+(\eta_1 \bmod p)+(\eta_2 \bmod p) \le p \\ \lceil \frac{\eta_1+1}{p} \rceil + \lceil \frac{\eta_2+1}{p} \rceil - 1, & \text{if } 1+(\eta_1 \bmod p)+(\eta_2 \bmod p) > p \end{cases}.$$

Moreover, by (A-3), we have

$$(\Psi_1 \cdot \Psi_2 D^{\eta_1+\eta_2})^{[p]} = D^{\phi_4} \begin{pmatrix} 0 & \cdots & 0 & \Psi_1 \cdot \Psi_2 & & \\ & & & & \ddots & \\ & & & & & \Psi_1 \cdot \Psi_2 \\ & & & & & 0 \\ & & & & & \vdots \\ & & & & & 0 \end{pmatrix} + D^{\phi_4+1} \begin{pmatrix} 0 & & & & & \\ \vdots & & & & & \\ 0 & & & & & \\ \Psi_1 \cdot \Psi_2 & & & & & \\ & \ddots & & & & \\ & & \Psi_1 \cdot \Psi_2 & 0 & \cdots & 0 \end{pmatrix} \quad \text{(A-4)}$$

where $\phi_4 = \lceil \frac{\eta_1+\eta_2+1}{p} \rceil - 1$. Since it can be shown that $\phi_4 = \phi_3 \ \forall\ \eta_1, \eta_2$, comparing (A-3) with (A-4), we thus obtain $(\Psi_1 \cdot \Psi_2 D^{\eta_1+\eta_2})^{[p]} = (\Psi_1 D^{\eta_1})^{[p]} \cdot (\Psi_2 D^{\eta_2})^{[p]}$.

Let $\Theta(D)$ and $G(D)$ be decomposed as $\Theta(D) = \sum_{l=0}^{d_\Theta} \Theta_l D^l$ and $G(D) = \sum_{l=0}^{d_G} G_l D^l$, where Θ_l's and G_l's are some binary matrices, and d_Θ and d_G denote the maximum degrees of $\Theta(D)$ and $G(D)$, respectively. By Properties 1 and

2, we have

$$\begin{aligned}
& (\Theta(D) \cdot G(D))^{[p]} \\
= & (\textstyle\sum_{l_1=0}^{d_\Theta} \Theta_{l_1} D^{l_1} \cdot \sum_{l_2=0}^{d_G} G_{l_2} D^{l_2})^{[p]} \\
= & (\textstyle\sum_{l_1=0}^{d_\Theta} \sum_{l_2=0}^{d_G} \Theta_{l_1} \cdot G_{l_2} D^{l_1+l_2})^{[p]} \\
= & \textstyle\sum_{l_1=0}^{d_\Theta} \sum_{l_2=0}^{d_G} (\Theta_{l_1} \cdot G_{l_2} D^{l_1+l_2})^{[p]} \qquad \text{(by Property 1)} \\
= & \textstyle\sum_{l_1=0}^{d_\Theta} \sum_{l_2=0}^{d_G} (\Theta_{l_1} D^{l_1})^{[p]} \cdot (G_{l_2} D^{l_2})^{[p]} \quad \text{(by Property 2)} \\
= & \textstyle\sum_{l_1=0}^{d_\Theta} (\Theta_{l_1} D^{l_1})^{[p]} \cdot \sum_{l_2=0}^{d_G} (G_{l_2} D^{l_2})^{[p]} \\
= & (\textstyle\sum_{l_1=0}^{d_\Theta} \Theta_{l_1} D^{l_1})^{[p]} \cdot (\sum_{l_2=0}^{d_G} G_{l_2} D^{l_2})^{[p]} \quad \text{(by Property 1)} \\
= & \Theta(D)^{[p]} \cdot G(D)^{[p]}
\end{aligned}$$

therefore completing the proof.

REFERENCES

[1] J. Hagenauer, “Rate-compatible punctured convolutional codes (RCPC codes) and their applications,” *IEEE Trans. Commun.*, vol. 32, pp. 389–400, Apr. 1988.

[2] C.-H. Wang and C.-C. Chao, “Path-compatible pruned convolutional (PCPC) codes,” *IEEE Trans. Commun.*, vol. 50, pp. 213–224, Feb. 2002.

[3] L. H. C. Lee, “New rate-compatible punctured convolutional codes for Viterbi decoding,” *IEEE Trans. Commun.*, vol. 42, pp. 3073–3079, Dec. 1994.

[4] P. K. Frenger, P. Orten, T. Ottosson, and A. B. Svensson, “Rate-compatible convolutional codes for multirate DS-CDMA systems,” *IEEE Trans. Commun.*, vol. 47, pp. 828–836, June 1999.

[5] R. J. McEliece, “The algebraic theory of convolutional codes,” in *Handbook of Coding Theory*, V. S. Pless and W. C. Huffman, eds. Amsterdam, The Netherlands: Elsevier, 1998, pp. 1065–1138.