36-61
53207

TDA Progress Report 42-107

November 15, 1991

P. 16

N92-14243

Some Partial-Unit-Memory Convolutional Codes

K. Abdel-Ghaffar
University of California, Davis

R. J. McEliece[1]

G. Solomon[2]

This article presents the results of a study of a class of error-correcting codes called partial-unit-memory convolutional codes, or PUM codes for short. This class of codes, though not entirely new, has until now remained relatively unexplored. This article shows that it is possible to use the well-developed theory of block codes to construct a large family of promising PUM codes. Indeed, at the end of the article the performances of several specific PUM codes are compared with that of the Voyager standard $(2, 1, 6)$ convolutional code. It was found that these codes can outperform the Voyager code with little or no increase in decoder complexity. This suggests that there may very well be PUM codes that can be used for deep-space telemetry that offer both increased performance and decreased implementational complexity over current coding systems.

I. Introduction

This article gives a general construction for, and several interesting examples of, partial-unit-memory (PUM) convolutional codes. First, some definitions and notation are established.

A convolutional code $\mathcal{C}$ of length n and dimension k over a field F is defined by an encoder

$$G(D) = G_0 + G_1 D + \cdots + G_M D^M \qquad (1)$$

where $G_0, G_1, \ldots, G_M$ are $k \times n$ matrices with entries from F.[3] The ratio $R = k/n$ is called the rate of the code, and M is the code's memory. If $u(D) = u_0 + u_1 D + u_2 D^2 + \cdots$ is the input to the encoder (where the u_i's are elements of F, and D is an indeterminate), then $x(D) = u(D)G(D)$ is the output, which is also called a codeword. The encoder is said to be noncatastrophic if no infinite-weight input produces a finite-weight output. The free distance, d_{free}, of a convolutional code $\mathcal{C}$ is defined to be the minimum weight of any nonzero codeword $x(D) \in \mathcal{C}$. If the encoder $G(D)$ is noncatastrophic, then d_{free} is the minimum weight of all codewords $u(D)G(D)$ generated by inputs $u(D)$ of finite weight. All other things being equal, it is generally desirable to have the quantity

[1] Consultant to the Communications Systems Research Section from the California Insitute of Technology.

[2] Independent consultant to the Communications Systems Research Section.

[3] In this article, it is always assumed that $F = GF(2)$, but most or all of the results generalize easily to other finite fields.

$Q = Rd_{\text{free}}$ as large as possible, since Q is the asymptotic coding gain of the code, which is a good measure of the communications improvement afforded by the use of the code.

A convolutional code $\mathcal{C}$ has state complexity m if the sum of the maximum degrees of the rows of $G(D)$ is m. This terminology reflects the fact that a physical encoder for $\mathcal{C}$ based on $G(D)$ has 2^m states. It is desirable to have m as small as possible, since the computational complexity of the Viterbi decoding algorithm for $\mathcal{C}$ is proportional to 2^m.

The notation "$[n, k, m, d]$ code" is introduced to describe a convolutional code of length n, dimension k, state complexity m, and free distance d. The notation "(n, k, m)" code is sometimes used to describe the same code without explicitly referring to its free distance. For example, in this notation, an $[n, k, d]$ block code, which can be viewed as a convolutional code with $M = 0$, is both an $[n, k, 0, d]$ and an $(n, k, 0)$ convolutional code. For a given n, k, and m, a code for which d_{free} is as large as possible is said to be an optimal (more properly, distance optimal) code.

A convolutional code with $M = 0$ is just a block code. A convolutional code with $M = 1$ is called a unit-memory convolutional code. Unit-memory codes seem to form a class that lies halfway between block and convolutional codes. They were first studied seriously by Lee [5], who found a number of interesting examples of unit-memory convolutional codes. Thommesen and Justesen [10] have obtained bounds on the performance of unit-memory codes, and Justesen, Paaske, and Ballan [3] have constructed a class of unit-memory codes which they call quasi-cyclic codes. This article studies another subclass of unit-memory codes called partial-unit-memory codes.

For a unit-memory convolutional code, the state complexity is just the number of nonzero rows of G_1. If some of the rows of G_1 are zero, i.e., if $m < k$, then it is said that the code is a partial-unit-memory (PUM) convolutional code. Partial-unit-memory codes were introduced by Lauer [4], who constructed several optimal PUM codes. Some general constructions and further examples of PUM codes (under the name finite-state codes) were given in [7] and [8]. This article should be viewed as a continuation of these earlier studies, in which, among other things, it is shown that many of these earlier results follow from the authors' methods. (For example, Lauer's equidistant PUM codes appear in Example 2 in Section III of this article, and the general construction of Theorem 5 in [7] appears as Corollary 4 in Section II.)

Here is a summary of this article. Section II gives a general construction for PUM codes based on the existence of certain block codes. Roughly speaking, the main result is that if there exist two distinct $[n, k, d_0]$ block codes with a common $[n, k^*, d^*]$ subcode, then there exists a noncatastrophic $[n, k, k - k^*, d]$ PUM code with $d \geq \min(d^*, 2d_0)$. (As a point of comparison, Theorem 5 in [7] shows that if there exists a single $[n, k, d_0]$ block code with an $[n, k^*, d^*]$ subcode, then there exists a noncatastrophic $[n, k - 1, k - k^* - 1, d]$ PUM code with $d \geq \min(d^*, 2d_0)$.) Since there is a huge existing catalog of block codes, what this means is that it is possible to construct a very large number of interesting PUM codes. This is illustrated in Section III with several examples called Hamming, Reed-Muller, and Golay PUM codes. While these examples are possibly interesting and potentially important, the authors believe that they have only scratched the surface, and hope that future authors, using these techniques, or ones of their own devising, will unearth many more examples.

II. Main Results

Theorem 1. Suppose that $\mathcal{C}_0$ is an $[n, k, d_0]$ linear block code, $\mathcal{C}_1$ is an $[n, k, d_1]$ linear block code, and $\mathcal{C}_0 \neq \mathcal{C}_1$. Suppose further that $\mathcal{C}_0$ and $\mathcal{C}_1$ contain a common subcode $\mathcal{C}^*$, which is an $[n, k^*, d^*]$ code. Then there exists a noncatastrophic $[n, k, m, d]$ PUM convolutional code, with $m = k - k^*$ and $d \geq \min(d^*, d_0 + d_1)$.

Proof: Begin by choosing $k \times n$ generator matrices G_0 and G_1 for $\mathcal{C}_0$ and $\mathcal{C}_1$ of the form

$$G_0 = \begin{bmatrix} K^* \\ K_0 \end{bmatrix} \qquad G_1 = \begin{bmatrix} K^* \\ K_1 \end{bmatrix} \tag{2}$$

where K^* is a $k^* \times n$ generator matrix for $\mathcal{C}^*$. Note that both K_0 and K_1 are $m \times n$ matrices; for future reference, let $\mathcal{C}_0^*$ and $\mathcal{C}_1^*$ be the corresponding codes, i.e.,

$$\mathcal{C}_0^* = \langle K_0 \rangle \tag{3}$$

$$\mathcal{C}_1^* = \langle K_1 \rangle \tag{4}$$

Next, define the matrix G_1^0 as

$$G_1^0 = \begin{bmatrix} \mathbf{O} \\ K_1 \end{bmatrix} \tag{5}$$

where O is a $k^* \times n$ matrix of 0's. Then define a $k \times n$ polynomial matrix $G(D)$ as follows:

$$G(D) = G_0 + G_1^0 D \tag{6}$$

Plainly, $G(D)$ is the generator matrix for an (n, k, m) PUM code. The proof will be complete when the following two things are shown: (1) $d_{\text{free}} \geq \min(d^*, d_0 + d_1)$, and (2) $G(D)$ is noncatastrophic. Begin with the assertion about d_{free}.

Assume that $u(D) = u_0 + u_1 D + u_2 D^2 + \cdots$ is a finite nonzero input sequence, where $u_0 \neq 0$ and each u_i is a k-dimensional row vector. Then the corresponding output sequence is $x(D) = u(D)G(D) = x_0 + x_1 D + x_2 D^2 + \cdots$, where

$$\left.\begin{aligned} x_0 &= u_0 G_0 \\ x_i &= u_i G_0 + u_{i-1} G_1^0, \qquad \text{for } i \geq 1 \end{aligned}\right\} \tag{7}$$

If $u = (\mu_1, \mu_2, \ldots, \mu_k)$ is a k-dimensional vector, u^L (the left part of u) and u^R (the right part of u) are defined as follows:

$$\left.\begin{aligned} u^L &= (\mu_1, \ldots, \mu_{k^*}) \\ u^R &= (\mu_{k^*+1}, \ldots, \mu_k) \end{aligned}\right\} \tag{8}$$

Then [see Eq. (2)], for any vector u, one has

$$uG_0 = u^L K^* + u^R K_0, \qquad uG_1^0 = u^R K_1 \tag{9}$$

After combining Eq. (8) with Eq. (6), one has

$$\left.\begin{aligned} x_0 &= u_0^L K^* + u_0^R K_0 \\ x_i &= u_i^R K_0 + [u_i^L, u_{i-1}^R] G_1, \qquad \text{for } i \geq 1 \end{aligned}\right\} \tag{10}$$

Now, either $[u_i^L, u_{i-1}^R] = 0$ for all $i \geq 1$, or not. It will be shown that weight $(x(D)) \geq d^*$ in the first case, and that weight $(x(D)) \geq d_0 + d_1$ in the second case.

If $[u_i^L, u_{i-1}^R] = 0$ for all $i \geq 1$, then by Eq. (6) and Eq. (9), one has

$$x(D) = u_0 G_0 = u_0^L K^* \tag{11}$$

which means that $x(D)$ is a nonzero vector in the rowspace of K^*, i.e., a nonzero codeword in $\mathcal{C}^*$, and so weight $(x(D)) \geq d^*$ in this case.

If, on the other hand, $[u_i^L, u_{i-1}^R] \neq 0$ for some $i \geq 1$, let M denote the largest such index. Then, $u_M^R = 0$, and Eq. (9) implies

$$x(D) = u_0 G_0 + \cdots + [u_M^L, u_{M-1}^R] G_1 D^M \tag{12}$$

But $u_0 G_0$ is a nonzero word from $\mathcal{C}_0$, and so has weight $\geq d_0$, and $[u_M^L, u_{M-1}^R] G_1$ is a nonzero word from $\mathcal{C}_1$, and so has weight $\geq d_1$; thus, weight $(x(D)) \geq d_0 + d_1$ in this case, which proves the assertion about d_{free}.

It remains to be shown that $G(D)$ can be chosen noncatastrophically. Lemma 1, which follows, tells, in principle, whether a given $G(D)$ is catastrophic or not. Lemma 2 then tells that it is always possible to choose the matrices G_0 and G_1 so that $G(D)$ is noncatastrophic.

Lemma 1. Let the linear transformation $T : \mathcal{C}_0 \to \mathcal{C}_1$ be defined by $uG_0 \to uG_1$. Then $G(D)$ is noncatastrophic if and only if every subspace of $\mathcal{C}_0$ fixed by T is a subspace of $\mathcal{C}^*$.

Proof: Denote the rows of K^* by $(x_1, x_1, \ldots, x_{k^*})$, the rows of K_0 by $(y_1, y_1, \ldots, y_m)$, and the rows of K_1 by $(z_1, z_2, \ldots, z_m)$. Then T is completely characterized by the k values

$$Tx_i = x_i, \qquad \text{for } i = 1, 2, \ldots, k^* \tag{13}$$

$$Ty_i = z_i, \qquad \text{for } i = 1, 2, \ldots, m \tag{14}$$

Note that Eq. (13) says that T not only fixes $\mathcal{C}^*$, it fixes $\mathcal{C}^*$ pointwise.

It is first assumed that every T-fixed subspace of $\mathcal{C}_0$ is a subspace of $\mathcal{C}^*$, and then shown that $G(D)$ is noncatastrophic. Let $u(D)$ be a nonzero input such that the corresponding output $x(D)$ is finite, i.e., $x_i = 0$ for $i > i_0$. If one defines

$$a_i = u_i^L K^* \in \mathcal{C}^* \tag{15}$$

$$b_i = u_i^R K_0 \in \mathcal{C}_0^* \tag{16}$$

one has, by Eq. (9),

$$x_i = a_i + b_i + Tb_{i-1}, \qquad \text{for } i \geq 1 \tag{17}$$

Thus, since it is assumed that $x_i = 0$ for $i > i_0$, it follows that (recall that the codes are binary)

$$Tb_{i-1} = b_i + a_i, \qquad \text{for } i > i_0 \tag{18}$$

so that $\langle b_{i_0}, b_{i_0+1}, \dots \rangle + \mathcal{C}^*$ is a T-fixed subspace of $\mathcal{C}_0$. However, it is assumed that all T-fixed subspaces of $\mathcal{C}_0$ are subspaces of $\mathcal{C}^*$, and so $b_i \in \mathcal{C}^*$ for all $i \geq i_0$. But since the rows of K_0 are linearly independent of the rows of K^*, this means that $b_i = 0$, and so $u_i^R = 0$, for all $i \geq i_0$. But then, by Eq. (17), $a_i = 0$, and so $u_i^L = 0$, for all $i > i_0$. Thus, the input $u(D)$ is necessarily finite, and so $G(D)$ is noncatastrophic.

Conversely, suppose that B is a nonzero T-fixed subspace of $\mathcal{C}_0$ that properly contains $\mathcal{C}^*$. Choose $a_0 \in \mathcal{C}^*$ and $b_0 \in B - \mathcal{C}_0^*$ arbitrarily. Now, since B is T-fixed, Tb_0 is also in B, and so it can be decomposed uniquely into the sum of an element of $\mathcal{C}^*$, which is called a_1, and an element of $\mathcal{C}_0^*$, which is called b_1. Note that since $b_0 \neq 0$, then $b_1 \neq 0$ also, for otherwise T would map the $k^* + 1$ dimensional space $\mathcal{C}^* + \langle b_0 \rangle$ into the k^*-dimensional space $\mathcal{C}^*$. This process is continued inductively by constructing an infinite sequence of pairs (a_i, b_i), with $a_i \in \mathcal{C}^*$, $b_i \in \mathcal{C}_0^*$, $b_i \neq 0$, such that

$$Tb_i = a_{i+1} + b_{i+1}, \qquad \text{for } i \geq 0 \tag{19}$$

Now for each $i \geq 1$, define the vector u_i as follows:

$$u_i^L K^* = a_i \tag{20}$$

$$u_i^R K_0 = b_i \tag{21}$$

Since $b_i \neq 0$, then by Eq. (20), $u_i \neq 0$, and so the sequence (u_i) is an infinite sequence of nonzero elements. It will now be shown that, if (u_i) is the input, then the corresponding output is finite. One has

$$\begin{aligned} Tb_i &= T(u_i^R K_0) \\ &= T([0, u_i^R]G_0) \\ &= [0, u_i^R]G_1, \qquad \text{by definition of } T \\ &= u_i^R K_1 \end{aligned}$$

and so by Eq. (9), for $i \geq 1$,

$$\begin{aligned} x_i &= u_i^L K^* + u_i^R K_0 + u_{i-1}^R K_1, \qquad \text{by Eq. (9)} \\ &= a_i + b_i + Tb_{i-1}, \qquad \text{by Eqs. (19) and (20)} \\ &= 0, \qquad \text{by Eq. (18)} \end{aligned}$$

Thus, the infinite input sequence (u_i) produces a finite output sequence (x_i), and so $G(D)$ is catastrophic, as was asserted. ❐

Corollary 1. If $\mathcal{C}_0 \cap \mathcal{C}_1 = \mathcal{C}^*$, then any generator matrix of the form Eq. (6) is noncatastrophic.

Proof: If B is a T-fixed subspace of T, then since $T : \mathcal{C}_0 \to \mathcal{C}_1$ and $B = T(B)$, B is a subspace of both $\mathcal{C}_0$ and $\mathcal{C}_1$. Thus, $B \subseteq \mathcal{C}_0 \cap \mathcal{C}_1 = \mathcal{C}^*$, and so by Lemma 1, $G(D)$ is noncatastrophic. ❐

Lemma 1 allows one to tell, in principle, whether or not a given $G(D)$ is catastrophic. Corollary 1 assures that if $\mathcal{C}_0 \cap \mathcal{C}_1 = \mathcal{C}^*$, then nothing can go wrong. However, if $\mathcal{C}_0 \cap \mathcal{C}_1 \supset \mathcal{C}^*$, more work is necessary to find a noncatastrophic generator matrix. Lemma 2, which follows, gives an explicit construction for a noncatastrophic $G(D)$ in the general case.

Lemma 2. Suppose that $\mathcal{C}_0$ and $\mathcal{C}_1$ are subspaces of $V_n(F)$, the n-dimensional vector space over F, with $\mathcal{C}_0 \neq \mathcal{C}_1$ but $\dim(\mathcal{C}_0) = \dim(\mathcal{C}_1) = k$, and that $\mathcal{C}^*$ is a subspace of both $\mathcal{C}_0$ and $\mathcal{C}_1$, with $\dim(\mathcal{C}^*) = k^*$. Then, if $(u_1, u_2, \dots, u_{k^*})$ is a basis for $\mathcal{C}^*$, there exist bases for $\mathcal{C}_0$ and $\mathcal{C}_1$ of the form

$$\left.\begin{aligned} \langle \mathcal{C}_0 \rangle &= (u_1, \dots, u_{k^*}, \alpha_1, \dots, \alpha_m) \\ \langle \mathcal{C}_1 \rangle &= (u_1, \dots, u_{k^*}, \beta_1, \dots, \beta_m) \end{aligned}\right\} \tag{22}$$

such that the linear transformation $T : \mathcal{C}_0 \to \mathcal{C}_1$ defined by

$$\left.\begin{aligned} Tu_i &= u_i, \qquad \text{for } i = 1, \dots, k^* \\ T\alpha_i &= \beta_i, \qquad \text{for } i = 1, \dots, m \end{aligned}\right\} \tag{23}$$

fixes no subspace of $\mathcal{C}_0$ larger than $\mathcal{C}^*$.

Proof: Begin by constructing two descending sequences of subspaces $(\mathcal{A}_i)$ and $(\mathcal{B}_i)$:

$$\mathcal{C}_0 = \mathcal{A}_0 \supset \mathcal{A}_1 \supset \cdots \supset \mathcal{A}_{N+1} = \mathcal{C}^*$$

$$\mathcal{C}_1 = \mathcal{B}_0 \supset \mathcal{B}_1 \supset \cdots \supset \mathcal{B}_{N+1} = \mathcal{C}^*$$

such that

$$\dim(\mathcal{A}_i) = \dim(\mathcal{B}_i), \qquad i = 0, 1, \ldots, N+1$$

$$\mathcal{A}_{i+1} = \mathcal{A}_i \cap \mathcal{B}_i, \qquad i = 0, 1, \ldots, N$$

Figure 1 illustrates the construction of Lemma 2. This construction can be done inductively as follows. Assume that $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_i$ and $\mathcal{B}_0, \mathcal{B}_1, \ldots, \mathcal{B}_i$ have already been constructed. (For $i = 0$, this simply requires setting $\mathcal{A}_0 = \mathcal{C}_0$ and $\mathcal{B}_0 = \mathcal{C}_1$.) Let $\mathcal{A}_{i+1} = \mathcal{A}_i \cap \mathcal{B}_i$. If $\mathcal{A}_{i+1} = \mathcal{C}^*$, define $\mathcal{B}_{i+1} = \mathcal{C}^*$, $N = i$, and stop. Otherwise, one has $\mathcal{C}^* \subset \mathcal{A}_{i+1} \subset \mathcal{B}_i$, and so by Lemma A2 in Appendix A, there exists a subspace $\mathcal{B}_{i+1} \neq \mathcal{A}_{i+1}$ such that $\mathcal{C}^* \subset \mathcal{B}_{i+1} \subset \mathcal{B}_i$, with $\dim \mathcal{B}_{i+1} = \dim \mathcal{A}_{i+1}$.

Now define integers $k_0, \ldots, k_N$ by $k_i = \dim(\mathcal{A}_i) - k^*$, so that

$$k_N < k_{N-1} < \cdots < k_0 = m$$

Next, using Lemma A1 in Appendix A, choose bases $\langle u_1, \ldots, u_{k^*}, \alpha_1, \ldots, \alpha_m \rangle$ for $\mathcal{C}_0$ and $\langle u_1, \ldots, u_{k^*}, \beta_1, \ldots, \beta_m \rangle$ for $\mathcal{C}_1$ such that

$$\left.\begin{aligned} \langle u_1, \ldots, u_{k^*}, \alpha_1, \ldots, \alpha_{k_i} \rangle &= \mathcal{A}_i \\ \langle u_1, \ldots, u_{k^*}, \beta_1, \ldots, \beta_{k_i} \rangle &= \mathcal{B}_i \end{aligned}\right\} \qquad (24)$$

for $i = 1, 2, \ldots, N$. Now define the transformation T as in Eq. (22). Plainly, T fixes $\mathcal{C}^*$ pointwise, and also, from Eq. (23),

$$T : \mathcal{A}_i \to \mathcal{B}_i, \qquad \text{for } i = 0, 1, \ldots, N \qquad (25)$$

If now $\mathcal{D}$ is a subspace of $\mathcal{C}_0$ fixed by T, then $\mathcal{D} \subseteq \mathcal{C}_0 = \mathcal{A}_0$, and $\mathcal{D} = T(\mathcal{D}) \subseteq T(\mathcal{C}_0) = \mathcal{C}_1 = \mathcal{B}_0$, so that $\mathcal{D} \subseteq \mathcal{A}_0 \cap \mathcal{B}_0 = \mathcal{A}_1$. Also, $\mathcal{D} = T(\mathcal{D}) \subseteq T(\mathcal{A}_1) = \mathcal{B}_1$, using Eq. (24), so that $\mathcal{D} \subseteq \mathcal{A}_1 \cap \mathcal{B}_1 = \mathcal{A}_2$. Continuing inductively, one finds that in fact $\mathcal{D} \subseteq \mathcal{A}_3, \ldots, \mathcal{D} \subseteq \mathcal{A}_{N+1} = \mathcal{C}^*$. Thus, a linear transformation T is constructed such that any T-fixed subspace of $\mathcal{C}_0$ is a subspace of $\mathcal{C}^*$. ❐

Lemma 2 tells how to construct noncatastrophic generator matrices G_0 and G_1: Just let the rows of G_0 be the vectors $(u_1, \ldots, u_{k^*}, \alpha_1, \ldots, \alpha_m)$, and let the rows of G_1 be the vectors $(u_1, \ldots, u_{k^*}, \beta_1, \ldots, \beta_m)$ in Eq. (21). Then, the mapping $T : \mathcal{C}_0 \to \mathcal{C}_1$ defined by $uG_0 \to uG_1$ is the same as the mapping described in Lemma 2, and so the resulting $G(D)$ is noncatastrophic. This completes the proof of Theorem 1. ❐

Corollary 2. Suppose that $\mathcal{C}_0$ is an $[n, k, d_0]$ linear block code, and that $\mathcal{C}^*$ is an $[n, k^*, d^*]$ code, which is a subcode of $\mathcal{C}_0$. If the automorphism group of $\mathcal{C}^*$ contains a permutation that does not fix $\mathcal{C}_0$, then there exists an $[n, k, m, d]$ PUM convolutional code, with $m = k - k^*$ and $d \geq \min(d^*, 2d_0)$.

Proof: Let π be an automorphism of $\mathcal{C}^*$ that does not fix $\mathcal{C}_0$, and let $\mathcal{C}_1 = \mathcal{C}_0^{\pi}$. Then, $\mathcal{C}_1$ is an $[n, k, d_0]$ code not equal to $\mathcal{C}_0$. Now apply Theorem 1. ❐

Corollary 3. If $\mathcal{C}_0$ is an $[n, k, d_0]$ linear block code that contains the all-ones vector, and if $k \neq 1, n-1, n$, then there exists an $(n, k, k-1)$ PUM code with $d_{\text{free}} \geq 2d_0$.

Proof: Here, Corollary 2 is applied, with $\mathcal{C}^*$ being the $[n, 1, n]$ code consisting of the two vectors $[00\cdots]$ and $[11\cdots1]$. Clearly $\mathcal{C}^*$ is fixed by all permutations of $\{1, 2, \ldots, n\}$. Furthermore, the only binary linear codes that are fixed by all permutations of $\{1, 2, \ldots, n\}$ have dimensions 0, 1, $n-1$, or n, so there must be an automorphism of $\mathcal{C}^*$ that doesn't fix $\mathcal{C}_0$. Thus, by Corollary 2, there exists an $(n, k, k-1)$ PUM code with $d_{\text{free}} \geq \min(2d_0, n)$. However, since $k \geq 2$, the mimimum distance d_0 of $\mathcal{C}_0$ must be $< n$; and since $\mathcal{C}_0$ contains the all-ones vector, there must be a word of weight $n - d_0$. Hence, $n - d_0 \leq d_0$, and so $d_0 \leq n/2$. Hence, $\min(2d_0, n) = 2d_0$, so that in fact $d_{\text{free}} \geq 2d_0$. ❐

Corollary 4. (Same as Theorem 5 in [7]). Suppose that $\mathcal{C}_0$ is an $[n, k, d_0]$ linear block code, $\mathcal{C}^*$ is an $[n, k^*, d^*]$ code that is a subcode of $\mathcal{C}_0$, and $k - k^* 2$. Then, for every integer i in the range $1 \leq i \leq k - k^* - 1$, there is a noncatastrophic $[n, k-i, k-i-k^*, d]$ PUM code with $\delta \geq \min(d^*, 2d_0)$.

Proof: Let $\mathcal{C}_0'$ be any $(n, k-i)$ subcode of $\mathcal{C}_0$ that contains $\mathcal{C}^*$. (The conditions on i guarantee that $\dim \mathcal{C}^* z \leq \dim \mathcal{C}_0' \leq \dim \mathcal{C}_0$, so this is possible.) By Lemma A2, there exists a subcode $\mathcal{C}_1'$ not equal to $\mathcal{C}_0'$ but having the same dimension, and also lying between $\mathcal{C}_0$ and $\mathcal{C}^*$. Thus, $\mathcal{C}_0'$ and $\mathcal{C}_1'$ are both $[n, k-i, d']$ block codes, with $d' \geq d_0$. By applying Theorem 1 to the codes $\mathcal{C}_0'$, $\mathcal{C}_1'$, and $\mathcal{C}^*$, one obtains a noncatastrophic $[n, k-i, k-i-k^*, d]$ PUM code with $\delta \geq \min(d^*, 2d') \geq \min(d^*, 2d_0)$. ❐

III. Examples

In this section, four examples of PUM codes are presented, that were constructed with the help of the results in Section II. Example 1 describes a Hamming $[8,4,3,8]$ PUM code, which was originally discovered by Lauer [4]. Example 2 gives a generalization of Example 1 to a class of Reed-Muller $[\nu 2^\mu, \mu+1, \mu, \nu 2^\mu]$ PUM codes, one code for each pair of positive integers (μ,ν) except $(1,1)$ and $(2,1)$. (The code of Example 1 corresponds to the pair $(3,1)$.) The codes in Example 2 were also found, using different methods, by Lauer. Finally, in Examples 3 and 4, two new PUM Golay codes, with parameters $[24,12,7,12]$ and $[24,12,10,16]$, are presented.

Example 1 (a Hamming PUM Code). Let $\mathcal{C}_0'$ be the $[7,4,3]$ binary cyclic code with generator polynomial $g_0(x) = 1+x+x^3$, and let $\mathcal{C}_1'$ be the $[7,4,3]$ binary cyclic code with generator polynomial $g_1(x) = 1+x^2+x^3$. Take as a generator matrix for $\mathcal{C}_0'$ the 4×7 binary matrix G_0', whose rows are $g_1(x)g_0(x)$, $g_0(x)$, $xg_0(x)$, and $x^2g_0(x)$, and for $\mathcal{C}_1'$ the 4×7 binary matrix G_1', whose rows are $g_0(x)g_1(x)$, $g_1(x)$, $xg_1(x)$, and $x^2g_1(x)$, i.e.,

$$G_0' = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

$$G_1' = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Now, if each code is extended to length 8 by appending an overall parity-check, one obtains codes $\mathcal{C}_0$ and $\mathcal{C}_1$, both of which are binary $[8,4,4]$ codes with generator matrices

$$G_0 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

$$G_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Since $\mathcal{C}_0' \cap \mathcal{C}_1'$ is the binary $[7,1,7]$ repetition code, it follows that $\mathcal{C}_0 \cap \mathcal{C}_1$ is the binary $[8,1,8]$ repetition code. Thus, in Theorem 1, $\mathcal{C}^*$ can be taken to be the $[8,1,8]$ repetition code with the 1×8 generator matrix

$$K^* = (1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1)$$

It follows from Corollary 1 that the matrix

$$G(D) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1+D & 1+D & 1 & D & 1+D & 0 & 0 & 0 \\ 1+D & 0 & 1+D & 1 & D & 1+D & 0 & 0 \\ 1+D & 0 & 0 & 1+D & 1 & D & 1+D & 0 \end{pmatrix}$$

generates a noncatastrophic $[8,4,3,8]$ PUM code. Furthermore, from [4] (Formula (3) with $L=0$) or [7] (Corollary 1 to Theorem 1, with $L=1$), any $(8,4,3)$ convolutional code must have $d_{\text{free}} \leq 8$, so this code is optimal.[4] ❐

Example 2 (Some Reed-Muller PUM Codes). Let μ and $\nu \geq 1$ be positive integers. Let A_μ be the $[2^\mu, \mu+1, 2^{\mu-1}]$ first-order Reed-Muller code, and let B_μ be the $[2^\mu, 1, 2^\mu]$ zeroth-order Reed-Muller code (a repetition code), which is a subcode of A_μ.[5] Now let $C_0(\mu,\nu)$ be the $[\nu 2^\mu, \mu+1, \nu 2^{\mu-1}]$ code obtained by repeating A_μ ν times, and let $\mathcal{C}^*(\mu,\nu)$ be the $[\nu 2^\mu, 1, \nu 2^\mu]$ code obtained by repeating B_μ ν times. Then, according to Corollary 3, unless $(\mu,\nu) = (1,1)$ or $(2,1)$, there exists a noncatastrophic $[\nu 2^\mu, \mu+1, \mu, \nu 2^\mu]$ PUM code. These codes are all optimal by the above-cited bounds in [4] or [7]. (This family of codes was originally constructed by Lauer [4], using a different approach. He called them equidistant PUM

[4] This code first appeared in the literature in [4], Table 1. It is apparently used by the Soviets in their *Regatta* space communication system.

[5] MacWilliams and Sloane [6], Chapter 13, is a good reference for Reed-Muller codes.

codes. A similar family of $[2^\mu, \mu, \mu-1, 2^\mu]$ codes was constructed in [7], Example 4.) ❐

Example 3 (the [24, 12, 7, 12] Golay PUM Code). It is well known that there exists a $[24, 12, 8]$ binary linear code; viz., the famous Golay code. It turns out that there are two isomorphic copies of the Golay code that contain a common $[24, 5, 12]$ subcode, so that by Theorem 1, there exists a noncatastrophic $[24, 12, 7, 12]$ PUM convolutional code. In this example, the construction of this code is detailed.

Define, for $A, B, C, D, E, F \in GF(8)$, the following two functions:

$$f_{A,B,C}(x, y) = \mathrm{Tr}(Axy) + \mathrm{Tr}\big((B + Cy)x^6\big) \tag{26}$$

$$g_{D,E,F}(x, y) = \mathrm{Tr}\big((Dy + E)x\big) + \mathrm{Tr}\big((Fy)x^6\big) \tag{27}$$

where $\mathrm{Tr}(x) = x + x^2 + x^4$ is the trace mapping from $GF(8)$ to $GF(2)$. Then, if β is a fixed nonzero element of trace 0 in $GF(8)$, the following set of 2^{12} length-24 vectors is a $[24, 12, 8]$ Golay code, which is called A_0:

$$A_0 = \big[f_{A,B,C}(x, \beta) + \epsilon_0 | f_{A,B,C}(x, \beta^2) + \epsilon_1 |$$

$$\times f_{A,B,C}(x, \beta^4) + \epsilon_2\big]_{x \in GF(8)} \tag{28}$$

In Eq. (28), the parameters A, B, and C assume all values in $GF(8)$, and the parameters ϵ_0, ϵ_1, and ϵ_2 assume all values in $GF(2)$. The proof that A_0 is indeed a $[24, 12, 8]$ code appears in Appendix B as Lemma B5.

Similarly, the following set of 2^{12} length-24 vectors is another $[24, 12, 8]$ Golay code, which is called B_0:

$$B_0 = \big[g_{D,E,F}(x, \beta) + \delta_0 | g_{D,E,F}(x, \beta^2)$$

$$+ \delta_1 | g_{D,E,F}(x, \beta^4) + \delta_2\big]_{x \in GF(8)} \tag{29}$$

In Eq. (29), the parameters D, E, and F assume all values in $GF(8)$, and the parameters δ_0, δ_1, and δ_2 assume all values in $GF(2)$. The proof that B_0 is indeed a $[24, 12, 8]$ code appears in Appendix B as Lemma B6.

In Appendix B (Lemma B10), it is shown that $A_0 \cap B_0$ is a $[24, 9, 8]$ code consisting of the following set of vectors:

$$A_1 = \big[f_{A,0,C}(x, \beta) + \epsilon_0 | f_{A,0,C}(x, \beta^2) + \epsilon_1 |$$

$$\times f_{A,0,C}(x, \beta^4) + \epsilon_2\big]_{x \in GF(8)} \tag{30}$$

The code A_1, in turn, contains a $[24, 5, 12]$ subcode consisting of the following set of vectors:

$$A_3 = \big[f_{0,0,C}(x, \beta) + \epsilon_0 | f_{0,0,C}(x, \beta^2) + \epsilon_1 |$$

$$\times f_{0,0,C}(x, \beta^4) + \epsilon_0 + \epsilon_1\big]_{x \in GF(8)} \tag{31}$$

(see Lemma B8 in Appendix B). Finally, A_3 contains a $[24, 2, 16]$ subcode A_4:

$$A_4 = [\epsilon_0 | \epsilon_1 | \epsilon_0 + \epsilon_1]_{x \in GF(8)} \tag{32}$$

(see Lemma B9). It follows from Theorem 1 that there exists both a $[24, 12, 7, 12]$ code and a $[24, 12, 10, 16]$ code, and by the bounds in [4] and [7], they are optimal.[6] To actually construct noncatastrophic generator matrices for these two codes, however, more work is necessary. Here are the needed intermediate subspaces (see Fig. 2):

$$B_1 = \big[g_{0,E,F}(x, \beta) + \delta_0 | g_{0,E,F}(x, \beta^2) + \delta_1 |$$

$$\times g_{0,E,F}(x, \beta^4) + \delta_2\big]_{x \in GF(8)} \tag{33}$$

$$A_2 = \big[f_{0,0,C}(x, \beta) + \epsilon_0 | f_{0,0,C}(x, \beta^2) + \epsilon_1 |$$

$$\times f_{0,0,C}(x, \beta^4) + \epsilon_2\big]_{x \in GF(8)} \tag{34}$$

$$B_2 = \big[g_{0,e,0}(x, \beta) + \delta_0 | g_{0,e,0}(x, \beta^2) + \delta_1 |$$

$$\times g_{0,e,0}(x, \beta^4) + \delta_2\big]_{x \in GF(8)} \tag{35}$$

In Eq. (35), e assumes only the two values 0 and 1.

[6] In [8], $[24, 5, 12]$ and $[24, 2, 16]$ subcodes of a $[24, 12, 8]$ Golay code were found, which led, via Corollary 4, to the construction of both $[24, 11, 6, 12]$ and $[24, 11, 9, 16]$ PUM codes.

In order to show that the subspaces in Fig. 2 behave as depicted, the following must be proved:

$$A_0 \cap B_0 = A_1 \tag{36}$$

$$A_1 \cap B_1 = A_2 \tag{37}$$

$$A_2 \cap B_2 = A_3 \tag{38}$$

These relationships are proved in Appendix B in Lemmas B10, B11, and B12.

It thus follows that for the $[24, 12, 7, 12]$ code, a noncatastrophic choice for G_0 and G_1 is as follows:

$$G_0 = \begin{bmatrix} f_{001} \\ f_{00\beta} \\ f_{00\beta^2} \\ \mathbf{011} \\ \mathbf{101} \\ \mathbf{100} \\ f_{100} \\ f_{\beta 00} \\ f_{\beta^2 00} \\ f_{010} \\ f_{0\beta 0} \\ f_{0\beta^2 0} \end{bmatrix} \qquad G_1 = \begin{bmatrix} f_{001} \\ f_{00\beta} \\ f_{00\beta^2} \\ \mathbf{011} \\ \mathbf{101} \\ g_{010} \\ \mathbf{100} \\ g_{0\beta 0} \\ g_{0\beta^2 0} \\ g_{100} \\ g_{\beta 00} \\ g_{\beta^2 00} \end{bmatrix}$$

Here, $f_{A,B,C}$ denotes the length-24 vector obtained by taking $\epsilon_0 = \epsilon_1 = \epsilon_2 = 0$ in Eq. (28), and $g_{A,B,C}$ denotes the length-24 vector obtained by taking $\delta_0 = \delta_1 = \delta_2 = 0$ in Eq. (29). The first five rows of G_0 and G_1 are identical, and they generate the $[24, 5, 12]$ subcode referred to above. In binary, these two matrices are as follows:

$$G_0 = \begin{bmatrix} 01110100 & 00111010 & 01001110 \\ 00111010 & 10011100 & 10100110 \\ 10011100 & 01001110 & 11010010 \\ 00000000 & 11111111 & 11111111 \\ 11111111 & 00000000 & 11111111 \\ 11111111 & 00000000 & 00000000 \\ 00101110 & 01011100 & 01110010 \\ 01011100 & 10111000 & 11100100 \\ 10111000 & 01110010 & 11001010 \\ 11101000 & 11101000 & 11101000 \\ 01110100 & 01110100 & 01110100 \\ 00111010 & 00111010 & 00111010 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} 01110100 & 00111010 & 01001110 \\ 00111010 & 10011100 & 10100110 \\ 10011100 & 01001110 & 11010010 \\ 00000000 & 11111111 & 11111111 \\ 11111111 & 00000000 & 11111111 \\ 10010110 & 10010110 & 10010110 \\ 11111111 & 00000000 & 00000000 \\ 00101110 & 00101110 & 00101110 \\ 01011100 & 01011100 & 01011100 \\ 00101110 & 01011100 & 01110010 \\ 01011100 & 10111000 & 11100100 \\ 10111000 & 01110010 & 11001010 \end{bmatrix}$$

□

Example 4 (the $[24, 12, 10, 16]$ Golay PUM Code). The $[24, 5, 12]$ binary linear code of Example 3 contains a $[24, 2, 16]$ subcode, so that by Theorem 1, there exists a noncatastrophic $[24, 12, 10, 16]$ PUM convolutional code. In this example, the construction of this code is detailed. The subspaces A_0, B_0, A_1, B_1, A_2, and A_4 defined in Example 3 are used. Additionally, subspaces B_2', A_3', and B_3' are defined as follows:

$$B_2' = [g_{0,E,0}(x, \beta) + \delta_0 | g_{0,E,0}(x, \beta^2) + \delta_1 | g_{0,E,0}(x, \beta^4) + \delta_2]_{x \in GF(8)} \tag{39}$$

$$A_3' = [\epsilon_0 | \epsilon_1 | \epsilon_2]_{x \in GF(8)} \tag{40}$$

$$B_3' = [g_{0,e,0}(x, \beta) + \delta_0 | g_{0,e,0}(x, \beta^2) + \delta_1 | g_{0,e,0}(x, \beta^4) + \delta_0 + \delta_1]_{x \in GF(8)} \tag{41}$$

In Eq. (41), e assumes only the two values 0 and 1.

The proof that the subspaces behave as depicted in Fig. 3, i.e., that $A_2 \cap B_2' = A_3'$ and $A_3' \cap B_3' = A_4$, is given in Appendix B, Lemmas B13 and B14.

Now one can see that a noncatastrophic choice for G_0 and G_1 for this code is as follows:

$$G_0 = \begin{bmatrix} \mathbf{011} \\ \mathbf{101} \\ \mathbf{100} \\ f_{001} \\ f_{00\beta} \\ f_{00\beta^2} \\ f_{100} \\ f_{\beta 00} \\ f_{\beta^2 00} \\ f_{010} \\ f_{0\beta 0} \\ f_{0\beta^2 0} \end{bmatrix} \qquad G_1 = \begin{bmatrix} \mathbf{011} \\ \mathbf{101} \\ g_{010} \\ \mathbf{100} \\ g_{0\beta 0} \\ g_{0\beta^2 0} \\ f_{001} \\ f_{00\beta} \\ f_{00\beta^2} \\ f_{100} \\ f_{\beta 00} \\ f_{\beta^2 00} \end{bmatrix}$$

In binary, these matrices are

$$G_0 = \begin{bmatrix} 00000000 & 11111111 & 11111111 \\ 11111111 & 00000000 & 11111111 \\ 11111111 & 00000000 & 00000000 \\ 01110100 & 00111010 & 01001110 \\ 00111010 & 10011100 & 10100110 \\ 10011100 & 01001110 & 11010010 \\ 00101110 & 01011100 & 01110010 \\ 01011100 & 10111000 & 11100100 \\ 10111000 & 01110010 & 11001010 \\ 11101000 & 11101000 & 11101000 \\ 01110100 & 01110100 & 01110100 \\ 00111010 & 00111010 & 00111010 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} 00000000 & 11111111 & 11111111 \\ 11111111 & 00000000 & 11111111 \\ 10010110 & 10010110 & 10010110 \\ 11111111 & 00000000 & 00000000 \\ 00101110 & 00101110 & 00101110 \\ 01011100 & 01011100 & 01011100 \\ 01110100 & 00111010 & 01001110 \\ 00111010 & 10011100 & 10100110 \\ 10011100 & 01001110 & 11010010 \\ 00101110 & 01011100 & 01110010 \\ 01011100 & 10111000 & 11100100 \\ 10111000 & 01110010 & 11001010 \end{bmatrix}$$

□

The codes in Examples 1, 3, and 4 are quite interesting as combinatorial objects, but they have potential for applications. To illustrate, Fig. 4 shows a plot of the performance of these three codes and the NASA standard [2, 1, 6, 10] (non-PUM) code on an additive white Gaussian channel. Figure 4 shows that the low-complexity Hamming [8, 4, 3, 8] code is only a bit weaker than the NASA code, while the two Golay codes are both a bit stronger. Since the state complexity of the [24, 12, 7, 12] Golay code is only 1 greater than that of the NASA code, it may be that there is a relatively low-complexity decoding algorithm for this code, whose performance will significantly exceed that of the NASA code. In any case, these performance curves certainly justify a serious study of efficient decoding algorithms for these and other PUM codes.

Acknowledgment

The authors thank Ivan Onyszchuk of the Communications Systems Research Section for providing the curves used in Fig. 4.

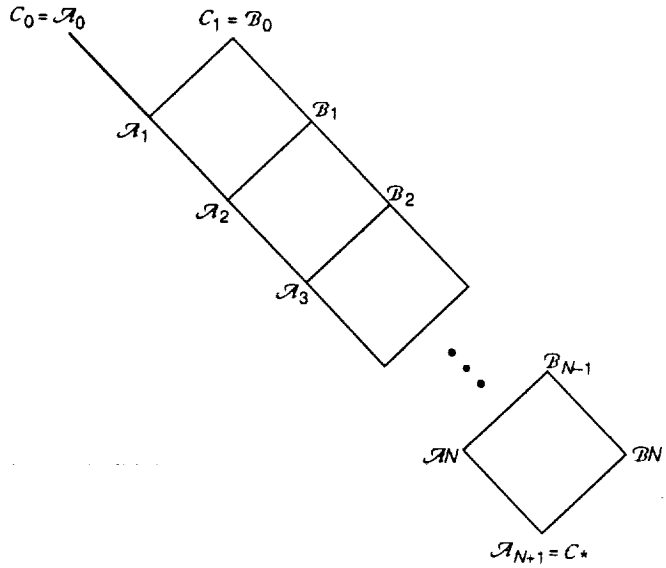

Fig. 1. The construction of Lemma 2.

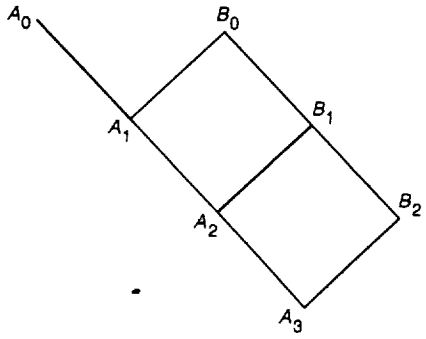

Fig. 2. The subspaces needed for the construction of a noncatastrophic encoder for the [24, 12, 7, 12] PUM code.

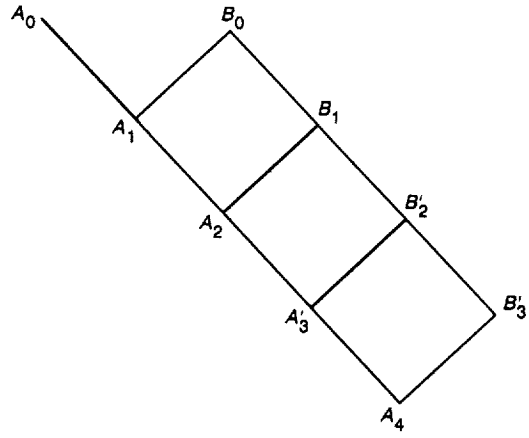

Fig. 3. The subspaces needed for the construction of a noncatastrophic encoder for the [24, 12, 10, 16] code.

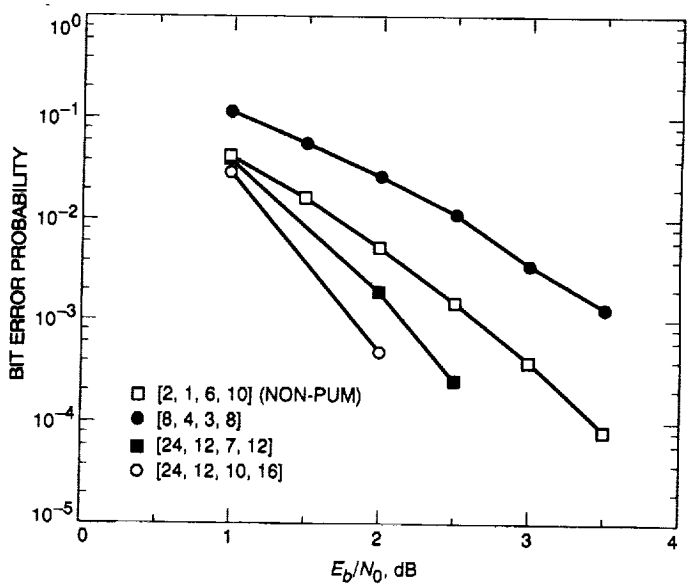

Fig. 4. Performance curves for three PUM codes, compared with the NASA standard [2, 1, 6, 10] code, on an additive white Gaussian channel.

Appendix A

Two Results From Linear Algebra

In this Appendix, two simple results from linear algebra are provided that are needed in the proof of Lemma 2.

Lemma A1. If $\langle 0 \rangle = \mathcal{A}_0 \subseteq \mathcal{A}_1 \subseteq \cdots \subseteq \mathcal{A}_m = \mathcal{V}$ is an ascending chain of subspaces of an n-dimensional vector space $\mathcal{V}$, with $\dim \mathcal{A}_i = k_i$, then there exists a basis $\langle \alpha_1, \ldots, \alpha_n \rangle$ for $\mathcal{V}$ such that

$$\langle \alpha_1, \ldots, \alpha_{k_i} \rangle = \mathcal{A}_i, \qquad \text{for } i = 1, \ldots, m \qquad \text{(A-1)}$$

Proof: One proceeds recursively, as follows. Choose a basis $\langle \alpha_1, \ldots, \alpha_{k_1} \rangle$ for $\mathcal{A}_1$. If $m = 1$, one is done. Otherwise, by using a standard result in linear algebra [2, Lemma 4.2.5], the basis $\langle \alpha_1, \ldots, \alpha_{k_1} \rangle$ for $\mathcal{A}_1$ can be extended to a basis $\langle \alpha_1, \ldots, \alpha_{k_1}, \ldots, \alpha_{k_2} \rangle$ of $\mathcal{A}_2$, etc. ❐

Lemma A2. Suppose that $\mathcal{V}$ is an n-dimensional vector space over F, and $\mathcal{S}$ and $\mathcal{T}$ are subspaces of $\mathcal{V}$ with $\mathcal{S} \subset \mathcal{T} \subset \mathcal{V}$. Then there exists a subspace $\mathcal{T}' \neq \mathcal{T}$ such that $\dim \mathcal{T}' = \dim \mathcal{T}$ and $\mathcal{S} \subset \mathcal{T}' \subset \mathcal{V}$.

Proof: Suppose that $\dim \mathcal{S} = k$ and $\dim \mathcal{T} = k + j$, where $j > 0$. By Lemma A1, it is possible to find a basis for $\mathcal{V}$ of the form

$$\langle \mathcal{V} \rangle = \langle \alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_j, \gamma_1, \ldots, \gamma_h \rangle$$

where $k + j + h = n$, and

$$\langle \alpha_1, \ldots, \alpha_k \rangle = \mathcal{S}$$

$$\langle \alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_j \rangle = \mathcal{T}$$

If $h \geq j$, define $\mathcal{T}'$ as follows:

$$\mathcal{T}' = \langle \alpha_1, \ldots, \alpha_k, \gamma_1, \ldots, \gamma_j \rangle$$

If, on the other hand, $h < j$, define $\mathcal{T}'$ as follows:

$$\mathcal{T}' = \langle \alpha_1, \ldots, \alpha_k, \gamma_1, \ldots, \gamma_h, \beta_1, \ldots, \beta_{j-h} \rangle$$

❐

Appendix B

Proofs Needed in Examples 3 and 4

In this Appendix, the assertions made in Section III about the subspaces A_0, A_1, A_2, A_3, A_3', A_4, B_0, B_1, B_2, B_2', and B_3' of $V_{24}(2)$ are proved.

Lemma B1. Let A and B be elements of $GF(8)$ such that $\mathrm{Tr}(Ax + Bx^6) = 0$ for all $x \in GF(8)$. Then $A = B = 0$.

Proof: Since $\mathrm{Tr}(y) = y + y^2 + y^4$ and $y^8 = y$ for all $y \in GF(8)$, it follows that

$$\mathrm{Tr}(Ax + Bx^6) = Ax + A^2x^2 + A^4x^4 + Bx^6 + B^2x^5 + B^4x^3 \tag{B-1}$$

for all $x \in GF(8)$. Thus, the equation $\mathrm{Tr}(Ax + Bx^6) = 0$ is a polynomial equation of sixth degree with 8 roots in $GF(8)$, and hence, the coefficients of the polynomial must be zero, i.e., $A = B = 0$, as asserted. ❐

Lemma B2. Let $f_{A,B,C}(x, y)$ be defined as in Eq. (26), and suppose there exists a nonzero element y^* in $GF(8)$ such that

$$f_{A,B,C}(x, y^*) = 0, \qquad \text{for all } x \in GF(8) \tag{B-2}$$

Then, $A = 0$ and $B = Cy^*$. Further, if Eq. (B-2) holds and if $f_{A,B,C}(x, y)$ is not identically zero, then for any $y \neq y^*$, the number of solutions $x \in GF(8)$ to $f_{A,B,C}(x, y) = 0$ is exactly four.

Proof: If Eq. (B-2) holds, then by Eq. (26), one has

$$\mathrm{Tr}\big((Ay^*)x + (B + Cy^*)x^6\big) = 0, \qquad \text{for all } x \in GF(8) \tag{B-3}$$

Then, since $y^* \neq 0$, Lemma B1 implies that $A = 0$ and $B + Cy^* = 0$, i.e., $B = Cy^*$. This proves the first statement of the Lemma. To prove the second statement, assume that Eq. (B-2) holds and $f_{A,B,C}(x, y)$ is not identically zero. Then, since it is already known that $A = 0$ and $B = Cy^*$, it must be true that $C \neq 0$, so that the equation $f_{A,B,C}(x, y) = 0$ becomes

$$\mathrm{Tr}\big((Cy^* + Cy)x^6\big) = 0 \tag{B-4}$$

Since $C \neq 0$ and $y \neq y^*$, it follows that $Cy^* + Cy \neq 0$, so that Eq. (B-4) has the form $\mathrm{Tr}(Dx^6) = 0$, with $D \neq 0$. But since for $z \in GF(8)$, $\mathrm{Tr}(z) = 0$ has exactly four solutions, viz., $z = 0, \beta, \beta^2, \beta^4$, it follows that Eq. (B-4) has exactly four solutions. ❐

Lemma B3. Let $g_{D,E,F}(x, y)$ be defined as in Eq. (27), and suppose there exists a nonzero element y^* in $GF(8)$ such that

$$g_{D,E,F}(x, y^*) = 0, \qquad \text{for all } x \in GF(8) \tag{B-5}$$

Then, $F = 0$ and $E = Dy^*$. Further, if Eq. (B-5) holds and if $g_{D,E,F}(x, y)$ is not identically zero, then for any $y \neq y^*$, the number of solutions $x \in GF(8)$ to $g_{D,E,F}(x, y) = 0$ is exactly four.

Proof: The proof of Lemma B3 is similar to the proof of Lemma B2 and is omitted. ❐

Lemma B4. Let A, B, C, D, E, and F be elements of $GF(8)$ such that

$$f_{A,B,C}(x, y) = g_{D,E,F}(x, y) \tag{B-6}$$

for all $x \in GF(8)$, for two distinct values of y, say, $y = y_1$ and $y = y_2$. Then, $A = D$, $C = F$, and $B = E = 0$.

Proof: In view of the definitions in Eq. (26) and Eq. (27) of f and g, the given conditions are equivalent to

$$\mathrm{Tr}\big((Ay + Dy + E)x + (B + Cy + Fy)x^6\big) = 0, \qquad \text{for all } x \in GF(8) \tag{B-7}$$

for $y = y_1, y_2$. Thus, according to Lemma B1, $Ay + Dy + E = 0$ and $B + Cy + Fy = 0$ for $y = y_1, y_2$. The two equations $Ay_i + Dy_i + E = 0$ imply that $A = D$ and $E = 0$, and the two equations $Cy_i + Fy_i + B = 0$ imply that $C = F$ and $B = 0$. ❐

Lemma B5. The code A_0 defined in Eq. (28) is a $[24, 12, 8]$ code.

Proof: The mapping from 6-tuples $[A, B, C, \epsilon_0, \epsilon_1, \epsilon_2]$ to codewords in A_0 is linear. The kernel of this mapping is the set of 6-tuples such that the corresponding codeword is 0, i.e.,

$$f_{A,B,C}(x, \beta^{2^i}) + \epsilon_i = 0, \qquad \text{for } i = 0, 1, 2 \tag{B-8}$$

for all $x \in GF(8)$. By substituting $x = 0$ into these equations, one finds that $\epsilon_i = 0$ for $i = 0, 1, 2$, so that in fact

$$f_{A,B,C}(x, \beta^{2^i}) = 0, \qquad \text{for } i = 0, 1, 2 \tag{B-9}$$

for all $x \in GF(8)$. It then follows from Lemma B2 that $A = B = C = 0$. Thus, the kernel of the mapping contains only the 6-tuple $[0, 0, 0, 0, 0, 0]$, and so the mapping is one-to-one. But since the set of 6-tuples is 12-dimensional, it follows that the code is also 12-dimensional. Thus, A_0 is a $(24, 12)$ code. It remains to prove that its minimum distance is 8.

To show that the minimum distance is 8, first consider the $(24, 9)$ code A_0' defined by Eq. (28) with $\epsilon_0 = \epsilon_1 = \epsilon_2 = 0$. Each word in A_0' has three 8-bit segments, viz., the 8 bits corresponding to the function $f_{A,B,C}(x, y)$ for $y = \beta$, β^2, and β^4. Since in each segment the bit corresponding to $x = 0$ is 0, each segment may in fact be viewed as a 7-bit codeword with components indexed by the consecutive powers of a primitive root of $GF(8)$. Thus, for the "A, B, C" codeword in A_0', the y-segment's ith component is given by $f_{A,B,C}(\beta^i, y)$, where

$$\begin{aligned} f_{A,B,C}(x, y) &= \mathrm{Tr}\big(Ayx + (B + Cy)x^6\big) \\ &= (Ay)x + (A^2y^2)x^2 + (B^4 + C^4y^4)x^3 \\ &\quad + (A^4y^4)x^4 + (B^2 + C^2y^2)x^5 + (B + Cy)x^6 \end{aligned} \tag{B-10}$$

It follows that each 7-bit segment is a codeword in the $(7, 6)$ binary cyclic code with generator polynomial $g(x) = x - 1$. In particular, each segment has even weight. The value of the weights modulo 4 can be computed by a theorem of McEliece-Solomon [9, Theorem 1] or [1, Theorem 16.33], which says that if an even-weight binary vector $a = (a_0, \ldots, a_{n-1})$ is described by its Mattson-Solomon (MS) polynomial (discrete Fourier transform) $A(x) = A_0 + \cdots + A_{n-1}x^{n-1}$, i.e., if

$$a_i = \sum_{j=0}^{n-1} A_j \beta^{-ij} \tag{B-11}$$

where β is a primitive nth root of unity, and if $\Gamma_2(a) = \sum_{j=0}^{(n-1)/2} A_j A_{n-j}$, then $w(a) \equiv 2\Gamma_2(a) \pmod 4$. The MS polynomials for the 7-bit segments are given by Eq. (B-10), and so the value of Γ_2 for the y-segment is

$$\begin{aligned} \Gamma_2(y) &= (Ay)(B + Cy) + (A^2y^2)(B^2 + C^2y^2) \\ &\quad + (B^4 + C^4y^4)(A^4y^4) \\ &= (AB + A^4C^4)y + (A^2B^2 + AC)y^2 \\ &\quad + (A^4B^4 + A^2C^2)y^4 \\ &= \mathrm{Tr}\big((AB + A^4C^4)y\big) \end{aligned} \tag{B-12}$$

If the three segments are combined into one 21-bit word, the overall weight is still even, and the overall weight mod 4 is determined by the sum of the Γ_2s, viz.,

$$\begin{aligned} \Gamma_2 &= \Gamma_2(\beta) + \Gamma_2(\beta^2) + \Gamma_2(\beta^4) \\ &= \mathrm{Tr}\big((AB + A^4C^4)(\beta + \beta^2 + \beta^4)\big) \\ &= \mathrm{Tr}(0) = 0 \end{aligned} \tag{B-13}$$

Thus, each 7-bit segment has weight 0, 2, 4, or 6, and the overall weight is divisible by four. Furthermore, if one of the segments has weight zero, then by Lemma B2 either the other two segments are both zero, or else the other two segments have weight 4. It follows that the weights in the $(24, 9)$ code are 0, 8, 12, and 16. Now the original code A_0 is obtained from A_0' by complementing some or all of the segments, i.e., by replacing a segment of weight w with one of weight $8 - w$. Thus, in A_0, the segments have weight 0, 2, 4, 6, or 8. But since $8 - w \equiv w \pmod 4$, the weights in A_0 must also be divisible by four, and so in A_0 the only weights that can occur are 0, 4, 8, 12, 16, 20, and 24. The weight 4 can only occur as $0 + 0 + 4$ and $0 + 2 + 2$. Both of these cases can be eliminated by observing that since a zero-weight segment can only occur in an uncomplemented segment, and Lemma B2 says that if a codeword in A_0 has a zero-weight segment, then either the other two segments both have weight 8, or both have weight 4. Weight 20 is

ruled out by observing that the complement of a word of weight 20 is a word of weight 4. ❐

Lemma B5 says that the code A_0 is a $[24, 12, 8]$ binary linear code. According to MacWilliams and Sloane [6, Section 20.6], such a code must be equivalent to the Golay code. The next Lemma indicates that the code B_0 defined in Eq. (29) is also equivalent to the Golay code.

Lemma B6. The code B_0 defined in Eq. (29) is a $[24, 12, 8]$ code.

Proof: The proof is virtually the same as the proof of Lemma B5. The key difference is that in place of Eq. (B-10), one has for the code B_0

$$\begin{aligned} g_{D,E,F}(x,y) &= \mathrm{Tr}\big((Dy+E)x + (Fy)x^6\big) \\ &= (Dy+E)x + (D^2y^2+E^2)x^2 \\ &\quad + (F^4y^4)x^3 + (D^4y^4+E^4)x^4 \\ &\quad + (F^2y^2)x^5 + (Fy)x^6 \end{aligned} \tag{B-14}$$

so that in place of Eq. (B-12) one has

$$\begin{aligned} \Gamma_2(y) &= (Dy+E)(Fy) + (D^2y^2+E^2)(F^2y^2) \\ &\quad + (F^4y^4)(D^4y^4+E^4) \\ &= (EF+D^4F^4)y + (E^2F^2+DF)y^2 \\ &\quad + (E^4F^4+D^2F^2)y^4 \\ &= \mathrm{Tr}\big((EF+D^4F^4)y\big) \end{aligned} \tag{B-15}$$

and in place of Eq. (B-13) one has

$$\begin{aligned} \Gamma_2 &= \Gamma_2(\beta) + \Gamma_2(\beta^2) + \Gamma_2(\beta^4) \\ &= \mathrm{Tr}\big((EF+D^4F^4)(\beta+\beta^2+\beta^4)\big) \\ &= \mathrm{Tr}(0) = 0 \end{aligned} \tag{B-16}$$

Further details are omitted here. ❐

Lemma B7. The code A_1 defined in Eq. (30) is a $[24, 9, 8]$ code.

Proof: Just as in the proof of Lemma B5, the mapping from 6-tuples $[A, 0, C, \epsilon_0, \epsilon_1, \epsilon_2]$ to codewords in A_1 is a linear, one-to-one mapping, which implies that A_1 is a $(24, 9)$ code. Since A_1 is a subcode of A_0, its minimum distance must be ≥ 8. There are, however, many codewords in A_1 of weight 8, e.g., that obtained by taking $A = C = 0$, $\epsilon_0 = 1$, and $\epsilon_1 = \epsilon_2 = 0$. ❐

Lemma B8. The code A_3 defined in Eq. (31) is a $[24, 5, 12]$ code.

Proof: Using the formula Eq. (26) for $f_{A,B,C}(x,y)$, one finds that any codeword in A_3 can be represented as follows:

$$\begin{aligned} &\big[\mathrm{Tr}(C\beta x^6) + \epsilon_0 | \mathrm{Tr}(C\beta^2 x^6) + \epsilon_1 | \mathrm{Tr}(C\beta^4 x^6) \\ &\quad + (\epsilon_0+\epsilon_1)\big]_{x \in GF(8)} \end{aligned} \tag{B-17}$$

Two cases are considered: $C = 0$ and $C \neq 0$. If $C = 0$, then the codeword in Eq. (B-17) becomes $[\epsilon_0|\epsilon_1|\epsilon_0+\epsilon_1]$, which is either identically zero or has weight 16. If $C \neq 0$, then since there are exactly 4 elements in $GF(8)$ with trace 0, the codeword in Eq. (B-17) has weight 12. Thus, the only weights that occur in A_3 are 0, 12, and 16, and so A_3 is a $[24, 5, 12]$ code, as asserted. ❐

Lemma B9. The code A_4 defined in Eq. (32) is a $[24, 2, 16]$ code.

Proof: According to the definition in Eq. (32), each codeword in A_4 has three 8-bit segments. Either all three segments are identically zero, or else one segment is zero and the other two have weight 8. Thus, in A_4 the only weights that occur are 0 and 16, so that A_4 is a $[24, 2, 16]$ code, as asserted. ❐

Lemma B10. $A_0 \cap B_0 = A_1$.

Proof: Note that from Eq. (26) and Eq. (27), $f_{A,0,C}(x,y) = g_{A,0,C}(x,y)$. Thus, the code $A_0 \cap B_0$ contains the code A_1 as defined in Eq. (30). To prove the opposite inclusion, note that by the definitions Eq. (28) and Eq. (29) of A_0 and B_0, any word in the intersection will produce an equation of the form $f_{A,B,C}(x,\beta^{2^i}) + \epsilon_i = g_{D,E,F}(x,\beta^{2^i}) + \delta_i$ for all $x \in GF(8)$, for $i = 0, 1, 2$. By substituting $x = 0$ on both sides of this equation, one gets $\epsilon_i = \delta_i$, so that in fact, $f_{A,B,C}(x,\beta^{2^i}) = g_{D,E,F}(x,\beta^{2^i})$ for

all $x \in GF(8)$, for $i = 0, 1, 2$. By Lemma B4, $A = D$, $C = F$, $E = 0$, and $B = 0$, so that a word in the intersection must be of the form Eq. (30), i.e., it must lie in A_1. ❐

Lemma B11. $A_1 \cap B_1 = A_2$.

Proof: Given the definitions in Eq. (30) and Eq. (33) of A_1 and B_1, any word in the intersection $A_1 \cap B_1$ will produce an equation of the form $f_{A,0,C}(x, \beta^{2^i}) + \epsilon_i = g_{0,E,F}(x, \beta^{2^i}) + \delta_i$, for all $x \in GF(8)$ and $i = 0, 1, 2$. By substituting $x = 0$ on both sides of these equations, one gets $\epsilon_i = \delta_i$, so that in fact one has $f_{A,0,C}(x, \beta^{2^i}) = g_{0,E,F}(x, \beta^{2^i})$. By Lemma B4, this implies $A = E = 0$ and $C = F$. Thus, the intersection $A_1 \cap B_1$ is exactly the same as A_2, as defined in Eq. (34). ❐

Lemma B12. $A_2 \cap B_2 = A_3$.

Proof: Given Lemma B4 and the definitions Eq. (34) and Eq. (35) of A_2 and B_2, this result is immediate. ❐

Lemma B13. $A_2 \cap B_2' = A_3'$.

Proof: If a word in A_2, as defined in Eq. (34), is the same as a word in B_2', as defined in Eq. (39), then by setting $x = 0$, one finds that $\epsilon_0 = \delta_0$, $\epsilon_1 = \delta_1$, and $\epsilon_2 = \delta_2$. Thus, also $f_{0,0,C}(x, y) = g_{0,E,0}(x, y)$ for all $x \in GF(8)$ and $y = \beta, \beta^2, \beta^4$. It then follows from Lemma B4 that $C = E = 0$, and so a word in the intersection $A_2 \cap B_2'$ must be of the form described in Eq. (40). ❐

Lemma B14. $A_3' \cap B_3' = A_4$.

Proof: If a word in A_3', as defined in Eq. (40), is the same as a word in B_3', as defined in Eq. (41), then by setting $x = 0$, one finds that $\epsilon_0 = \delta_0$, $\epsilon_1 = \delta_1$, and $\epsilon_2 = \delta_0 + \delta_1$. Thus, also $f_{0,0,0}(x, y) = g_{0,e,0}(x, y)$ for all $x \in GF(8)$ and $y = \beta, \beta^2, \beta^4$. It then follows from Lemma B4 that $e = 0$, and so a word in the intersection $A_3' \cap B_3'$ must be of the form described in Eq. (32). ❐

References

[1] E. R Berlekamp, *Algebraic Coding Theory*, rev. ed., Laguna Hills, California: Aegean Park Press, 1984.

[2] I. N. Herstein, *Topics in Algebra*, 2nd ed., New York: Wiley and Sons, 1975.

[3] J. Justesen, E. Paaske, and M. Ballan, "Quasi-Cyclic Unit Memory Convolutional Codes," *IEEE Trans. Inform. Theory*, vol. IT-36, no. 3, pp. 540–547, May 1990.

[4] G. S. Lauer, "Some Optimal Partial-Unit-Memory Codes," *IEEE Trans. Inform. Theory*, vol. IT-25, no. 2, pp. 240–243, March 1979.

[5] L.-N. Lee, "Short Unit Memory Byte-Oriented Convolutional Codes Having Maximal Free Distance," *IEEE Trans. Inform. Theory*, vol. IT-22, no. 3, pp. 349–352, May 1976.

[6] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, Amsterdam: North-Holland, 1977.

[7] F. Pollara, R. McEliece, and K. Abdel-Ghaffar, "Finite-State Codes," *IEEE Trans. Inform. Theory*, vol. IT-34, no. 5, pp. 1083–1088, September 1988.

[8] F. Pollara, K.-M. Cheung, and R. J. McEliece, "Further Results on Finite-State Codes," *TDA Progress Report 42-92*, vol. October–December 1987, pp. 56–62, February 15, 1988.

[9] G. Solomon and R. McEliece, "Weights of Cyclic Codes," *J. Combinatorial Theory*, vol. 1, no. 4, pp. 459–475, December 1966.

[10] C. Thommesen and J. Justesen, "Bounds on Distances and Error Exponents of Unit Memory Codes," *IEEE Trans. Inform. Theory*, vol. IT-29, no. 5, pp. 637–649, September 1983.