In general, an \([n,k,d;m]\) convolutional code over a field \(F\) has generator matrix \(G(D) = G_0 + G_1 D + \cdots + G_K D^K\), where each \(G_i\) is a \(k \times n\) matrix with entries from \(F\). Here \(n\) is the branch length, \(k\) is the dimension per branch, \(m\) is the memory (i.e., the total number of nonzero rows in the matrices \(G_1, \ldots, G_K\)), and \(d\) is the free distance. Thus in this notation an \([n,k,d]\) block code is an \([n,k,d;0]\) convolutional code. A partial unit memory (PUM) convolutional code is one for which \(K = 1\) (hence the term “unit memory”) and at least one of the rows of \(G_1\) is zero (hence the term “partial unit memory.”) Indeed, if the first \(k \times m\) rows of \(G_0\) are all zero, then the resulting code is an \([n,k,d;m]\) PUM code.

In this paper we will give a general construction for partial unit memory convolutional codes. This construction may be used to design efficient finite state codes [2], [3]. Informally, the construction goes like this: Suppose \(C_0\) and \(C_0\) are two linear block codes of length \(n, n\) with \(C_0 < C_0\). Suppose \(C_0\) is an \([n,k^*,d^*]\) code, and \(C_0\) is a \([n,k,d]\) code. Then almost always we can combine these two codes to make a noncatastrophic partial unit memory convolutional code with parameters \([n,k,d^*\cdot k^*]\), where \(d^* \geq \min(d^*, 2d)\). Formally, the construction is described in the following theorem.

**Theorem 1.** Suppose that \(C_0\) is an \([n,k,d]\) linear block code, and that \(C_1\) is an \([n,k,d]\) linear block code, and \(C_0 \neq C_1\). Suppose further that \(C_0\) and \(C_1\) contain a common subcode \(C^*\) which is a \([n,k^*,d^*]\) code. Then there exists a noncatastrophic \([n,k^*,d^*, m]\) PUM convolutional code, with \(m = k - k^*\) and \(d^* \geq \min(d^*, 2d)\).

In applications, almost always (but not always) we only need two codes, \(C^*\) and \(C_0\). This is because as a rule the automorphism group of \(C^*\) will contain a permutation \(\sigma\) that does not fix \(C_0\), and we can take \(C^* = C_0 \sigma\) in Theorem 1. The following Corollary spells this out.

**Corollary 1.** Suppose that \(C_0\) is an \([n,k,d]\) linear block code, and that \(C^*\) is a \([n,k^*,d^*]\) code which is a subcode of \(C_0\). If the automorphism group of \(C^*\) contains a permutation \(\sigma\) that does not fix \(C_0\), then there exists a \([n,k,d; m]\) PUM convolutional code, with \(m = k - k^*\) and \(d^* \geq \min(d^*, 2d)\).

Theorem 1 and Corollary 1 permit us to construct a large number of PUM codes, many of which are optimal, in the sense of having the largest possible \(d_{max}\) for the given \(n, k, m\). Here are two examples.

**Example 1.** Let \(C^*\) be the \([8,1,8]\) binary repetition code, and let \(C_0\) be the \([8,4,8]\) extended Hamming code. The automorphism group of \(C^*\) is the symmetric group \(S_8\), which plainly does not fix \(C_0\). Thus Corollary 1 is applicable. The existence of a \([8,4,8]\) PUM code, which is optimal. This code was previously known (see e.g. [1]), but it is interesting to see how easily our construction finds it. It is also the inner code in the well-known Soviet “Regatta” system.

**Example 2.** Let \(C_0\) be the binary Golay \([24,12,8]\) code. It is possible to show that there is an isomorphic copy of \(C_0\), which we call \(C_1\), such that the dimension of the intersection \(C_0 \cap C_1\) is 9. This intersection contains both a \([24,5,12]\) code, and a \([24,2,16]\) code. Thus by Theorem 1 we can construct both a \([24,12,12; 7]\) PUM code, and a \([24,12,16; 10]\) PUM code, which are both optimal.

In the special case that \(C^*\) is the \([n,1,n]\) binary repetition code (as in Example 1), the automorphism group of \(C^*\) contains all permutations on \(\{1,2, \ldots, n\}\). Then unless \(k = 1, n - 1, n\), or \(n, C_0\) can’t be fixed by all such permutations. This leads to the following Corollary to Theorem 1.

**Corollary 2.** If \(C_0\) is a \([n,k,d]\) binary block code containing the all-ones vector, and if \(k \neq 1, n - 1, n\), then there exists a \([n,k,d,k - 1]\) PUM code with \(d \geq 2d_0\).

Corollary 2 naturally leads one to ask how large can \(d_0\) be, given that \(C_0\) contains the all-ones vector. We do not have a full answer to this question, but the following modification of the classic Greiner bound is useful.

Thus let \(N(k,d)\) denote the minimum length of a binary code with Hamming distance \(\geq d\) and dimension \(k\) which contains the all-ones vector.

**Theorem 2.** If \(k \geq 2\), then

\[ N(k,d) \geq d + N(k - 1, [d/2]) \]

**Corollary 3.** \(N(1,d) = d,\) and \(N(2,d) = 2d,\) and for \(k \geq 3,\)

\[ N(k,d) \geq d + [d/2] + [d/2^2] + \cdots + [d/2^{k-2}] + 2[d/2^{k-2}]. \]

Theorem 2 proves, for example, that there is no \([7,3,4]\) binary code containing the all-ones vector, although there is a \([7,3,2]\) code. Similarly, there is no \([20,5,9]\) linear code with the all-ones vector, although there is an \([21,5,9]\) such code. This is of interest, since Lauer [1] constructed a \([20,5,18; 4]\) PUM code, which therefore cannot be constructed by our methods. However, all of Lauer’s other codes, and many others scattered throughout the literature, can be constructed by our methods. Theorem 2 also raises the following question: Give a bound on the minimum distance of a linear block code that contains a known subcode. Except for the special case where the subcode is the repetition code, we know practically nothing about this question.

**Acknowledgements**

Abdel-Ghaﬀar’s contribution to this paper was supported by NSF Grant NCR 89-08105 and by an IBM Faculty Development Award. McEliece’s contribution was partially supported by a consulting contract with Caltech’s Jet Propulsion Laboratory under contract to the National Aeronautics and Space Administration. The research was also partially supported by AFOSR grant 88-0247 and a grant from Pacific Bell. Solomon’s contribution was supported by a consulting contract with Caltech’s Jet Propulsion Laboratory under contract to the National Aeronautics and Space Administration.

**References**

