SOME PARTIAL UNIT MEMORY CONVOLUTIONAL CODES

Khaled Abdel-Ghaffar
University of California
Davis, CA 95616

Robert McEliece
California Institute of Technology
Pasadena, CA 91125

Gustave Solomon
10747 Wilshire Blvd.
Los Angeles, CA 90024

Summary

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 a $[n, k, d; 0]$ 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 $k - m$ rows of G_1 are all zero, then the resulting code is a $[n, k, d; m]$ 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 $\mathcal{C}^*$ and $\mathcal{C}_0$ are two linear block codes of length n, with $\mathcal{C}^* \subseteq \mathcal{C}_0$. Suppose $\mathcal{C}^*$ is a $[n, k^*, d^*]$ code, and $\mathcal{C}_0$ is a $[n, k, d_0]$ code. Then almost always we can combine these two codes to make a noncatastrophic partial unit memory convolutional code with parameters $[n, k, d; k - k^*]$, where $d \geq \min(d^*, 2d_0)$. Formally, the construction is described in the following theorem.

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

In applications, almost always (but not always) we only need two codes, $\mathcal{C}^*$ and $\mathcal{C}_0$. This is because as a rule the automorphism group of $\mathcal{C}^*$ will contain a permutation π that does not fix $\mathcal{C}_0$, and we can take $\mathcal{C}_1 = \mathcal{C}_0^\pi$ in Theorem 1. The following Corollary spells this out.

Corollary 1. *Suppose that $\mathcal{C}_0$ is an $[n, k, d_0]$ linear block code, and that $\mathcal{C}^*$ is a $[n, k^*, d^*]$ 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 a $[n, k, d; m]$ PUM convolutional code, with $m = k - k^*$ and $d \geq \min(d^*, 2d_0)$.*

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_{free} for the given n, k, and m. Here are two Examples.

Example 1. Let $\mathcal{C}^*$ be the $[8, 1, 8]$ binary repetition code, and let $\mathcal{C}_0$ be the $[8, 4, 4]$ extended Hamming code. The automorphism group of $\mathcal{C}^*$ is the symmetric group S_8, which plainly does not fix $\mathcal{C}_0$. Thus Corollary 1 implies the existence of a $[8, 4, 8; 3]$ 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 concatenated "Regatta" system.

Example 2. Let $\mathcal{C}_0$ be the binary Golay $[24, 12, 8]$ code. It is possible to show that there is an isomorphic copy of $\mathcal{C}_0$, which we call $\mathcal{C}_1$, such that the dimension of the intersection $\mathcal{C}_0 \cap \mathcal{C}_1$ 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 $\mathcal{C}^*$ is the $[n, 1, n]$ binary repetition code (as in Example 1), the automorphism group of $\mathcal{C}^*$ contains all permutations on $\{1, 2, \ldots, n\}$. Then unless $k = 1$, $n - 1$, or n, $\mathcal{C}_0$ can't be fixed by all such permutations. This leads to the following Corollary to Theorem 1.

Corollary 2. *If $\mathcal{C}_0$ is a $[n, k, d_0]$ 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 $\mathcal{C}_0$ contains the all-ones vector. We do not have a full answer to this question, but the following modification of the classic Griesmer 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*

$$N(k, d) \geq d + N(k - 1, \lceil d/2 \rceil).$$

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

$$N(k, d) \geq d + \lceil d/2 \rceil + \lceil d/2^2 \rceil + \cdots + \lceil d/2^{k-3} \rceil + 2\lceil d/2^{k-2} \rceil.$$

Theorem 2 proves, for example, that there is no $[7, 3, 4]$ binary code containing the all-ones vector, although there is a $[7, 3, 4]$ 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, and also partially supported by AFOSR grant 88-0247 and a grant from Pacific Bell. Solomon's contribution was supported by a consuling contract with Caltech's Jet Propulsion Laboratory under contract to the National Aeronautics and Space Administration.

References

[1] Lauer, G. S., "Some Optimal Partial-Unit-Memory Codes," *IEEE Trans. Inform. Theory* vol. IT-25 (March 1979), pp. 240–243.

[2] Pollara, F., McEliece, R., and Abdel-Ghafar, K., "Finite-State Codes," *IEEE Trans. Inform. Theory* vol. IT-34 (September 1988), pp. 1083–1088.

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