Quasi-cyclic unit memory convolutional codes

Justesen, Jørn; Paaske, Erik; Ballan, Mark

Published in:
I E E E Transactions on Information Theory

Link to article, DOI:
10.1109/18.54876

Publication date:
1990

Document Version
Publisher's PDF, also known as Version of record

Quasi-Cyclic Unit Memory Convolutional Codes

JØRN JUSTESEN, ERIK PAASKE, MARK BALLAN

Abstract—**Unit memory convolutional codes with generator matrices, which are composed of circulant submatrices, are introduced. This structure facilitates the analysis as well as an efficient search for good codes. Equivalences among such codes and consider some of the basic structural properties are discussed. In particular, catastrophic encoders and minimal encoders are characterized and dual codes treated. Further, various distance measures are discussed and a number of good codes, some of which result from efficient computer search and some that result from known block codes are presented.**

I. Introduction

UNIT memory convolutinal codes (UMC) were introduced by Lee [1], who also showed that they can have larger free distance d_∞ than the usual multimemory convolutional codes (MMC) with the same rate and the same number v of memory elements in the encoder. The distance properties of UMC were further studied in [2], where upper and lower bounds were derived. These bounds indicate that in many cases of interest, UMC may be expected to have superior properties. However, so far, few specific UMC have been investigated [1], [3] and most of them have a very irregular structure.

Exhaustive computer search for good codes becomes practically impossible even for moderate values of the number v of delay elements. Standard methods for analyzing structural properities of encoders are often very difficult for UMC. We suggest a class of codes that is defined by generator matrices composed of circulant submatrices. In analogy with a well-known class of block codes, we refer to these codes as quasi-cyclic UMC (QCUMC). The structure of these codes facilitates the analysis as well as an efficient search for good codes, and we shall present the results of a search with block lengths up to 32.

In Section II we give a definition of the class of QCUMC. We discuss equivalences among such codes and selection criteria for good codes. Recent work has shown that the selection of good codes cannot be based solely on d_∞, but that several distance parameters should be evaluated and compared. Examples of such a broader selection of criteria for MMC can be found in [4], [5], and the importance of the extended row distance was pointed out in [2]. We extend this earlier work on selection methods and bounds for certain distances. Section III contains an algebraic analysis which exploits the circulant structure. In Section IV we describe a computer search for good rate 1/2 QCUMC and present tables of these codes. This search also produced a couple of block codes with better minimum distances than any previously reported. Furthermore, we present some QCUMC that are mainly derived from known block codes.

Manuscript received July 1986 and March 30, 1989; revised July 17, 1989. This work was presented in part at the IEEE International Symposium on Information Theory, Les Arcs, France, June 21–25, 1982.

J. Justesen and E. Paaske are with the Institute of Circuit Theory and Telecommunication, Technical University of Denmark, DK-2800 Lyngby, Denmark.

M. Ballan was with the Institute of Circuit Theory and Telecommunication, Technical University of Denmark. He is now with Telecon Denmark, Telegade 2, DK-2630 Taastrup, Denmark.

IEEE Log Number 8932923.

II. Quasi-cyclic Unit Memory Codes

A. Definition

A unit memory encoder is defined by an encoding rule of the form

$$\boldsymbol{y}_j = \boldsymbol{x}_j \boldsymbol{G}_0 + \boldsymbol{x}_{j-1} \boldsymbol{G}_1$$

where the x_j are k-bit binary information vectors, and the y_j are n-bit binary vectors of encoded symbols. The matrices $\boldsymbol{G}_0$ and $\boldsymbol{G}_1$ are $k \times n$, and we assume that $\boldsymbol{G}_0$ has rank k. The rate of the code is $R = k/n$.

The UMC is the set of sequences that can be generated by the encoder. Many different encoders may generate the same code, and we shall have to distinguish between structural properties of the code and properties of a specific encoder.

A quasi-cyclic (QC) unit memory encoder is an encoder where $\boldsymbol{G}_0$ and $\boldsymbol{G}_1$ are composed of nontrivial circulant $m \times m$ matrices, i.e., matrices where the ith row equals the top row cyclically shifted $i-1$ times. We shall represent a circulant matrix $\boldsymbol{A}$ by a polynomial

$$a(x) = a_0 + a_1 x \cdots + a_{m-1} x^{m-1} \tag{1}$$

where $(a_0, a_1, a_2, \cdots, a_{m-1})$ is the top row of $\boldsymbol{A}$. We note that the rank of $\boldsymbol{A}$ is m if, and only if

$$gcd\,(a(x), x^m + 1) = 1.$$

The most interesting codes have parameters k and n that are small multiples of m, and we shall give particular attention to the case $(n,k) = (2m, m)$. We shall call a UMC quasi-cyclic if it has a QC encoder. It may occur that a QC code does not have a minimal encoder [6] in

0018-9448/90/0500-0540$01.00

QC form. We shall return to the question of minimal encoders in Section III. In this paper we shall also restrict ourselves only to encoders where $\boldsymbol{G}_0$ can be put in a form that includes a $k \times k$ identity matrix, i.e.,

$$\boldsymbol{G}_0 = \{\boldsymbol{E}, \boldsymbol{A}\} \tag{2}$$

where $\boldsymbol{E}$ is a $k \times k$ identity matrix. We notice, however, that QCUMC exist that cannot be put in this form. Consider for example the code generated by a 7×14 matrix $\boldsymbol{G}_0 = \{\boldsymbol{A}_1, \boldsymbol{A}_2\}$ where $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are specified by

$$\{x^3 + x + 1, x^3 + x^2 + 1\}.$$

Since neither $\boldsymbol{A}_1$ nor $\boldsymbol{A}_2$ has an inverse, $\boldsymbol{G}_0$ cannot be put in the form (2).

To some extent our choice of the QC structure has been motivated by the success of putting generator matrices of good block codes in QC form [7], [8]. Thus some known results about good block codes can be used as a basis for developing UMC. It might appear even more attractive to use cyclic block codes as a basis, and in fact this approach has been followed by Piret [9]. However, so far there has been little progress in algebraic analysis of distances of such codes. On the other hand, many cyclic codes can be put into QC form, and for this reason a few of our codes are closely related to codes mentioned in [9].

B. Equivalences Among Encoders and Codes

We shall take advantage of the QC structure by noting a number of equivalences between encoders and codes. Consistent with common practice two encoders are considered to be equivalent if they generate the same code. Having restricted ourselves to the form (2) we are left with only one encoder for $k = m$, but for $k \geq 2m$ there are several equivalent encoders if the rank of $\boldsymbol{G}_1$ is $< k$. Consider for example the encoder

$$\boldsymbol{G}_0 = \begin{bmatrix} \boldsymbol{E} & \boldsymbol{0} & \boldsymbol{A}_1 \\ \boldsymbol{0} & \boldsymbol{E} & \boldsymbol{A}_2 \end{bmatrix} \qquad \boldsymbol{G}_1 = \begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{B}_2 & \boldsymbol{B}_3 \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$$

where $\boldsymbol{E}$ is now a $k/2 \times k/2$ identity matrix. In this code the matrix $\{\boldsymbol{0}\ \boldsymbol{E}\ \boldsymbol{A}_2\}$ spans a vectorspace of codewords of length one block. This matrix may be added to $\boldsymbol{G}_1$ to produce an equivalent QC encoder with unit memory.

We shall say also that two encoders are equivalent if they generate codes that differ only by a permutation of the positions. This terminology is common practice for block codes, but is not often used for convolutional codes. First we note that there are several permutations within each block that preserve the QC structure. If the same permutation is applied to each block, we obviously get an equivalent code. A number of permutations that preserve the QC structure are listed in [7]. We may use the equivalences to select a single representative for each set of equivalent (n, k) block codes, and use the generator matrix of this code as the matrix $\boldsymbol{G}_0$.

In addition to these equivalences, there are permutations that apply to convolutional codes, but do not affect the related block codes. If π is a permutation of positions in each block that preserves the block codes spanned by $\boldsymbol{G}_0$ and $\boldsymbol{G}_1$, and $\pi\boldsymbol{G}_1$ denotes the matrix obtained by permuting $\boldsymbol{G}_1$, then the encoders

$$\{\boldsymbol{G}_0, \boldsymbol{G}_1\} \quad \text{and} \quad \{\boldsymbol{G}_0, \pi\boldsymbol{G}_1\}$$

generate equivalent convolutional codes. In particular, we may take a cyclic shift of j positions in each circulant of $\boldsymbol{G}_1$ and obtain an equivalent code. A permutation of this type will produce a periodic permutation of the encoded sequence where the kth block is permuted by π^k.

Example 1: We shall consider (14,7) QCUMC with generator matrices

$$\boldsymbol{G}_0 = \{\boldsymbol{E}, \boldsymbol{A}_1\} \quad \text{and} \quad \boldsymbol{G}_1 = \{\boldsymbol{A}_2, \boldsymbol{A}_3\}$$

where $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{A}_3$ are 7×7 circulant matrices. There are a total of 2^7 7×7 circulant matrices. Excluding the trivial all-zero and all-one matrices these can be divided into 18 classes, where each class consists of 7 matrices that are equivalent under cyclic shifts. The equivalence

$$a(x) \approx a(x^j) \bmod (x^m + 1), \qquad j = 1, 2, \cdots, m-1$$

reduces the set of inequivalent matrices to 8. If $\boldsymbol{A}$ is invertible, we may interchange $\boldsymbol{E}$ and $\boldsymbol{A}_1$ and multiply by $\boldsymbol{A}^{-1}$ to obtain an equivalent code. In this way we are left with six inequivalent (14,7) codes.

For a fixed $\boldsymbol{G}_0$, $\boldsymbol{G}_1$ may be selected in 2^{14} ways. If we exclude those $\boldsymbol{G}_1$ that include an all-zero or all-one matrix, the remaining matrices generate $(128-2)(128-2)/7 = 2268$ inequivalent convolutional codes.

C. Distance Measures

A number of distance measures have been proposed for convolutional codes. We shall define the order j column distance, d_j^c to be the minimum weight of all paths in the trellis having a nonzero first branch and truncated after $j+1$ branches, and the extended row distance, $\hat{d}_j^r$, to be the minimum weight of all paths in the trellis having a nonzero first branch and returning for the first time to the zero state after $j+1$ branches. As shown in [2] the sequence of extended row distances is most closely related to the performance of the code, and the minimum value of this sequence d_∞ is generally recognized as the single most important parameter and called the free distance. However, as was pointed out by Hemmati and Costello [10], it is important that $\hat{d}_j^r$ increases sufficiently quickly with j, and in particular that the minimum average linear growth rate for large j, w_0, which was originally defined by Huth and Weber [11], is large. For some applications it is also important that the column distances d_j^c grow as rapidly as possible [12].

Upper bounds on d_∞ may be obtained from known bounds on minimum distances $d(n, k)$ of block codes that correspond to terminated convolutional codes [1], [2], [3] such that

$$d_\infty \leq \min_{j \geq 1} \{d((j+1)n, jk)\}. \tag{3}$$

Upper bounds on column distances were discussed in [2] but the bounds are usually not tight.

As regards $\hat{d}_j^r$, which for large j is determined by w_0, tight bounds seem difficult to obtain. Huth and Weber [11] give bounds for specific $(n,1)$ MMC, and in particular they noted that the best values of w_0 and d_∞ that can be achieved for some fixed code parameters, can often not be achieved with the same code. Hemmati and Costello [10] gave an improved upper bound, but only for a certain class of (2,1) codes, and in general there has been little work in this direction. Nevertheless, it may be reasonable to compare $\hat{d}_j^r$ to $d((j+1)n, jk)$. In particular one might for large j use the Gilbert bound and require that for good codes $w_0 > nH^{-1}(1-R)$, where H^{-1} is the inverse binary entropy function.

As appears from the previous discussion the choice of criteria to select good codes is not obvious, but we believe that large d_∞ should still be considered as the most important first criterion. The second criterion would naturally be the sequence of extended row distances $\hat{d}_j^r$ but for small values of j, $\hat{d}_j^r$ equals d_∞ or increases slowly, and for that reason this sequence is not a useful criterion in a computer search for good codes. On the other hand, a rapid growth of column distances d_j^c is also important and will in most cases imply a fast growth in $\hat{d}_j^r$. Furthermore, it is a criterion that can be used for fast rejection of codes in a computer search. We have, therefore, searched for codes with the largest d_∞, and among these we have preferred codes where d_j^c increase quickly to d_∞. Finally, we have calculated $\hat{d}_j^r$ and only selected codes that have an acceptable growth rate, i.e., $w_0 > nH^{-1}(1-R)$.

III. Algebraic Structure

In this section we shall consider some of the basic structural properties of UMC, and we shall point out how the QC structure facilitates the analysis. In particular, we shall characterize catastrophic encoders and minimal encoders, and we shall consider dual codes. Standard methods for deciding properties of convolutional codes are often not practical when the block length is large, and we shall use methods that are particularly suited for UMC.

In the analysis of QC block codes, it was noted in [8] that the ring of circulant $m \times m$ matrices over the binary field, F, is isomorphic to the ring of polynomials of degree less than m

$$R_m = F[x]/(x^m+1).$$

Using this isomorphism, we may replace the binary $k \times n$ matrix $\boldsymbol{G}$, composed of circulant matrices, by a $k/m \times n/m$ matrix, $\hat{\boldsymbol{G}}$ over R_m. The set of vectors spanned by $\hat{\boldsymbol{G}}$ is a module over R_m. Many calculations are greatly simplified if they can be performed on this compressed matrix, but unfortunately some important results do not apply to modules over rings with zero divisors. In particular concepts like dimension and dual spaces are more difficult to apply, and there is no equivalent of the invariant factor theorem, which is important in the analysis of convolutional codes [6].

If G has QC form and generates a binary (n,k) code, the dual code is also QC. If $\boldsymbol{H}$ is a QC parity check matrix,

$$\boldsymbol{G}\boldsymbol{H}^T = \boldsymbol{0}$$

we can calculate the product in R_m

$$\hat{\boldsymbol{G}}\hat{\boldsymbol{H}}^T = \boldsymbol{0}$$

provided that we let the indeterminate x represent a shift to the right in $\boldsymbol{G}$, but a left shift in $\boldsymbol{H}$. In this way, each block of m parity checks in F is replaced by a single parity check in R_m. This interpretation of parity check polynomials is well known from cyclic codes.

The rank of a single $m \times m$ circulant matrix, $\boldsymbol{A}$ may be found as

$$rk(\boldsymbol{A}) = m - \deg\{gcd[x^m+1, a(x)]\} \tag{4}$$

Again, this fact is well-known from the theory of cyclic codes. Since x^m+1 always factors into at least two binary factors, there are always zero divisors in R_m.

In order to calculate the rank of matrices composed of several circulant submatrices and find dual codes, we shall use a reduction of matrices over R_m to echelon form.

Lemma 1: A QC matrix $\boldsymbol{G}$ consisting of circulant submatrices $\boldsymbol{g}_{ij}$, may be reduced to a QC matrix with the same rank in a standard form by row operations on $\hat{\boldsymbol{G}}$. The reduced matrix has the following properties.

- If $\boldsymbol{g}_{ij}$ is the leading nonzero circulant in the i th row, then $\boldsymbol{g}_{mn}$ $m > i$ and $n \le j$, are zero matrices (all zero rows are eliminated).
- The rank of $\boldsymbol{G}$ equals the sum of the ranks of the leading nonzero matrices in the rows.

Proof: We give a proof by induction on the number of columns in $\hat{\boldsymbol{G}}$. Assume that there is only one column:

$$\hat{\boldsymbol{G}} = (a_{11}, a_{21}, \cdots, a_{r1})^T.$$

By suitable row operations we can make the first row equal

$$(a(x), 0, 0, \cdots, 0)$$

where

$$a(x) = gcd\{x^m+1, a_{11}(x), \cdots, a_{r1}(x)\}$$

and the remaining rows can then be reduced to zero and eliminated. The rank may be found from (4) in agreement with the lemma. Now assume that the lemma is true for a number of columns less than c, and consider a matrix with c columns. We may find the first nonzero column, and as in the case of a single column, the first element is made equal to $a(x)$ while the others are reduced to zero. The new first row is now multiplied by $(x^m+1)/a(x)$, and we obtain a new row where the first element is zero. If the new row is all zeros, the rank of the binary image of the first row equals the rank of the leading nonzero circulant. If the new row is nonzero, the rank of the binary image of the first row is the sum of the ranks of the leading circulant and the rank of the image of the new row. In this case we add the new row as the second row of the matrix. If the first row is eliminated from the reduced

matrix, the remaining matrix may be put into echelon form by the induction hypothesis, and the rank satisfies the lemma.

Clearly the calculations are complicated by the zero divisors, and there are extra difficulties when m is even. On the other hand, when x^m+1 has few factors, or the leading matrices are chosen to be nonsingular, the calculations proceed almost as in a field. In addition we shall be most interested in matrices consisting of few circulants, which also facilitates the calculations.

In order to find a parity check matrix for a given code or, equivalently, a generator matrix for the dual code, we may solve a system of linear equations over F. In most situations of practical interest, this system can be replaced by a small system over R_m, and a solution may be readily obtained using the reduction method of Lemma 1. We shall not discuss the modifications that are needed in order to handle singular leading circulants. □

Catastrophic UMC are characterized by the following theorem:

Theorem 1: A UM encoder $(\boldsymbol{G}_0, \boldsymbol{G}_1)$ is catastrophic if, and only if, there exists an $s \times k$ matrix $\boldsymbol{P}$ of rank s, $s > 0$, and a nonsingular $s \times s$ matrix $\boldsymbol{Q}$ such that

$$\boldsymbol{QPG}_0 = \boldsymbol{PG}_1.$$

Proof: If the condition is satisfied, the set of critical infinite input sequences

$$\boldsymbol{IP}, \boldsymbol{IQP}, \boldsymbol{IQ}^2\boldsymbol{P}, \cdots$$

where $\boldsymbol{I}$ is an arbitrary nonzero vector, are encoded as codewords with only the first block nonzero. The matrices $\boldsymbol{PG}_0$ and $\boldsymbol{PG}_1$ span the same vector space with dimension > 0. The existence of such a common subspace for the vector spaces spanned by $\boldsymbol{G}_0$ and $\boldsymbol{G}_1$ is necessary if the encoder is to produce a zero output on a nonzero input. An infinite zero output can occur only if the subspace is produced on the same set of input sequences.

If $(\boldsymbol{G}_0, \boldsymbol{G}_1)$ has QC form, then $\boldsymbol{P}$ and $\boldsymbol{Q}$ may also be expressed in this form (possibly with some redundant rows in $\boldsymbol{P}$). Thus the necessary calculations can be performed in R_m. □

The following approach may be used to decide whether a given encoder is catastrophic: Calculate the sequence of matrices $\boldsymbol{P}_1, \boldsymbol{P}_2, \cdots$, such that $\boldsymbol{P}_1\boldsymbol{G}_0$ spans the intersection of the subspaces spanned by $\boldsymbol{G}_0$ and $\boldsymbol{G}_1$, and $\boldsymbol{P}_j\boldsymbol{G}_0$ spans the intersection of the subspaces spanned by $\boldsymbol{P}_{j-1}\boldsymbol{G}_0$ and $\boldsymbol{P}_{j-1}\boldsymbol{G}_1$. Thus, in each step the rank of $\boldsymbol{P}_j$ decreases until either a zero matrix or a matrix satisfying the conditions of Theorem 1 is obtained.

Theorem 2: A noncatastrophic UM encoder

$$\{\boldsymbol{G}_0, \boldsymbol{G}_1\} = \begin{Bmatrix} \boldsymbol{G}_0' & \boldsymbol{O} \\ \boldsymbol{G}_0'' & \boldsymbol{G}_1'' \end{Bmatrix}$$

where $\boldsymbol{G}_0''$, $\boldsymbol{G}_1''$ have v rows, in minimal if, and only if,

$$rk\begin{Bmatrix} \boldsymbol{G}_1'' \\ \boldsymbol{G}_0' \end{Bmatrix} = k.$$

Proof: The encoder is minimal if, among encoders for the same code, $\boldsymbol{G}_1$ has the smallest number of nonzero rows. We may add rows from $\boldsymbol{G}_0'$ to $\boldsymbol{G}_1''$ without changing the code. Further we note that since the encoder is assumed to be noncatastrophic, the last nonzero block in a codeword is spanned by $\boldsymbol{G}_0'$ and $\boldsymbol{G}_1''$. Since the rows of these two matrices are linearly independent, the dimension of the codewords with only the first block nonzero cannot exceed $k - v$. Thus the condition is sufficient. If the condition is not satisfied, we may perform row operations on the top v rows and add rows from $\boldsymbol{G}_0'$ to $\boldsymbol{G}_1''$ to reduce the number of nonzero rows. Consequently the condition is necessary. □

When the encoder has the form used in Theorem 2, we shall say that the memory is v bits. Since v is not necessarily a multiple of m, it may not be possible to write a minimal encoder in QC form unless we allow the matrices to have more than k rows. However, if we include the extra rows, a minimal encoder can be expressed in QC form.

It is sometimes helpful to use the familiar D notation to compress the generator matrix into a single $k \times n$ matrix

$$\hat{\boldsymbol{G}} = \hat{\boldsymbol{G}}_0 + D\hat{\boldsymbol{G}}_1.$$

In particular it is often easier to find the parity check matrix for a UMC from this representation.

In the construction of QC block codes, an extra row and column of ones is often added to the generator matrix. The extra row may be interpreted as representing m identical rows, and the need for such redundant rows has already been discussed. We may fit blocks of a length m' that divides m into the analysis, provided that m/m' is odd. The calculations in R_m are carried out as though the short block were repeated to make a circulant $m \times m$ matrix. In calculating scalar products, we get an odd number of repeated terms from such blocks, but the result is not changed. In particular a column of ones may be treated as an all ones matrix when m is odd.

IV. Search for and Derivation of Good QCUMC

A. Search for (2m, m) Codes

Regardless of the fact that one can take great advantage of the QC structure in a computer search for good UMC's, the computational work involved depends heavily on the criteria used to select good codes. As mentioned in Section II, we have chosen the following criteria:

1) optimum free distance, d_∞,
2) optimum distance profile $\boldsymbol{d} = [d_0^c, d_1^c, \cdots, d_\infty]$ conditioned on 1),

3) check for "acceptable" extended row distance growth.

As mentioned in Section II we consider d_x to be the most natural first criterion, but another good reason not to choose $\boldsymbol{d}$ as the first criterion is that $\boldsymbol{d}$ must be calculated by a unidirectional search, while d_x can be calculated by a bidirectional search [13], [14] which e.g., for the (32,16) code is about 50 times faster. By an "acceptable" extended row distance growth we mean that only codes where the minimum average asymptotic growth rate w_0 exceeds the Gilbert bound are accepted. In other words, we require $w_0 > nH^{-1}(1-R) = 0.11n$.

We have used a computer to search for good $(2m, m)$ codes with $m \le 16$. However, we have not performed an exhaustive search, but have limited the search to encoders with a value of d_0^c which is optimum or one below the optimum value, except for the (24,12) code where the only optimal candidates for $\boldsymbol{G}_0$ are the two quasi-cyclic versions of the Golay code, and there are no candidates with $d_0^c = 7$. Thus, in this case we have also searched for encoders with $d_0^c = 6$. The main reason not to perform an exhaustive search among all QCUMC is that the changes of improving the results are considered small compared to the cost.

To reduce the computational work further, we used the equivalences mentioned in Section II and the block code upper bound in (3), whereby we achieve a substantial reduction in the number of QCUMC for which d_x must be evaluated.

We shall point out next that a very important factor in reducing the computational work lies also in the choice of the QC structure, since we can exploit this structure in the evaluation of the distance measures. To see this we first notice that evaluation of d_j^c and $\hat{d}_j^r$ (including d_x) is, in principle, based on evaluation of minimum weight paths diverging from the all-zero state with a nonzero first input and ending at any state in the trellis for d_j^c and the all-zero state for $\hat{d}_j^r$. Thus, it is important to reduce the number of minimum weight paths that has to be evaluated. For the $(2m, m)$ QCUMC this is done by dividing the encoder states into cyclic subsets $C_0, C_1, \cdots, C_{\tau-1}$ in such a way that two states belong to the same cyclic subset if one is simply a cyclic shift of the other. An example is shown in Fig. 1 for the trellis of the (6,3) code specified in Table I. If we now let $d_l(i)$ denote the Hamming weight of the minimum weight path diverging from the all-zero state with a nonzero first input and ending at state i in level l, then $d_l(i) = d_l(j)$ if i and j belong to the same cyclic subset. This is seen by observing that if $\{\boldsymbol{u}_0, \boldsymbol{u}_1, \cdots, \boldsymbol{u}_{l-1}\}$ is an input sequence that generates a minimum weight path to some state in a cyclic subset C_a, then minimum weight paths to all states in C_a are generated by the input sequences $\{\boldsymbol{u}_0^t, \boldsymbol{u}_1^t, \cdots, \boldsymbol{u}_{l-1}^t\}$, $0 \le t \le m-1$, where $\boldsymbol{u}^t$ denotes $\boldsymbol{u}$ cyclically shifted t times.

This means that calculation of distance measures does not require evaluation of minimum weight paths to *all* states in the trellis, but only to *one* state in each cyclic subset, and thus the computational load is reduced by a factor $2^m/\tau$ [16], where

$$\tau = \frac{1}{m} \sum_{(m/a) \in N} \Phi(a) 2^{m/a}$$

and $\Phi(a)$ is Euler's Phi-function.

We present the codes in Table I and Table II. In Table I we list the first row of $\boldsymbol{G}_0$ and $\boldsymbol{G}_1$ in octal form. The binary translation of the octal number is truncated to n digits, e.g., for $n = 10$, the first row of $\boldsymbol{G}_0 = 4170$, and the entire matrix becomes

$$\boldsymbol{G}_0 = \begin{Bmatrix} 10000 & 11110 \\ 01000 & 01111 \\ 00100 & 10111 \\ 00010 & 11011 \\ 00001 & 11101 \end{Bmatrix},$$

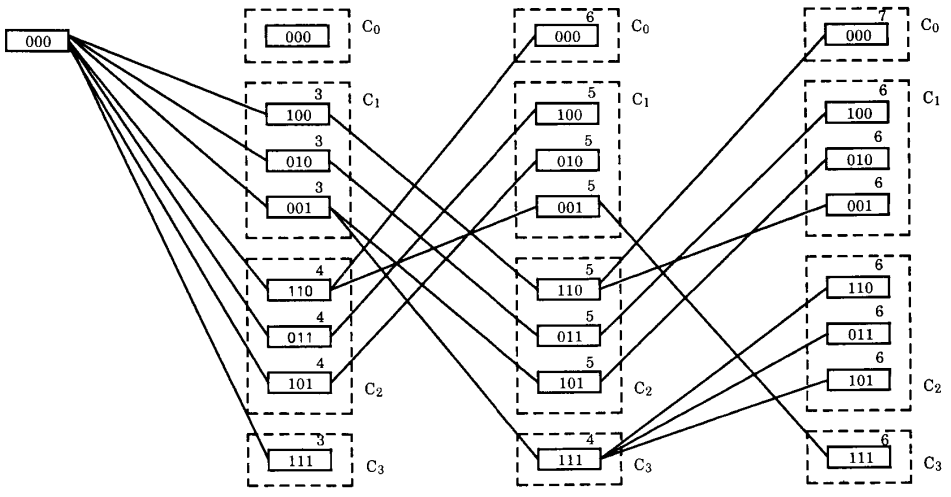

Fig. 1. Minimum weight paths in trellis for (6,3) QCUMC in Table I.

TABLE I
QCUMC WITH $R = \frac{1}{2}$

n	G_0	G_1	d_0^c	d_1^c	d_2^c	d_3^c	d_4^c	d_∞	d_∞(MMC)	d_∞(UB)
4*	50	54	2	3	4	5		5	5	5
6*	43	74	3	4	6			6	6	6
8	430	466	3	5	6	7	8	8	7	8
10	4170	7130	4	6	7	9		9	8	9
12*	4027	6061	4	6	8	10		10	10	10
14	40036	73114	4	7	9	11	12	12	10	12
16*	400134	510474	5	8	10	12		12	12	13
18	400027	524155	5	8	11	13	14	14	12	15
20	4000476	7520054	6	9	12	14	16	16	14	16
22	40001334	72002270	7	10	13	15	16	16	15	16
24	40003367	52546076	6	10	13	16	17	17	16	19
26	400000372	735530474	6	10	13	16	19	19	16	20
28	4000002364	6614137364	7	11	14	18	20	20	18	22
30 [1]	4000002167	6534041701	8	12	15	19	20	20	19	24
32 [1]	40000000656	67756145026	7	11	15	18	22	22	20	24

[1] The search for a code with optimal $\boldsymbol{d}$ was not exhaustive and thus a slightly better code may exist.

TABLE II
EXTENDED ROW DISTANCES FOR QCUMC IN TABLE I

n	$\hat{d}_0^r$	$\hat{d}_1^r$	$\hat{d}_2^r$	$\hat{d}_3^r$	$\hat{d}_4^r$	$\hat{d}_5^r$	$\hat{d}_6^r$	$\hat{d}_7^r$	$\hat{d}_8^r$	$\hat{d}_9^r$	$\hat{d}_{10}^r$	$\hat{d}_{11}^r$	$\hat{d}_{12}^r$	$\hat{d}_{13}^r$	$\hat{d}_{14}^r$	$\hat{d}_{15}^r$	$\hat{d}_{16}^r$	$\hat{d}_{17}^r$	$\hat{d}_{18}^r$	$\hat{d}_{19}^r$	$\hat{d}_{20}^r$	w_0	GB[1]
4	5	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	1.00	0.44
6	6	7	8	10	11	12	14	15	16	18	19	20	22	23	24	26	27	28	30	31	32	1.33	0.66
8	8	8	10	12	12	14	16	16	18	20	20	22	24	24	26	28	28	30	32	32	34	1.33	0.88
10	9	10	11	13	15	16	18	20	21	23	25	26	28	30	31	33	35	36	38	40	41	1.66	1.10
12	10	12	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	1.96	1.32
14	12	12	13	15	18	19	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	2.00	1.54
16	12	13	15	17	19	22	25	26	29	31	33	36	38	40	43	45	47	50	52	54	57	2.33	1.76
18	14	14	18	20	22	24	26	30	32	34	36	38	42	44	46	48	50	54	56	58	60	2.40	1.98
20	16	16	18	20	24	26	30	32	34	36	40	42	44	48	50	52	54	58	60	62	66	2.57	2.20
22	16	16	20	22	24	28	30	32	36	38	40	42	46	48	50	52	56	58	60	62	66	2.50	2.42
24	17	17	20	24	26	29	32	35	38	41	44	47	50	53	56	59	62	65	68	71	74	3.00	2.64
26	19	19	22	24	26	31	34	37	40	43	46	49	52	56	58	61	65	67	71	75	77	3.17	2.86
28	20	20	22	26	28	32	36	38	42	46	48	52	56	60	62	66	70	72	76	80	82	3.17	3.08
30	20	21	24	28	31	36	39	42	46	50	54	57	60	64	68	71	75	78	82	85	89	3.50	3.30
32	22	22	26	30	34	38	42	46	50	54	58	62	66	70	74	78	82	86	90	94	98	4.00	3.52

[1] GB is the Gilbert bound on block codes corresponding to terminated convolutional codes, i.e., $nH^{-1}(1-R)$.

d_∞(MMC) is the largest value of d_∞ for any known $R = 1/2$MMC with k memory elements. d_∞(UB) is the block code upper bound.

The (8,4) UMC found by Lee [1] and Lauer [3] by far outperforms the (8,4) QCUMC in Table I, since their UMC has $\boldsymbol{d} = \{4,6,8\}$ and only 8 encoder states while our encoder has $\boldsymbol{d} = \{3,5,6,8\}$ and 16 encoder states. There is no direct QC form of the Lee/Lauer code, but it may be put into a form consisting of $m = 3$ circulant matrices with an extra row and column added such that

$$\boldsymbol{G}_0 \begin{Bmatrix} 111 & 1 & 111 & 1 \\ 100 & 0 & 110 & 1 \\ 010 & 0 & 011 & 1 \\ 001 & 0 & 101 & 1 \end{Bmatrix} \qquad \boldsymbol{G}_1 = \begin{Bmatrix} 000 & 0 & 000 & 0 \\ 110 & 0 & 100 & 1 \\ 011 & 0 & 010 & 1 \\ 101 & 0 & 001 & 1 \end{Bmatrix}.$$

For the values of n marked with an asterisk there exist MMC with the same value of d_∞, but for all other values of n, the encoders in Table I are—to our knowledge—better than any previously known MMC or UMC with the same number of states, either because they achieve a larger free distance or a better distance profile.

It is also worth noting that for the (48,12) and (52,13) block codes our search improved earlier results from the tables in [7] on minimum distances. As reported in [17], our corresponding $\boldsymbol{G}_0\boldsymbol{G}_1$ codes have minimum distances 17 and 19, respectively.

We shall further mention that all QCUMC in Table I are noncatastrophic encoders, which is always the case when $\boldsymbol{G}_0$ and $\boldsymbol{G}_1$ generate codes that have only the all-zero codeword in common.

In Table II we have listed the extended row distances $\hat{d}_j^r$ for $j \le 20$, w_0, and the Gilbert bound $(H^{-1}(1-R))n = 0.11n$. We have calculated $\hat{d}_j^r$ for $j \le 28$ and in most cases minimum average weight cycles resulting in w_0 can be found from the minimum weight paths we evaluate to determine $\hat{d}_j^r$. For some of the codes with large n there

might, nevertheless, be a very long cycle with a smaller w_0 that has not appeared yet. Thus an exhaustive search for average minimum weight cycles may in a few cases reduce our values of w_0 slightly. An example with one of the smaller codes is the (12,6) code that has a 3 branch cycle of weight 6, but also a 24 branch cycle of weight 47.

B. Codes of Rate $> 1/2$

UMC's with high rates are particularly interesting because a low redundancy is often desirable in applications and because these codes have good free distances [2]. The analysis of high rate UMC's is complicated compared to rate $1/2$ codes with similar parameters. For full UMC's, i.e., codes with memory k, there are many different encoders, and long codewords have to be generated in the calculation of d_∞. For these reasons we have found that partial UMC's are often preferable.

As a simple case of practical interest we have considered codes of rate $2/3$ generated by matrices of the form

$$\{\boldsymbol{G}_0, \boldsymbol{G}_1\} = \begin{Bmatrix} \boldsymbol{G}_0' & \boldsymbol{0} \\ \boldsymbol{G}_0'' & \boldsymbol{G}_1'' \end{Bmatrix} \tag{5}$$

where the nonzero submatrices consist of 3 $m \times m$ circulants. If the $3m$ rows of the matrices $\boldsymbol{G}_0'$, $\boldsymbol{G}_0''$ and $\boldsymbol{G}_1''$ are linearly independent, it follows from Theorems 1 and 2 that the encoder is noncatastrophic and minimal. The dual code is a QCUMC with rate $1/3$, and the generating polynomials of this code can be found as the determinants of the 2×2 submatrices of the generator matrix in D-notation. Table III contains the generator matrices and distances of some codes of this type and as in Table I we list the first row of the submatrices in octal form.

As simple upper bounds on the free distances we may take the minimum distances of the best $(3m, m)$ and $(6m, 3m)$ block codes.

In some cases it is possible to improve the distances significantly by adding overall parity checks to $\boldsymbol{G}_0$ and/or $\boldsymbol{G}_1$. As shown in Table III, an even weight (16,10) code with free distance 8 and linear growth rate 2 may be obtained from the (15,10) code.

A full (16,10) UMC with free distance 12 has been found by using a (31,10,12) block code as a starting point. There are two nonequivalent cyclic codes with these parameters, but it is not practical to consider all possible partitionings and permutations of the generator matrices. By suitable permutations of the positions of the cyclic codes, one can obtain a generator matrix in QC form [15]. The matrix consists of 12 5×5 circulants and a column of ones, which may be interpreted as an overall parity check. The matrix may be modified to a (32,10,12) code by replacing this parity check by two columns, which are parity checks on the first and the last three circulants, respectively. This modification does not change the minimum distance.

For each of the two nonequivalent cyclic codes there are 20 ways of partitioning the circulants between $\boldsymbol{G}_0$ and $\boldsymbol{G}_1$. If we require that $d_0^c = 4$, we end up with only 6 and 4 partitionings. For each partition we can fix $\boldsymbol{G}_0$, and there are 6×5^3 QCUMC's corresponding to different orderings of the blocks in $\boldsymbol{G}_1$ and shifts of the circulants, but many of these codes are equivalent, and using the rules from Section II, the overall number of codes is reduced to 1500. Among these we found 14 codes with $d_\infty = 12$, but only one with the distance profile $\boldsymbol{d} = \{4,6,8,8,10,12\}$. This code, which can also be written in the form (5) except that the all-zero submatrix in $\boldsymbol{G}_1$ is now substituted by $\boldsymbol{G}_1'$, is also listed in Table III.

The extended row distances for the codes in Table III are shown in Table IV. A comparison of the two (16,10) codes shows that, for short codes, large free distance

TABLE III
QCUMC WITH $R > \frac{1}{2}$

n,k,v	G_0'	G_0''	G_1'	G_1''	d_0^c	d_1^c	d_2^c	d_3^c	d_4^c	d_5^c	d_∞	d_∞(MMC)	d_∞(UB)
6,4,2	56	13	00	12	2	3	4				4	3	4
12,8,4	4316	0206	000	0236	3	4	5	6			6	5	6
15,10,5	41626	01033	00000	50465	3	4	5	6			6	6	7
16,10,5	416264	010334	000000	504654	4	6	8				8		8
16,10,10	400274	010064	621154	545620	4	6	8	8	10	12	12		12

TABLE IV
EXTENDED ROW DISTANCES FOR QCUMC IN TABLE III

n,k,r	d_0^r	d_1^r	d_2^r	d_3^r	d_4^r	d_5^r	d_6^r	d_7^r	d_8^r	d_9^r	d_{10}^r	d_{11}^r	d_{12}^r	d_{13}^r	d_{14}^r	d_{15}^r	d_{16}^r	d_{17}^r	d_{18}^r	d_{19}^r	d_{20}^r	w_0	GB[1]
6,4,2	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	1.00	0.37
12,8,4	6	6	8	8	10	12	12	14	16	16	18	20	20	22	24	24	26	28	28	30	32	1.33	0.74
15,10,5	6	8	8	10	11	13	14	16	17	19	20	22	23	25	26	28	29	31	32	34	35	1.50	0.92
16,10,5	8	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	2.00	1.16
16,10,10	12	12	12	14	14	16	18	20	20	22	22	24	24	26	28	28	30	30	32	32	34	1.14	1.16

[1] GB is the Gilbert bound on block codes corresponding to terminated convolutional codes, i.e., $nH^{-1}(1-R)$.

often implies a low linear growth rate. This phenomenon may be explained by interpreting the free distance as the minimum distance of a critical block code. For a given rate, this distance can only be increased by increasing the length of the code, and for short codes, increasing length implies a decreasing relative distance.

REFERENCES

[1] L. N. Lee, "Short unit-memory byte-oriented binary convolutional codes having maximal free distance," *IEEE Trans. Inform, Theory*, vol. IT-22, p. 349–352, May 1976.

[2] C. Thommesen and J. Justesen, "Bounds on distances and error exponents of unit-memory codes," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 637–649, Sept. 1983.

[3] G. S. Lauer, "Some optimal partial-unit-memory codes," *IEEE Trans. Inform. Theory*. vol. IT-25, pp. 240–243, Mar. 1979.

[4] R. Johannesson, "Robustly optimal rate one-half binary convolutional codes," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 464–468, July 1975.

[5] R. Johannesson and E. Paaske, "Further results on binary convolutional codes with optimal distance profile," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 264–268, Mar. 1978.

[6] G. D. Forney, Jr., "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 720–738, Nov. 1970.

[7] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correction Codes*. New York: North Holland, 1977.

[8] G. Solomon and H. C. A. van Tilborg, "A connection between block and convolutional codes," *SIAM J. Appl. Math.*, vol. 37, pp. 358–369, Oct. 1979.

[9] P. Piret, "Structure and constructions of cyclic convolutional codes," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 147–155, Mar. 1976.

[10] F. Hemmati and D. J. Costello, Jr., "Asymptotically catastrophic convultional codes," *IEEE Trans. Inform. Theory*, vol. IT-26, pp. 298–304, May 1980.

[11] G. K. Huth and C. L. Weber, "Minimum weight convolutional codewords of finite length," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 243–246, Mar. 1976.

[12] P. R. Chevillat and D. J. Costello, Jr., "Distance and computation in sequential decoding," *IEEE Trans. Commun. Technol*, vol. COM-24, pp. 440–447, Apr. 1976.

[13] L. R. Bahl, C. D. Cullum, W.D. Frazer, and F. Jelinek, "An efficient algorithm for computing free distance," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 437–439, May 1972.

[14] K. J. Larsen,, "Comments on An efficient algorithm for computing free distance," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 577–579, July 1973.

[15] C. W. Hoffner and S. M. Reddy, "Circulant bases for cyclic codes," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 511–512, July 1970.

[16] S. W. Colomb, *Shift Register Sequences*. San Francisco, CA: Holden-Day, Inc., 1967.

[17] T. Verhoeff, "An updated table of minimum-distance bounds for binary linear codes," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 665–680, Sept. 1987.