International Journal of Pure and Applied Mathematics
Volume 50 No. 3 2009, 431-463

CONVOLUTIONAL CODES FROM UNITS IN MATRIX AND GROUP RINGS

Ted Hurley
School of Mathematics, Statistics and Applied Mathematics
National University of Ireland – Galway
Galway, IRELAND
e-mail: ted.hurley@nuigalway.ie

Abstract: A general method for the construction and analysis of convolutional codes from units in Laurent series over matrix rings is presented. Using group rings as matrix rings, this gives a general method for the construction and analysis of convolutional codes from group rings. A theory of group ring convolutional codes is developed. Series of convolutional codes are constructed algebraically and algebraic methods are used to compute free distances and to construct convolutional codes to prescribed minimum distances.

AMS Subject Classification: 16S34, 68P30
Key Words: convolutional code, group ring

1. Introduction

Methods are presented for constructing and analysing convolutional codes using units in Laurent series of finite support over matrix rings. By considering group rings as matrix rings, convolutional codes are constructed and analysed from units in Laurent series over group rings. Thus convolutional codes are constructed in group rings RG where the ring R is itself a group ring. Matrix versions may be directly obtained from the group ring description.

Group rings have a rich structure and this rich structure is exploited here just as, for example, the rich structure of the cyclic group ring is exploited in the study and construction of cyclic linear codes.

The lack of algebraic methods for constructing convolutional codes is men-

Received: January 7, 2009

tioned a number of times in [9] and many of the existing convolutional codes have been found by computer search. Using algebraic methods here, the range of convolutional codes available is expanded and series of convolutional codes are derived. Free distances and codes to a prescribed minimum free distances are also derived.

The methods are fairly general and use properties of group rings and their embeddings into matrix rings. Zero-divisors and units in group rings enable the construction of units in certain polynomial rings and/or group rings over these group rings from which convolutional codes may be constructed. Properties of the convolutional codes may be studied and derived from properties of group rings. In many instances the free distances can be calculated algebraically and convolutional codes to a specified free distance, as for example in Theorem 5.2 below, can be constructed. General theorems are proved and theory is developed.

The general method can be applied to specific applications and these in themselves constitute constructions for convolutional codes with their own theory. Such applications included in the paper are listed below. Here C_∞ denotes the infinite cyclic group, C_n is the cyclic group of order n and F is a field.

1. Infinite series of $(2,1)$ convolutional codes are constructed in Section 5 using $(FC_2)C_\infty$. Free distances are calculated algebraically.

2. Given a linear cyclic code $\mathcal{C}$ with $d = \min(d_1, d_2)$ where d_1 is the minimum distance of $\mathcal{C}$ and d_2 is the distance of the dual of $\mathcal{C}$, the generator polynomial f of $\mathcal{C}$ is mimicked in $(FC_2)C_\infty$ to construct convolutional $(2,1)$ codes of minimum free distance $d+2$.

3. Rate $\frac{3}{4}$ convolutional codes are constructed and free distances calculated. The method may be extended to construct higher rate convolutional codes but details on these are not included.

4. Convolutional codes over a field F of characteristic p, for any prime p, using nilpotent elements in the group ring FG are constructed; here G must be a group whose order is divisible by p. These codes can have rates of the order of $\frac{p-1}{p}$ since the matrices of the coefficients can have rank $|G|(\frac{p-1}{p})$.

5. Convolutional codes within $(FG)C_\infty$ are constructed and free distances estimated with matrices of coefficients similar to the matrices in the *Hamming linear codes*. These may be constructed to a desired minimum free distance.

6. Convolutional codes are constructed using idempotents in group rings. These are particularly used in cases where the characteristic of the field does not divide the order of the group; *characters* of groups and *character tables*

come into play in constructing these convolutional codes.

The convolutional codes in 1. and 2. above are reminiscent of cyclic codes with properties of cyclic codes but also being convolutional have the advantage of a 'memory'. Cyclic codes are obtained in the group ring FC_n for a ring (often a field) F and (finite) cyclic group C_n. Replacing the ring F by the ring FC_2 and the finite group C_n by the infinite cyclic group C_∞ in FC_n gives the group ring $(FC_2)C_\infty$. This group ring is then used to define and analyse the resulting convolutional codes.

To obtain convolutional codes over fields of characteristic p the group ring $(FC_p)C_\infty$ is used where p is the characteristic of F, or in general use $(FG)C_\infty$ where the characteristic of F divides the order of G. The 'Hamming type' codes in 5. above use the group ring $(\mathbb{Z}_2(C_4 \times C_2))C_\infty$ where $C_4 \times C_2$ denotes the direct product of C_4 and C_2.

1.1. Algebraic Description of Convolutional Codes

Background on general algebra and group rings may be obtained in [10].

$R[z]$ denotes the polynomial ring with coefficients from a ring R and $R_{r\times n}$ denotes the ring of $r \times n$ matrices with coefficients from R. R^n is used to denote $R_{1\times n}$ and thus $R^n = \{(r_1, r_2, \ldots, r_n) : r_i \in R\}$.

It is easy to verify that $R_{r\times n}[z] \cong R[z]_{r\times n}$.

$R[z, z^{-1}]$ denotes the ring of Laurent series of finite support in z with coefficients from R. *Finite support* means that only a finite number of the coefficients are non-zero. It is clear that $R[z, z^{-1}] \cong RC_\infty$, where C_∞ denotes the infinite cyclic group. Note also that $R[z] \cong T$ where T is the algebra of *non-negative elements* in RC_∞, i.e. the algebra of elements $w = \sum_{i=0}^{\infty} \alpha_i g^i$, in RC_∞, where C_∞ is generated by g.

If $\mathbb{F}$ is an integral domain then $\mathbb{F}[z]$ has no zero-divisors and only trivial units – the units of $\mathbb{F}[z]$ are the units of $\mathbb{F}$.

See [9] and/or [1] for basic information on convolutional codes, and algebraic descriptions presented therein. The (equivalent) useful algebraic description as described in [2] is given below.

A convolutional code $\mathcal{C}$ of length n and dimension k is a direct summand of $\mathbb{F}[z]^n$ of rank k. Here $\mathbb{F}[z]$ is the polynomial ring over $\mathbb{F}$ and $\mathbb{F}[z]^n = \{(v_1, v_2, \ldots, v_n) : v_i \in \mathbb{F}[z]\}$.

Suppose V is a submodule of $\mathbb{F}[z]^n$ and that $\{v_1, \ldots, v_r\} \subset \mathbb{F}[z]^n$ forms a generating set for V. Then $V = \text{Image}\, M = \{uM : u \in \mathbb{F}[z]^r\}$ where $M = \begin{bmatrix} v_1 \\ \vdots \\ v_r \end{bmatrix} \in \mathbb{F}[z]_{r\times n}$. This M is called a *generating matrix* of V.

A generating matrix $G \in \mathbb{F}[z]_{r\times n}$ having rank r is called a *generator* or *encoder matrix* of $\mathcal{C}$. A matrix $H \in \mathbb{F}[z]_{n\times(n-k)}$ satisfying $\mathcal{C} = \ker H = \{v \in \mathbb{F}[z]^n : vH = 0\}$ is said to be a *control matrix* of the code $\mathcal{C}$.

2. Convolutional Codes from Units

Let R be a subring of the ring of matrices $F_{n\times n}$. In particular the group ring FG is a subring of $F_{n\times n}$, where $n = |G|$, by an explicit embedding given in [4].

There is no restriction on F in general but it is assumed to be a field here; however many of the results will hold more generally.

Units and *zero-divisors* in any ring are defined in the usual manner.

Construct R-convolutional codes as follows.

2.1. General Construction

Suppose $f(z, z^{-1}), g(z, z^{-1}) \in R[z, z^{-1}]$ satisfy $f(z, z^{-1})g(z, z^{-1}) = 1$.

Consider (compatible) block decompositions of f, g as follows:

$$f(z, z^{-1}) = \begin{pmatrix} f_1(z, z^{-1}) \\ f_2(z, z^{-1}) \end{pmatrix},$$

$$g(z, z^{-1}) = \begin{pmatrix} g_1(z, z^{-1}), & g_2(z, z^{-1}) \end{pmatrix},$$

where $f_1(z, z^{-1})$ is an $r \times n$ matrix, $f_2(z, z^{-1})$ is an $(n-r)\times n$ matrix, $g_1(z, z^{-1})$ is an $n \times r$ matrix and $g_2(z, z^{-1})$ is an $n \times (n-r)$ matrix.

Then

$$\begin{pmatrix} f_1(z, z^{-1}) \\ f_2(z, z^{-1}) \end{pmatrix} \times \begin{pmatrix} g_1(z, z^{-1}), & g_2(z, z^{-1}) \end{pmatrix} = 1\,.$$

Thus

$$\begin{pmatrix} f_1g_1 & f_1g_2 \\ f_2g_1 & f_2g_2 \end{pmatrix} = 1\,.$$

From this it follows that

$$f_1(z,z^{-1})g_1(z,z^{-1}) = I_{r\times r}, \quad f_1(z,z^{-1})g_2(z,z^{-1}) = 0_{r\times(n-r)},$$
$$f_2(z,z^{-1})g_1(z,z^{-1}) = 0_{(n-r)\times r}, \quad f_2(z,z^{-1})g_2(z,z^{-1}) = I_{(n-r)\times(n-r)}.$$

Thus $f_1(z,z^{-1})$ is taken as the generator or encoder matrix of an (n,r) convolutional code and $g_2(z,z^{-1})$ is then the check or control matrix for the code. It is seen in particular that $f_1(z,z^{-1})$, (and $f_2(z,z^{-1})$) have right finite support inverses and thus by Theorem 6.6 of [9] the generator matrix f_1 is noncatastrophic.

More generally, given $f(z,z^{-1})g(z,z^{-1}) = 1$ *any* r rows of $f(z,z^{-1})$ can be used to construct a generator matrix of a convolutional code and the control matrix may be obtained directly from $g(z,z^{-1})$. If rows $\{j_1, j_2, \ldots, j_r\}$ are chosen from $f(z,z^{-1})$ then an encoding $F^r[z] \to F^n[z]$ is obtained with generator matrix consisting of these r rows of $f(z)$ and check/control matrix is obtained by deleting the $\{j_1, j_2, \ldots, j_r\}$ columns of $g(z)$. This can also be seen by permuting the rows of f appropriately and using the particular case as described above.

2.2. Polynomial Case

This construction holds in particular when one or both of f, g are polynomials.

It is known that convolutional codes have polynomial generator matrices but the more general case as described in Section 2.1 is often more suitable for initially constructing and analysing the codes. It is easy to obtain polynomial generator and control matrices from Laurent matrices of finite support.

For clarity a detailed analysis of this polynomial case is now given.

Suppose $f(z)g(z) = 1$ in $R[z]$. The encoder matrix is obtained from $f(z)$ and the decoder or control matrix is obtained from $g(z)$ using a general method for constructing codes from units.

Now $f(z) = (f_{i,j}(z))$ is an $n \times n$ matrix with entries $f_{i,j}(z) \in F[z]$. Similarly $g(z) = (g_{i,j}(z))$ is an $n \times n$ matrix over $F[z]$. Suppose $r[z] \in F[z]^r$ and consider $r[z]$ as an element of $\mathbb{F}[z]^n$ (by adding zeros to the end of it). Then define a mapping $\gamma : F[z]^r \to F[z]^n$ by $\gamma : r(z) \mapsto r(z)f(z)$. The code $\mathcal{C}$ is the image of γ. Since $r[z]$ has zeros in its last $(n-r)$ entries as a member of $F[z]^n$, this means that *the generator matrix is the first r rows* of $f(z)$ which is an $r \times n$ matrix over $F[z]$. Since $f(z)$ is invertible, this generator matrix has rank r and is thus the encoder matrix which we denote by $G(z)$. For this polynomial case,

$G(z)$ is a basic generator matrix – see A.1 Theorem in [9].

$w(z) \in \mathbb{F}[z]^n$ is a codeword if and only if $w(z)g(z)$ is in $\mathbb{F}[z]^r$, that is, if and only if the final $(n-r)$ entries of $w(z)g(z)$ are all 0. Suppose $w(z) = (\alpha_1(z), \alpha_2(z), \ldots, \alpha_n(z))$. Then this condition is that

$$(\alpha_1(z), \alpha_2(z), \ldots, \alpha_n(z)) \cdot \begin{pmatrix} g_{1,r+1}(z) & g_{1,r+2}(z) & \ldots & g_{1,n}(z) \\ g_{2,r+1}(z) & g_{2,r+2}(z) & \ldots & g_{2,n}(z) \\ \vdots & \vdots & \vdots & \vdots \\ g_{n,r+1}(z) & g_{n,r+2}(z) & \ldots & g_{n,n}(z) \end{pmatrix} = 0\,.$$

The check or *control matrix* $H(z)$ of the code is thus:

$$\begin{pmatrix} g_{1,r+1}(z) & g_{1,r+2}(z) & \ldots & g_{1,n}(z) \\ g_{2,r+1}(z) & g_{2,r+2}(z) & \ldots & g_{2,n}(z) \\ \vdots & \vdots & \vdots & \vdots \\ g_{n,r+1}(z) & g_{n,r+2}(z) & \ldots & g_{n,n}(z) \end{pmatrix}.$$

This has size $n \times (n-r)$ and is the matrix consisting of the last $(n-r)$ columns of $g(z)$ or in other words the matrix obtained by deleting the first r columns of $g(z)$. Since $f(z), g(z)$ are units, it is automatic that $\operatorname{rank} G(z) = r$ and $\operatorname{rank} H(z) = (n-r)$.

2.3. Particular Useful Case

Suppose $f(z)g(z) = z^t$ in $R[z]$. A general method for producing such f, g within group rings is given later. Then $f(z)(g(z)/z^t) = 1$. Now $(g(z)/z^t)$ involves negative powers of z but has finite support. The encoder matrix is obtained from $f(z)$ and the decoder or control matrix is obtained from $(g(z)/z^t)$ using the method as formulated in Section 2.1.

The control matrix contains negative powers of z but it is easy to obtain a polynomial control matrix from this.

In these cases it is possible to multiply by the unit z^{-t} and stay within $R[z, z^{-1}]$. Other elements in $R[z, z^{-1}]$ have inverses but with infinite support and thus outside $R[z, z^{-1}]$.

3. Group Ring Convolutional Codes

3.1. Group Ring Matrices

In the constructions of Section 2.1, R is any subring of $F_{n\times n}$. Suppose now $R = FG$ is the group ring of the group G over F.

The group ring $R = FG$ with $|G| = n$ is a subring of $F_{n\times n}$ using an explicit injection from FG to a subring of $F_{n\times n}$ as found for example in [4].

Thus the methods of Section 2.1 may be used to define convolutional codes by considering $R[z, z^{-1}] \cong RC_\infty$, which is the group ring over C_∞ with coefficients from the group ring $R = FG$.

To obtain units in $R[z, z^{-1}]$ (which includes $R[z]$) we are lead to consider zero-divisors and units in $R = FG$.

$R = FG$ is a rich source of zero-divisors and units, and consequently $R[z, z^{-1}]$ is a rich source of units. There are methods available for constructing units and zero-divisors in FG. What is required are units in $R[z]$, where $R = FG$, a group ring, and these can be obtained by the use of zero-divisors and units in R as coefficients of the powers of z.

FG also has a rich structure in which to analyse any resulting code. When F is a field, every non-zero element of FG is either a unit or a zero-divisor.

In what follows bear in mind that R has zero-divisors and units, as when R is a group ring, and that these zero-divisors and units are used to construct units in $R[z, z^{-1}]$.

3.2. Codes from Group Rings

Suppose then $(\sum_{i=-m}^{n} \alpha_i z^i)(\sum_{j=-m}^{n} \beta_j z^j) = 1$ in the group ring $RC_\infty = R[z, z^{-1}]$ with $\alpha_i.\beta_j \in R$ and C_∞ generated by z. By multiplying through by a power of z this is then $(\sum_{i=0}^{n} \alpha_i z^i)(\sum_{j=-m}^{n} \beta_j z^j) = 1$.

The case with $m = 0$ gives polynomials over z. Here $(\sum_{i=0}^{n} \alpha_i z^i)(\sum_{j=0}^{t} \beta_j z^j) = 1$ where $\alpha_n \neq 0, \beta_t \neq 0$ and looking at the coefficient of z^0 it is clear that we must also have $\alpha_0 \neq 0, \beta_0 \neq 0$. This can be considered an an equation in RC_∞ with

non-negative powers. Solutions may be used to construct convolutional codes.

By looking at the highest and lowest coefficients we then have that $\alpha_0\beta_0 = 1$ and $\alpha_n\beta_t = 0$. Thus in particular α_0 is a unit with inverse β_0 and α_n, β_t are zero divisors.

Solutions of the general equation $(\sum_{i=0}^{n} \alpha_i z^i)(\sum_{j=-m}^{n} \beta_j z^j) = 1$ can also be used to form convolutional codes and polynomial generator matrices may be derived from these.

3.3. Prototype Examples

First Prototype Example. Let $R = \mathbb{Z}_2 C_4$, where C_4 is the cyclic group of order 4 generated by a. Then $\alpha_0 = a + a^2 + a^3$ satisfies $\alpha_0^2 = 1$ and $\alpha_2 = a + a^3$ satisfies $\alpha_2^2 = 0$.

Thus $w = \alpha_0 + \alpha_1 z + \alpha_2 z^2$ in RC_∞ satisfies $w^2 = \alpha_0\alpha_0 + z(\alpha_0\alpha_1 + \alpha_1\alpha_0) + z^2(\alpha_0\alpha_2 + \alpha_1^2 + \alpha_2\alpha_0) + z^3(\alpha_1\alpha_2 + \alpha_2\alpha_1) + z^4(\alpha_2\alpha_2) = 1 + z^2\alpha_1^2$, since the α_i commute. Now require that $\alpha_1^2 = 0$ and then $w^2 = 1$.

In particular letting $\alpha_1 = \alpha_2$ implies that $w^2 = 1$. Consider for example $\alpha_1 = 1 + a^2$ and then also $\alpha_1^2 = 0$.

Now use the injection, [4], from $R = \mathbb{Z}_2 C_4$ into the ring of 4×4 matrices over $\mathbb{Z}_2$ (the natural listing $\{1, a, a^2, a^3\}$ of the elements of C_4 is used for this injection). Then under this injection α_0 corresponds to the matrix $\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$, α_2 corresponds to the matrix $\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$ and α_1 corresponds to the matrix $\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$.

Take the first two rows of w to generate a convolutional code and then the last two columns of w is the control matrix of this code as $w^2 = 1$.

This gives the following generator matrix:

$$G = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} z + \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} z^2 .$$

The control matrix is:

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} z + \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} z^2 .$$

The code has length 4 and dimension 2. It may be shown that the free distance of this code is 6.

Second Prototype Example. Let $u = 1 + h(a + a^2 + a^3)$ in $\mathbb{Z}_2(C_4 \times C_2)$. Then $u^2 = 0$ and $\operatorname{rank} u = 4$. Define $w = u + z + uz^2$. Then $w^2 = z^2$ and w is used to define an $(8, 4)$ convolutional code. The generator matrix is $G = (I, B) + (I, 0)z + (I, B)z^2$ where $B = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$. Now the code generated by (I, B) has distance 4 as a linear code. Any combination of $(I, B), (I, 0)$ has distance 1 at least as B is non-singular. Thus consider $(\sum_{i=0}^{t} \beta_i z^t)G$. The highest and lowest power of z has distance 4 and there is a power of z in between which has distance 1 so altogether we get a free distance of 9. The degree of the code is 8.

See Section 11 for further examples of these types.

4. Convolutional Codes from Nilpotent Elements

The following two Lemmas are useful in constructing new classes of convolutional codes. Their proofs are straight-forward. The second is a generalisation of the first but is listed separately for subsequent applications.

Lemma 4.1. *Let $R = FG$ be the group ring of a group G over a field F with characteristic p. Suppose $\alpha_i \in R$ commute. Let $w = \sum_{i=0}^{n} \alpha_i z^i \in RC_\infty$. Then $w^p = 1$ if and only if $\alpha_0^p = 1, \alpha_i^p = 0, i > 0$.*

Lemma 4.2. *Let $R = FG$ be the group ring of a group G over a field*

F with characteristic p. Suppose $\alpha_i \in R$ *commute. Let* $w = \sum_{i=0}^{n} \alpha_i z^i \in RC_\infty$. *Then* $w^p = z^{pt}$ *if and only if* $\alpha_i^p = 0, i \neq t$ *and* $\alpha_t^p = 1$.

Now construct convolutional codes by the following general method: Find elements α_i with $\alpha_i^p = 0$ and units u with $u^p = 1$ in the group ring R. Form units in $R[z]$ or $R[z, z^{-1}]$ using Lemma 4.1 or Lemma 4.2. From these units, convolutional codes are defined using the methods described in Section 2.1.

4.1. Class of Examples

Consider now $\alpha_0 = a + a^2 + a^3$ and for $i > 0$ define $\alpha_i = a + a^3$ or $\alpha_i = 0$ in the group ring $R = \mathbb{Z}_2 C_4$. Then $\alpha_0^2 = 1$ and $\alpha_i^2 = 0, i > 0$. We could also take $\alpha_i = 1 + a^2$.

Define $w(z) = \sum_{i=0}^{n} \alpha_i z^i$ in RC_∞. By Lemma 4.1, $w^2 = 1$.

The matrix corresponding to α_0 is $\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$ and the matrix corresponding to $\alpha_i, i \neq 0$ is $\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$ or else is the zero matrix.

Now specify that the first two rows of w give the generator matrix and from this it follows that the last two columns of w is a control matrix.

This gives the following generator matrix:

$$G = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} + \delta_1 \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} z + \delta_2 \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} z^2 + \ldots + \delta_n \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} z^n,$$

where $\delta_i = 1$ when $\alpha_i \neq 0$ and $\delta_i = 0$ when $\alpha_i = 0$.

The control matrix is:

$$H = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} + \delta_1 \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} z + \delta_2 \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} z^2 + \ldots + \delta_n \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} z^n.$$

The code has length 4 and dimension 2. Assume at least 3 of the α_i are non-zero. The free distance is at least 6 for any $n \geq 2$ and in many cases it will be larger. Polynomials used for generating cyclic linear codes suitably converted to polynomials in $R[z]$ prove particularly useful and amenable – see for example Section 5 below.

For example the $(4, 2)$ convolutional code with generator and check matrices as follows has free distance 8.

$$G = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} z + \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} z^3 + \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} z^4.$$

$$H = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} z + \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} z^3 + \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} z^4.$$

4.2. Direct Products: Turbo-Effect

Examples of convolutional codes formed using α_i with $\alpha_i^2 = 0$ in FG have been produced. Consider now $F(G \times H)$ and let $w = \beta \times \alpha_i$ for *any* $\beta \in FH$. Then $w^2 = \beta^2 \alpha_i^2 = 0$. This expands enormously the range of available elements whose square is zero. Note also that over a field of characteristic 2 if $\alpha^2 = 0 = \gamma^2$ then $(\alpha + \gamma)^2 = 0$.

For example in $\mathbb{Z}_2 C_2$ the element $1 + a$ was used where C_2 is generated by a. Then in $\mathbb{Z}_2(G \times C_2)$ consider $\alpha = \beta(1 + a)$ for any $\beta \in \mathbb{Z}_2 G$. Then $\alpha^2 = 0$.

A simple example of this is $\mathbb{Z}_2(C_2 \times C_2)$ where $\alpha = (1+a)b+(1+b)a = a+b$. The matrix of $a + b$ is $\begin{pmatrix} A & B \\ B & A \end{pmatrix}$ where $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ and $B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. In forming $(4, 2)$ convolutional codes we would only use the top half of the matrices, i.e. $P = \left(\begin{array}{cc|cc} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{array}\right)$. Note that in this encoding the vector

(γ, δ) is mapped to $(\gamma, \delta)P = \left(\begin{array}{cc|cc} \delta & \gamma & \gamma & \delta \end{array} \right)$. This is like an interweaving of two codes.

To get a permutation effect, use the direct product with S_n, the permutation or symmetric group on n letters.

5. (2,1) Codes

Examples of $(2, 1)$ optimal codes up to degree 10 may be obtained in [9]. Some of these can be reproduced algebraically and their properties derived using the methods developed here.

Further new $(2, 1)$ convolutional codes and series of convolutional $(2, 1)$ are constructed in this section as an application of the general methods described above. The free distances can often be determined algebraically and codes to a prescribed free distance can be constructed by using Theorem 5.2 below.

Let F be a field of characteristic 2 and $R = FC_2$, where C_2 is generated by a. Consider elements $\alpha_i \in R$, $i > 0$, where either $\alpha_i = 1 + a$ or $\alpha_i = 0$. Then $\alpha_i^2 = 0$.

Let $\alpha_0 = 1$ in R and define $w = \alpha_1 + \alpha_0 z + \alpha_2 z^2 + \ldots + \alpha_n z^n$. Then $w^2 = z^2$ and hence $w \times (w/z^2) = 1$. Thus w can be used to define a $(2, 1)$ convolutional code.

More generally let t be an integer, $0 \leq t \leq n$, and define $w = \sum_{i=0}^{n} \beta_i z^i$ where $\beta_i = \alpha_i, i \neq t$, $\beta_t = 1$. Then $w^2 = z^{2t}$ gives that $w \times (w/z^{2t}) = 1$. Thus w can be used to define a convolutional $(2, 1)$ code. The case $\alpha_0 = \beta_1$ is a special case.

Now determine the code by choosing the first row of the matrix of w to be the generator/encoder matrix and then the last column of w/z^{2t} is the control matrix.

The matrix of α_i is $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ when $\alpha_i = 1 + a$ and is the zero 2×2 matrix when $\alpha_i = 0$. .

Define $\delta_i = 1$ when $\alpha_i \neq 0$ and $i \neq t$; $\delta_i = 0$ when $\alpha_i = 0$ and $i \neq t$; and define $\delta_t(1, 1)$ to be $(1, 0)$.

Then the encoder matrix of the code is $G = (1, 1) + \delta_1(1, 1)z + \delta_2(1, 1)z^2 + \ldots + \delta_n(1, 1)z^n$ and with $H = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \delta_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix} z + \delta_2 \begin{pmatrix} 1 \\ 1 \end{pmatrix} z^2 + \ldots +$

$\delta_n \begin{pmatrix} 1 \\ 1 \end{pmatrix} z^n$, the control matrix is H/z^{2t}.

The generator matrix G obtained in this way is noncatastrophic as it has a right finite weight inverse – see Theorems 6.3 and 6.6 in [9].

For $n = 2$ we get as an example the code with the generator matrix $G = (1,1) + (1,0)z + (1,1)z^2$. This code has free distance 5 which is optimal. It is precisely the $(2,1,2,5)$ code as described in [9], p. 1085. A proof of the distance is given here as it demonstrates a general algebraic method for proving free distances, or getting a lower bound for distances, for codes constructed in the manner of this paper.

Basically wherever $(1,0)$ appears (once) in a sum making up a coefficient it will contribute a weight of 1, as the other non-zero coefficients which could possibly appear, $(1,1)$, will add up to $(1,1)$ or $(0,0)$.

Proposition 5.1. *The code determined by G has free distance* 5.

Proof. Consider $w = \sum_{i=0}^{s} \beta_i z^i G$, with $\beta_i \in \mathbb{Z}_2$ and $\beta_s \neq 0$. We may consider $\beta_0 \neq 0$. The coefficients of $z^0 = 1$ and z^{t+2} are $(1,1)$, and also $(1,0)$ occurs in the expression for at least one other coefficient. Thus the w is at least $2+2+1$ which is attained by G. □

The check matrix for this code is

$$\frac{\begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} z + \begin{pmatrix} 1 \\ 1 \end{pmatrix} z^2}{z^2} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} z^{-1} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} z^{-2}.$$

For $n \geq 3$ it may be verified directly by similar algebraic methods that the free distance is at least 6. Appropriate choices of the α_i will give bigger free distances. See Theorem 5.2 below.

For $n = 3$ and $\delta_2 = 1 = \delta_3$ a $(2,1,3,6)$ convolutional code is obtained which is also optimal. Thus a degree 3 optimal distance 6 is given by the encoder matrix $G = (1,1) + (1,0)z + (1,1)z^2 + (1,1)z^3$ and the control matrix is $H/z^2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} /z^2 + \begin{pmatrix} 1 \\ 0 \end{pmatrix} /z + \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} z$. It is clear that H is also a control matrix.

The next case is $(2,1,4)$ of degree 4. The optimal distance of one of these is 7. Consider $w = \alpha_1 + \alpha_0 z + \alpha_1 z^3 + \alpha_1 z^4$, where $\alpha_1 = 1 + a$ and $\alpha_0 = 1$ in $\mathbb{Z}_2 C_2$. Then $w^2 = z^2$ and thus w gives the encoder matrix and w/z^2 gives the check matrix. The encoder matrix is $G = (1,1) + (1,0)z + (1,1)z^3 + (1,1)z^4$.

Call this code $\mathcal{C}$.

Proposition 5.2. *The free distance of* $\mathcal{C}$ *is* 7.

Proof. Consider $(\sum_{i=0}^{t} \beta_i z^i)G$, with $\beta_i \in \mathbb{Z}_2$. In determining free distance we may consider $\beta_0 \neq 0$ and $\beta_t \neq 0$. The coefficients of $z^0 (= 1)$ and z^{t+4} are both $(1,1)$. If there are more than two non-zero β_i in the sum then $(1,0)$ occurs in at least three coefficients giving a weight of $2+2+3=7$ at least. It is now necessary to consider the case when there are just two β_i in the sum. It is easy to see then that at least three of the coefficients of z^i are $(1,1)$, and $(1,0)$ or $(0,1)$ is a coefficient of another. Thus the free distance is 7. □

Consider the next few degrees. Let $\alpha = 1 + a, \alpha_0 = 1$ in FC_2 where F has characteristic 2.

1. deg 5: $w = \alpha + \alpha_0 z + \alpha z^3 + \alpha z^4 + \alpha z^5$; gives a free distance of 8.

2. deg 6: $w = \alpha + \alpha z^2 + \alpha z^3 + \alpha_0 z^4 + \alpha z^5 + \alpha z^6$. This gives a free distance of 9.

3. This illustrates a general method of going from polynomials generating cyclic codes to polynomials generating $(2,1)$ convolutional codes. General results on these are given in Section 5.1. Consider the following degree 12 element.

$$w = \alpha + \alpha z^2 + \alpha z^4 + \alpha z^5 + \alpha z^6 + \alpha_0 z^9 + \alpha z^{10} + \alpha z^{11} + \alpha z^{12}\,.$$

This is obtained from the polynomial used for the (linear) Golay $(23,12)$ code – see e.g. [1] p. 119. The difference is that a z^{12} has been added and the coefficient of z^9 appears with coefficient α_0 and not 0 as in the Golay code. The exact free distance has still to be determined but is at least 14. Placing α_0 as the coefficient of other powers, besides the 9-th power, of z in w will also give good codes.

We thus study the best performance of convolutional codes derived from $w = \sum_{i=0}^{t} \alpha_i z^i$ where some $\alpha_t = 1 \in FC_2$, and all the other α_i are either 0 or else $1 + a$ in FC_2. Try to choose the α_i as one would for a linear cyclic code so as to maximise the (free) distance.

A method for forming convolutional codes and deriving bounds for the derived codes by mimicking the generator polynomials of cyclic codes is given in the following section.

5.1. From Cyclic Codes to Convolutional Codes

FC_n denotes the group ring of the cyclic group C_n over the field F and suppose C_n is generated by g.

A cyclic code of length n over F is given by a zero-divisor in this group ring. Suppose then the code $\mathcal{C}$ is generated by $f(g) \in FC_n$ and that $h(g)f(g) = 0$, where $h(g)$ is of minimal degree such that $h(g)f(g) = 0$. If $r(g)f(g) = 0$ then $r(g) = s(g)h(g)$. Also $h(g)$ is the generator polynomial for the dual code $\hat{\mathcal{C}}$ of $\mathcal{C}$.

Suppose then that C is an (n, k, d_1) code and that $\hat{\mathcal{C}}$ is an $(n, n-k, d_2)$ code. Let $d = \min(d_1, d_2)$.

Suppose $f(g) = \sum_{i=0}^{r} \epsilon_i g^i$, with $\epsilon_i \in F$ $(\epsilon_r \neq 0)$, is a generating polynomial for $\mathcal{C}$. In $f(g)$, assume $\epsilon_0 \neq 0$.

The degree of f is of course less than n. The number of non-zero coefficients in $f(g)$ is the support of $f(g)$.

Consider the polynomial $f(z)$ in $F[z]$. This has degree $< n$ and support $\geq d_1$ as does $f(g)$. Consider also $p(g) = (\sum_{i=0}^{t} \alpha_i g^i) f(g) = \sum_{i=0}^{n-1} \beta_i g^i$ in FC_n and $w(z) = (\sum_{i=0}^{t} \alpha_i z^i) f(z) = \sum_{i=0}^{s} \gamma_i z^i$ in $F[z]$. Note that it is not assumed that $t < n$.

The following lemma is straight-forward but fundamental.

Lemma 5.1. *For $0 \leq j \leq n-1$ the sum of the coefficients of $z^j, z^{j+n}, z^{j+2n}, \ldots$ in $w(z)$ is β_j.*

Corollary 5.1. *If $\sum_{i=0}^{n-1} \beta_i g^i$ has support $\geq d$, then $\sum_{i=0}^{s} \gamma_i z^i$ has support $\geq d$.*

Proof. Suppose in $\sum_{i=0}^{n-1} \beta_i g^i$ that $\beta_j \neq 0$. Then the sum of the coefficients of $z^j, z^{j+n}, \ldots$ in $\sum_{i=0}^{s} \gamma_i z^i$ is non-zero. Thus the coefficient of at least one of $z^j, z^{j+n}, \ldots$ in $\sum_{i=0}^{s} \gamma_i z^i$ is non-zero.

Hence the support of $\sum_{i=0}^{s} \gamma_i z^i \geq d$. □

Lemma 5.2. *Let* $w(z) = \sum_{i=0}^{t} \alpha_i z^i$ *and suppose for* $0 \leq j < n$ *the first non-zero coefficient in the sequence* $z^j, z^{j+n}, \ldots,$ *is* z^{j+kn}. *Then in* $w(z)(1+z^n)$ *the first term with non-zero coefficient in the sequence* $z^j, z^{j+n}, \ldots$ *is also* z^{j+kn}.

Theorem 5.1. *Let* $q(z) = \sum_{i=0}^{t} \alpha_i z^i$ *and* $w(z) = q(z)f(z) = (\sum_{i=0}^{t} \alpha_i z^i) f(z)$. *Then either* $w(z)$ *has support* $\geq d_1$ *or else* $q(g) = s(g)h(g)$ *in which case* $q(z)$ *has support* $\geq d_2$.

Proof. As the code generated by $f(g)$ has rank k, then $(\sum_{i=0}^{t} \alpha_i g^i) f(g) = (\sum_{i=0}^{k-1} \delta_i g^i) f(g)$.

Case 1. $\sum_{i=0}^{k-1} \delta_i g^i \neq 0$: In this case $\sum_{i=0}^{k-1} \delta_i g^i \neq 0$ has support > 0. Then $(\sum_{i=0}^{k-1} \alpha_i g^i) f(g)$ has support $\geq d_1$ as $\mathcal{C}$ has distance $\geq d_1$. Hence by Corollary 5.1 $w(z)$ has support $\geq d_1$.

Case 2. $\sum_{i=0}^{k-1} \delta_i g^i = 0$. Then $(\sum_{i=0}^{t} \alpha_i g^i) f(g) = 0$. Hence $\sum_{i=0}^{t} \alpha_i g^i = \alpha(g)h(g)$ for some polynomial $\alpha(g) \in FC_n$.

Case 2(1). $\sum_{i=0}^{t} \alpha_i g^i \neq 0$. Then $\sum_{i=0}^{t} \alpha_i g^i = \alpha(g)h(g)$ has support $\geq d_2$ as $\hat{\mathcal{C}}$ has distance d_2. Also in this case $\sum_{i=0}^{t} \alpha_i z^i$ has support $\geq d_2$ for if $\sum_{i=0}^{t} \alpha_i z^i$ has support $< d_2$ then $\sum_{i=0}^{t} \alpha_i g^i$ has support $< d_2$.

Case 2(2). Suppose now $\sum_{i=0}^{t} \alpha_i g^i = 0$. Then $\sum_{i=0}^{t} \alpha_i z^i = s(z)(1+z^n)^k$ where

$1 + z^n$ does not divide $s(z)$. Consider $w(z) = s(z)f(z)(1 + z^n)^k$. Now $s(z)$ is such that $s(g) = \sum_{i=0}^{q} \omega_i g^i \neq 0 \in FC_n$.

Thus as in Case 1 or Case 2(1), $s(z)f(z) = \sum_{i=0}^{t} \gamma_i z^i$ has support $\geq d$ and is such that for d of the j, $0 \leq j < n$, at least one of the coefficients in $z^j, z^{j+n}, \ldots$ is non-zero. Then by Lemma 5.2 $\alpha(z)f(z)(1 + z^n)^k$ has support $\geq d$. Hence support of $w(z)$ is this case is also $\geq d$. □

Assume now F has characteristic 2.

Let $\beta_i = \epsilon_i(1 + a)$ in the group ring FC_2 with C_2 generated by a and form $f_1(z) = \sum_{i=0}^{r} \beta_i z^i$. Then $\beta_j = 0$ in FC_2 if and only if $\epsilon_j = 0$ in F.

Choose a $\beta_t \neq 0$ in f_1 and replace this β_t in $f_1(z)$ by $1 \in FC_2$ to form $f_0(z) = \sum_{i=0}^{r} \alpha_i z^i$. Then $\alpha_j = 0$ if and only if $\epsilon_j = 0$.

So $f_0(z) = \sum_{i=0}^{r} \alpha_i z^i$ where $\alpha_i = \epsilon_i(1 + a)$ except when $i = t$ in which case $\alpha_t = \epsilon_t 1 = \epsilon_t$, where 1 denotes the identity of FC_2.

Then $f_0(z)^2 = z^{2t}$ and thus $f_0(z)(f_0(z)/z^{2t}) = 1$. We now use $f_0(z)$ to generate a convolutional code by taking just the first rows of the α_i. Thus the generating matrix is $\hat{f}(z) = \sum_{i=0}^{r} \hat{\alpha}_i z^i$ where $\hat{\alpha}_i$ is the first row of α_i. If $\alpha_i \neq 0$ and $i \neq t$ then $\hat{\alpha}_i = (1, 1)$, and $\hat{\alpha}_t = (1, 0)$.

Lemma 5.3. *Let* $w = \sum_{i=1}^{n} \beta_i(1, 1) + \beta(1, 0)$ *with* $\beta \neq 0$. *Then at least one component of* w *is not zero.*

Proof. Now $w = (\sum_{i=1}^{n} \alpha_i + \alpha, \sum_{i=1}^{n} \alpha_i)$. Since $\alpha \neq 0$ it is clear that one component of w is not zero. □

Assume now $\alpha_t \neq \alpha_0$ and $\alpha_t \neq \alpha_r$, i.e. 1 is not the first or last coefficient of $f_0(z)$ (A similar result holds if it does occur in one of these positions but the free distance derived is possibly less by 1).

Theorem 5.2. *Let* $\mathcal{C}$ *denote the convolutional code with generator matrix*

$\hat{f}(z)$. *Then the free distance of* $\mathcal{C}$ *is at least* $d+2$.

Proof. Consider $w(z) = (\sum_{i=0}^{t} \beta_i z^i)\hat{f}$ and we wish to show that its weight is $\geq d+2$. Here $\beta_i \in F$. Thus we need to show that the sum of the weights of the coefficients of z in $w(z)$ is $\geq d+2$. In calculating the weight of w we can assume $\beta_0 \neq 0$ and we also naturally assume $\beta_t \neq 0$.

Let $w_1(z) = \sum_{i=0}^{t} \beta_i z^i$. The support of w_1, supp(w_1), is the number of non-zero β_i. Suppose then supp$(w_1) \geq d$. Then in $w(z)$, α_t appears with the coefficient of z^i, for at least d different i with $0 < i < t+r$. Also the coefficient of $1 = z^0$ is $\beta_0(1,1)$ and the coefficient of z^{t+r} is $\beta_t(1,1)$ and each of these have distance 2. Then by Lemma 5.3, $w(z)$ has free distance at least d_2+2.

Now support of $w(z) = (\sum_{i=0}^{t} \beta_i z^i)\hat{f}$ is the same as the support of $w_1(z) = (\sum_{i=0}^{t} \beta_i z^i)f(z)$. By Theorem 5.1 $w_1(z)$ has support $\geq d_1$ or else $\sum_{i=0}^{t} \beta_i z^i$ has support $\geq d_2$. In the first case $w(z)$ has free distance $\geq d_1+2$ and in the second case, as already shown, w has free distance $\geq d_2+2$. Thus w has free distance $\geq d+2$ as required. □

The free distance may be bigger than $d+2$; an upper bound is $2d-1$. The free distance also depends on where the invertible element is placed in the expression for $f(z)$.

It is worth noting that if the support of the input element is $\geq t$ then the free distance is at least $t+2$; this may be seen from the proof of Theorem 5.2. Thus it is possible to avoid short weight codewords by ensuring that the input elements have sufficient support – this could be done by, for example, taking the complement of any element with small support.

For a self-dual code the distances of the code and its dual are the same, that is $d_1 = d_2 = d$ in the notation of Theorem 5.2, so that using the generator polynomial for a self-dual code may be optimal if a suitable one can be found.

These convolutional codes can be considered to be self-dual type convolutional codes in the sense that $f(z)$ determines the generator matrix and $f(z)/z^{2t}$ determines the control matrix.

Convolutional codes are constructed in [8] via cyclic block codes. The ones here are directly from cyclic polynomials and minimal distances are derived

algebraically. In order to get a convolutional code of a desired free distance, Theorem 5.1 shows that it can be obtained from a polynomial of a cyclic code.

6. (2m,1) Codes

The previous section Section 5 can be generalised to produce convolutional codes of smaller rate $(2m, 1)$ but with much bigger free distance. Essentially the free distance is multiplied by m over that obtained for similar $(2, 1)$ codes.

The group to consider is C_{2m} generated by a. Assume m is odd although similar results may be obtained when m is even. Let $\alpha = 1+a+a^2+\ldots+a^{2m-1}$ and $\alpha_0 = 1+a^2+\ldots+a^{2m-2}$. Then $\alpha^2 = 0$ and $\alpha_0^2 = 1$ as α_0 has odd support.

Define as before $f(z) = \sum_{i=1}^{r} \alpha_i z^i$ where now $\alpha_i = \beta_i \alpha$ in $\mathbb{Z}_2 C_{2m}$. Replace some $\alpha_i \neq 0$, say α_t, by α_0.

Then $f(z)^2 = z^{2t}$ and $f(z)(f(z)/z^{2t}) = 1$. Thus use $f(z)$ to define a convolutional code $\mathcal{C}$ by taking the first row of the α_i.

For example $G(z) = (1,1,1,1,1,1) + (1,0,1,0,1,0)z + (1,1,1,1,1,1)z^2$ defines a $(6, 1)$ convolutional code which has free distance 15. $G(z) = (1,1,1,1,1,1)$ $+(1,0,1,0,1,0)z + (1,1,1,1,1,1)z^3 + (1,1,1,1,1,1)z^4$ defines a convolutional code which has free distance 21.

A theorem similar to Theorem 5.2 is also true: Let $f(g)$ denote the generator polynomial of a cyclic code with distance d_1 and whose dual code has distance d_2. Let $d = \min(d_1, d_2)$ and let $\mathcal{C}$ denote the convolutional code obtained from $f(z)$ where the coefficients of $f(g)$ have been replaced by α_i in all but one coefficient which has been replaced by α_0 and the first row of each coefficient is used. Assume in the following theorem that α_0 is not the first or last coefficient. The (algebraic)) proof of the following theorem is similar to the proof of Theorem 5.2 and is omitted.

Theorem 6.1. *The free distance of $\mathcal{C}$ is at least $md + 2m$.*

7. Higher Rate Convolutional Codes

The methods of Section 5 can also be generalised to produce higher rate convolutional codes. Consider achieving a rate of 3/4. The rate of 1/2 was achieved using FC_2.

In C_4 generated by a define $\alpha = 1 + a$ and $\alpha_0 = 1$. Then $\alpha^4 = 0$ and α can be used to define a (linear) code of rate 3/4 and distance 2 in FC_4. Now α has matrix $\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}$ and the first three rows of this $A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$ generate a $(4, 3, 2)$ code.

The matrix of α_0 is $I_{4\times 4}$, the identity 4×4 matrix, and let B denote the first three rows of $I_{4\times 4}$.

Lemma 7.1. *Let $\underline{x} \neq 0$ be a 1×3 vector. Then $\underline{x}(A + B)$ is not the zero vector and thus $\underline{x}(A + B)$ has distance at least* 1.

Proof. Now $(\alpha + 1)^4 = \alpha^4 + 1 = 1$ and so $(\alpha + 1)$ is a non-singular matrix. Thus in particular the first three rows of the matrix of $(\alpha + 1)$ are linearly independent. The first three rows of $\alpha + 1$ precisely constitutes the matrix $A + B$. Thus $\underline{x}(A + B)$ is not the zero vector.

Another way to look at this is that $\alpha + 1 = a$ but it is useful to look at the more general way in Lemma 7.1 for further developments. □

Corollary 7.1. *If $\underline{x}A + \underline{y}B = \underline{0}$ then $\underline{x} \neq \underline{y}$ or $\underline{x} = \underline{0} = \underline{y}$.*

Form convolutional $(4, 3)$ codes as follows.

Let $f(z) = \sum_{i=0}^{n} \alpha_i z^i$ where $\alpha_i = \alpha$ or $\alpha_i = 0$ except for $\alpha_t = 1$ for some $t, 1 < t \leq n$. We could also use $\alpha_1 = \alpha_t = 1$ but this generally gives smaller distance codes.

Then $f(z)^4 = z^{4t}$ and so $f(z) \times (f(z)^3/z^{4t}) = 1$. Thus use $f(z)$ to generate the code and $(f(z)^3/z^{4t})$ to check/control the code. Take the first three rows of the matrix of $f(z)$ to generate a $(4, 3)$ code and delete the last three columns $(f(z)^3/z^{4t})$ to form the control matrix. Thus $G(z) = \sum_{i=0}^{n} \hat{\alpha}_i z^i$ is the generator matrix where $\hat{\alpha}_i$ is the first three rows of the matrix of α_i.

In Section 5 we had the situation that when α_0 occurred in any coefficient then it contributed a distance of 1, so that when the support of G is s then α_0 will contribute a free distance of s. Here we use the fact that if α_0 occurs then it will contribute a distance of at least 1 unless its coefficient equals the sum of the coefficients in the other non-zero α_i which occur with it in the same coefficient of z^j.

7.1. Examples of Higher Rate Convolutional Codes

The generator matrix

$$G = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} z + \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} z^2,$$

defines a $(4, 3)$ convolutional code. It may be shown that its free distance is 5. The proof is similar to the proof of Proposition 5.1 but also using Lemma 7.1.

The check matrix for the code is easy to write out.

Consider $n = 3$, and

$$G = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} z + \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} z^2 + \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} z^3.$$

This is a $(4, 3)$ convolutional code and its free distance is 6.

The next example is

$$G = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} z + \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} z^3 + \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} z^4.$$

This has free distance 7. This may be proved similar to Theorem 5.2 using Lemma 7.1.

It is then possible to proceed as in Section 5 to investigate further degrees (memories) with rate 3/4.

7.2. Polynomial

In cases where a polynomial generator *and* polynomial right inverse for this generator are required, insist that $\alpha_0 = 1$. This gives slightly less free distance but is interesting in itself.

For example consider the encoder matrix $G = (1, 0) + \delta_1(1, 1)z + \ldots +$

$\delta_n(1,1)z^n$ and the control matrix is $H = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \delta_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix} z + \ldots + \delta_n \begin{pmatrix} 1 \\ 1 \end{pmatrix} z^n$. Here $\delta_i = 0$ or $\delta_i = 1$.

This code has free distance 4 for $n = 2$. For $n \geq 2$ the free distance will depend on the choice of the δ_i. As already noted, the choices where the z-polynomial corresponds to a known cyclic code generator polynomial deserve particular attention.

We may also increase the size of the field as for example as follows.

Consider now $R = GF(4)C_2$, the group ring of the cyclic group of order 2 over the field of 4 elements. Define $\alpha_0 = \omega + \omega^2 g$, $\alpha_1 = \omega + \omega g, \alpha_2 = \omega^2 + \omega^2 g$, where ω is the primitive element in $GF(4)$ which satisfies $\omega^2 + \omega + 1 = 0, \omega^3 = 1$. Then $\alpha_0^2 = \omega^2 + \omega^4 = \omega^2 + \omega = 1$ and $\alpha_1^2 = \alpha_2^2 = 0$. Thus $w = \alpha_0 + \alpha_1 z + \alpha_2 z^2$ satisfies $w^2 = 1$ and can be used to define a convolutional code of length 2 and dimension 1. The encoder matrix is then $G = (\omega, \omega^2) + \delta_1(\omega, \omega)z + \delta_2(\omega^2, \omega^2)z^2 + \ldots + \delta_n(\omega^i, \omega^i)z^n$ and the control matrix is $H = \begin{pmatrix} \omega^2 \\ \omega \end{pmatrix} + \delta_1 \begin{pmatrix} \omega \\ \omega \end{pmatrix} z + \ldots + \delta_n \begin{pmatrix} \omega^i \\ \omega^i \end{pmatrix} z^n$.

The *degree* of a convolutional code with encoder matrix $G(z)$ is defined to be the maximal degree of the full $k \times k$ size minors of $G(z)$ where k is the dimension; see [1]. The maximum free distance of a length 2, dimension one, degree δ code over any field is by [12], $2\delta + 2$.

Consider the case $n = 2$. The encoder matrix is then $G = (\omega, \omega^2) + (\omega, \omega)z + (\omega^2, \omega^2)z^2$. The degree of this code is $\delta = 2$ since the dimension is 1. Let $G' = (1, \omega) + (1, 1)z + (\omega, \omega)z^2$ so that $\omega G' = G$.

Theorem 7.1. *The free distance of this code is* 6 *and so is thus a maximum distance separable convolutional code.*

Proof. Consider combinations $(\alpha_0 + \alpha_1 z + \ldots + \alpha_t z^t)G$ and we wish to show that this has (free) distance 6. We may assume $\alpha_0 \neq 0$. It is clear when $t = 0$ that w has a distance of 6 and so in particular a distance of 6 is attained. Since also ω is a factor of G we may now consider the minimum distance of $w = (\alpha_0 + \alpha_1 z + \ldots + \alpha_t z^t)G'$ with $\alpha_0 \neq 0, \alpha_t \neq 0$ and $t > 0$. The coefficient of z^0 is $\alpha_0(1, \omega)$; the coefficient of z^{t+2} is $\alpha_t(\omega, \omega)$, the coefficient of z^{t+1} is $\alpha_t(1, 1) + \alpha_{t-1}(\omega, \omega)$ and the coefficient of z^t is $\alpha_t(1, \omega) + \alpha_{t-1}(1, 1) + \alpha_{t-2}(\omega, \omega)$ when $t \geq 2$ and the coefficient of z is $\alpha_1(1, \omega) + \alpha_0(1, 1)$ and this is also the case when $t = 1$.

Case $t \geq 2$. If $\alpha_t \neq \alpha_{t-1}\omega$ then the coefficient of z^{t+1} has distance 2 giving a distance of 6 with 2 coming from each of the coefficients of z^0, z^{t+1}, z^{t+2}. If

$\alpha_t = \alpha_{t-1}\omega$ the coefficient of z^t is $\alpha_{t-1}(\omega+1, \omega^2+1) + \alpha_{t-1}(\omega, \omega)$; in any case this has distance ≥ 1. Also the coefficient of z has distance ≥ 1. Thus the total distance is at least $2+1+1+2=6$.

Case $t=1$. If $\alpha_0\omega \neq \alpha_1$ then the coefficient of z^2 has distance 2 and thus get a distance of $2+2+2=6$ for the coefficients of z^0, z^2, z^3. If $\alpha_0\omega = \alpha_1$ then the coefficient of z is $\alpha_1(1,\omega) + \alpha_0(1,1) = \alpha_0(\omega+1, \omega^2+1)$ which has distance 2. Thus also we get a distance of $2+2+2=6$ from coefficients of z^0, z, z^3.

Note that the proof depends on the fact that $\{1, \omega\}$ is linearly independent in $GF(4)$. □

8. General Rank Considerations

Let $w(z) = \sum_{i=0}^{t} \alpha_i z^i$ where $\alpha_i^2 = 0, i \neq t, \alpha_t^2 = 1$ with the α_i in some group ring RG. Suppose the α_i commute and that R has characteristic 2. Then $w(z)^2 = z^{2t}$.

Consider the ranks of the non-zero α_i in deciding which rows of w to choose with which to construct the convolutional code. For example if the non-zero α_i satisfy $\operatorname{rank} \alpha_i = 1/2|G| = m$ we choose the matrix with just half the rows of the matrix of each α_i.

Many good codes may be produced this way.

It is possible to have more than one α_t satisfying $\alpha_t^2 = 1$ in $w(z)$ but then the generator matrix produced can be catastrophic, although a valid code may still be defined.

8.1. Example

Let $u = 1 + h(a + a^2 + a^3)$ in $\mathbb{Z}_2(C_4 \times C_2)$.

Then $u^2 = 0$ and $\operatorname{rank} u = 4$. Define $w = u + z + uz^2$. Then $w^2 = z^2$ and w is used to define a $(8,4)$ convolutional code. The generator matrix is $G = (I,B) + (I,0)z + (I,B)z^2$ where $B = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$. Now the code generated by (I,B) has distance 4. Any combination of $(I,B), (I,0)$ has distance 1 at least

as B is non-singular. Thus consider $(\sum_{i=0}^{t} \beta_i z^t)G$. The highest and lowest power of z has distance 4 and there is a power of z in between which has distance 1 so altogether we get a free distance of 9. The degree of the code is 8.

This can be extended. It can also be extended by finding higher dimensional u with $u^2 = 0$. See Section 11 for further development of these ideas.

8.2. Higher Rates with Nilpotent Elements

Elements with α_i with $\alpha_i^2 = 0$ generally give rate 1/2 convolutional codes. Now look at elements α with $\alpha^4 = 0$ with which to produce convolutional rate 3/4 codes. See Section 7 for some preliminary examples on these.

Suppose then $w = \sum_{i=0}^{n} \alpha_i z^i$ in $\mathbb{F}G$ where $\alpha_i^4 = 0, i \neq t$ and $\alpha_t^4 = 1, 1 \leq t \leq n$. Suppose also $\mathbb{F}$ has characteristic 2 and that the α_i commute. Then $w^4 = z^{4t}$. Thus w is used to generate a 3/4 rate convolutional code by taking the first 3/4 of the rows of the α_i; then w^3/z^{4t} will be the control matrix using the last 1/4 of the columns of the α_i.

Consider as an example $\alpha = a + a^7 \in \mathbb{Z}_2 C_8$. Then $\alpha_i^4 = 0$ and α generates an $(8, 6, 2)$ linear cyclic code – this is the best distance for a linear $(8, 6)$ code. Now construct convolutional codes similar to the construction of the $(2, 1)$ codes.

An element $\alpha_0 \in \mathbb{Z}_2 C_8$ such that $\alpha_0^4 = 1$ is needed. There are a number of choices including $\alpha_0 = 1, \alpha_0 = 1 + a + a^3, \alpha_0 = 1 + a + a^7$. Choose α_0 so that the first 3 rows of the matrix of α_0 generate a linear code of largest distance. It is easy to verify that the first six rows of $\alpha_0 = 1 + a + a^3$ generate a linear code of distance 2.

— $w = \alpha + \alpha_0 z$. This gives a $(8, 6)$ code of free distance 4. The 'degree' in the convolutional sense is 6.

— $w = \alpha + \alpha_0 z + \alpha z^2$. This is a $(8, 6)$ convolutional code of free distance 6. The 'degree' here is 12.

— $w = \alpha + \alpha_0 z + \alpha z^2 + \alpha z^3$ gives an $(8, 6)$ code of free distance 6.

— $w = \alpha + \alpha_0 z + \alpha z^3 + \alpha z^4$ gives an $(8, 6)$ code of free distance 8.

— Polynomial degree 5: $w = \alpha + \alpha_0 z + \alpha z^3 + \alpha z^4 + \alpha z^5$. The free distance has to be determined.

— Polynomial degree 6: $w = \alpha + \alpha z^2 + \alpha z^3 + \alpha_0 z^4 + \alpha z^5 + \alpha z^6$. This should give a free distance of at least 10.

— As for the $(2,1)$ convolutional codes in Section 5, by mimicking the polynomials used to generate cyclic codes, it should be possible to get $(8,6)$ convolutional codes with increasing free distance.

9. Using Idempotents to Generate Convolutional Codes

Let FG be the group ring over a field F. For this section it is required that *orthogonal systems of idempotents* exist in FG and this places restrictions on thre field as for example that $\operatorname{char} F \not| |G|$ and it may also be necessary to require that F contains a primitive n^{th} root of unity.

The reader is (again) referred to [10] for background definitions and results on group rings in relation to this section.

Let $\{e_1, e_2, \ldots, e_k\}$ be a complete family of orthogonal idempotents in FG. Such sets always exist when $\operatorname{char} F \not| |G|$.

Thus:

(i) $e_i \neq 0$ and $e_i^2 = e_i$, $1 \leq i \leq k$.

(ii) If $i \neq j$ then $e_i e_j = 0$.

(iii) $1 = e_1 + e_2 + \ldots + e_k$.

Here 1 is used for the identity of FG.

Proposition 9.1. *Let* $f(z) = \sum_{i=0}^{k} \pm e_i z^{t_i}$. *Then* $f(z)f(z^{-1}) = 1$.

Proof. Since $e_1, e_2, \ldots, e_k$ is a set of orthogonal primitive idempotents, $f(z)f(z^{-1}) = e_1^2 + e_2^2 + \ldots + e_k^2 = 1$. □

As already pointed out, the variable z in $f(z)$ can be thought of as an element in C_∞ and thus Proposition 9.1 is an identity in RC_∞ wherein $R = FG$ is a group ring.

To now construct convolutional codes, decide on the rank r and then use the first r rows of the matrices of the e_i in Proposition 9.1. The control matrix is obtained from $f(z^{-1})$ by deleting the last r columns of the e_i.

The rank r is often chosen by reference to the ranks of the matrices of the e_i. If the e_i have rank $\geq k$, and for some i $\operatorname{rank} e_i = k$, then it is probably best to take the $r = k$ for the rank of the convolutional code, although other cases also have uses depending on the application in mind.

9.1. Idempotents in Group Rings

Orthogonal sets of idempotents may be obtained in group rings from the conjugacy classes and character tables, see e.g. [10].

Notice also that a product $h(z) = \prod_i f_i(z)$ where the $f_i(z)$ satisfy the conditions of Proposition 9.1 also satisfies $h(z)\hat{h}(z^{-1}) = 1$, where $\hat{h}(z^{-1})$ is the product of the $f_i(z^{-1})$ in reverse order, and thus $h(z)$ can then be used to define convolutional codes.

In the ring of matrices define e_{ii} to be the matrix with 1 in the i^{th} diagonal and zeros elsewhere. Then $e_{11}, e_{22}, \ldots, e_{nn}$ is a complete set of orthogonal idempotents and can be used to define such $f(z)$. These in a sense are trivial but can be useful when combined with others as for example they are usual in filter banks.

To construct convolutional codes:

— Find sets of orthogonal idempotents.

— Decide on the $f(z)$ to be used with each set.

— Take the product of the $f(z)$.

— Decide on the rate.

— Convert these idempotents into matrices as per the isomorphism between the group ring and a ring of matrices.

Group rings are a rich source of complete sets of orthogonal idempotents. This brings *character theory* in group rings into consdieration.

The computer algebra packages *GAP* and *Magma* can construct character tables and conjugacy classes from which complete sets of orthogonal idempotents may be obtained.

9.2. Examples

Consider $\mathcal{C}C_2$ where C_2 is generated by a. Define $e_1 = \frac{1}{2}(1 + a)$ and $e_2 = 1 - e_1 = \frac{1}{2}(1 - a)$. This gives $f(z) = e_1 + e_2 z^t$ or $f(z) = e_2 + e_1 z^t$ for various t. Products of these could also be used but in this case we get another of the same form by a power of z.

Idempotents from Cyclic Groups. The orthogonal idempotents and character table of the cyclic group are well-known and are closely related to the Fourier matrix, [10].

This gives for example in C_4, $e_1 = \frac{1}{4}(1 + a + a^2 + a^3), e_2 = \frac{1}{4}(1 + \omega a + \omega^2 a^2 + \omega^3 a^3), e_3 = \frac{1}{4}(1 - a + a^2 - a^3), e_4 = \frac{1}{4}(1 + \omega^3 a + \omega^2 a^2 + \omega a^3)$ from which 4×4 matrices with degree 4 in z may be constructed, where ω is a primitive 4-th root of unity. Notice in this case that $\omega^2 = -1$.

Let $f(z) = e_1 + e_2 z + e_3 + e_4 z^3$. Then $f(z)f(z^{-1}) = 1$. We take the first row of the matrices to give the following generator matrix for a $(4, 1, 3)$ convolutional code:

$G(z) = \frac{1}{4}\{(1, 1, 1, 1) + (1, \omega, -1, -\omega)z + (1, -1, 1, -1)z^2 + (1, -\omega, -1, \omega)z^3\}$.

It is easy to check that a combination of any one, two or three of the vectors $(1, 1, 1, 1), (1, \omega, -1, -\omega), (1, -1, 1, -1), (1, -\omega, -1, \omega)$, which are the rows of the Fourier matrix, has distance at least 2 and a combination of all four of them has distance 1. From this it is easy to show that the code has free distance 14 – any combination of more than one will have 4 at each end and three in the middle with distance at least 2. This gives a $(4, 1, 3, 14)$ convolutional code which is optimal – see [12].

The e_i can be combined to obtain real sets of orthogonal idempotents and it is enough to combine the conjugacy classes of g and g^{-1} to obtain these.

Combining the classes of g and g^{-1} in this case then gives:

$\hat{e}_1 = e_1 = \frac{1}{4}(1+a+a^2+a^3), \hat{e}_2 = e_2+e_4 = \frac{1}{2}(1-a^2), \hat{e}_3 = e_3 = \frac{1}{4}(1-a+a^2-a^3)$,

which can then be used to construct real convolutional codes.

Then $G(z) = \frac{1}{4}\{(1, 1, 1, 1)+2(2, 0, -2, 0)z+(1, -1, 1, -1)z^2\}$ gives a $(4, 1, 2)$ convolutional code. Its free distance is 10 which is also optimal.

Using $C_2 \times C_2$ gives different matrices. Here the set of orthogonal idempotents consists of $e_1 = \frac{1}{4}(1 + a + b + ab), e_2 = \frac{1}{4}(1 - a + b - ab), e_3 = \frac{1}{4}(1 - a - b + ab), e_4 = \frac{1}{4}(1 + a - b - ab)$ and the matrices derived are all real.

This gives $G(z) = \frac{1}{4}\{(1, 1, 1, 1) + (1, -1, 1, -1)z + (1, -1, -1, 1)z^2$
$+ (1, 1, -1, -1)z^3\}$. Its free distance is also 14.

9.3. Symmetric Group

The orthogonal idempotents of the symmetric group are well-understood and are real.

We present an example here from S_3, the symmetric group on 3 letters.

Now $S_3 = \{1, (1, 2), (1, 3), (2, 3), (1, 2, 3), (1, 3, 2)\}$ where these are cycles.

We also use this listing of S_3 when constructing matrices.

There are three conjugacy classes: $K_1 = \{1\}$; $K_2 = \{(1,2),(1,3),(2,3)\}$; $K_3 = \{(1,2,3),(1,3,2)\}$.

Define:
$\hat{e}_1 = 1 + (1,2) + (1,3) + (2,3) + (1,2,3) + (1,3,2)$,
$\hat{e}_2 = 1 - \{(1,2) + (1,3) + (2,3)\} + (1,2,3) + (1,3,2)$,
$\hat{e}_3 = 2 - \{(1,2,3) + (1,3,2)\}$,
and
$e_1 = \frac{1}{6}\hat{e}_1; \quad e_2 = \frac{1}{6}\hat{e}_2; \quad e_3 = \frac{1}{3}\hat{e}_3$.

Then $\{e_1, e_2, e_3\}$ form a complete orthogonal set of idempotents and may be used to construct convolutional codes.

The G-matrix of S_3 (see [4]) is

$$\begin{pmatrix} 1 & (12) & (13) & (23) & (123) & (132) \\ (12) & 1 & (132) & (123) & (23) & (13) \\ (13) & (123) & 1 & (132) & (12) & (23) \\ (23) & (132) & (123) & 1 & (13) & (12) \\ (132) & (23) & (12) & (13) & 1 & (123) \\ (123) & (13) & (23) & (21) & (132) & 1 \end{pmatrix}.$$

Thus the matrices of e_1, e_2, e_3 are respectively

$$E_1 = \frac{1}{6}\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix},$$

$$E_2 = \frac{1}{6}\begin{pmatrix} 1 & -1 & -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 & 1 & 1 \end{pmatrix},$$

$$E_3 = \frac{1}{3}\begin{pmatrix} 2 & 0 & 0 & 0 & -1 & -1 \\ 0 & 2 & -1 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 2 & 0 & 0 \\ -1 & 0 & 0 & 0 & 2 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 \end{pmatrix}.$$

Note that e_1, e_2 have rank 1 and that e_3 has rank 2.

10. Other Characteristics

Convolutional codes over fields of arbitrary characteristic may also be constructed using the general method. Characteristic 2 were used mostly to construct rate 1/2 codes as matrices derived had rank less than or equal to half the size of the matrices. Using higher characteristics enables the construction of higher rate codes as matrices derived are then of bigger rank. In characteristic p matrices of ranks $|G|(\frac{p-1}{p})$ can be derived as coefficients and thus rates of $\frac{p-1}{p}$ are achievable. For example in characteristic 3 rates of $\frac{2}{3}$ are achievable and examples of these are given below.

The following lemma is similar to Lemma 4.2; its proof is straight-forward.

Lemma 10.1. *Let $R = FG$ be the group ring of a group G over a field F with characteristic p. Suppose $\alpha_i \in R$ commute. Let $w = \sum_{i=0}^{n} \alpha_i z^i \in RC_\infty$. Then $w^p = z^{pt}$ if and only if $\alpha_i^p = 0, i \neq t$ and $\alpha_t^p = 1$.*

Construct convolutional codes as follows. Find elements α_i with $\alpha_i^p = 0$ and units u with $u^p = 1$ in the group ring R. Then with Lemma 10.1 in mind find $f(z) \in R[z]$ such that $f(z)^p = z^{pt}$. It follows that $f(z) \cdot f(z)^{p-1}/(z^{pt}) = 1$.

From f and its inverse $f(z)^{p-1}/z^{pt}$ convolutional codes are defined as described in Section 2.1.

So for example choosing the first r rows of the α_i considered as matrices defines a (n, r) convolutional code where $n = |G|$. The generator matrix is $\hat{f}(z) = \sum_{i=0}^{n} \hat{\alpha_i} z^i$ where $\hat{\alpha_i}$ denotes the first r rows of the matrix of α_i.

It is necessary to decide how many and which rows of the matrix to choose in defining the convolutional code. This is often decided by considering the ranks of the non-zero α_i.

10.1. Examples for Characteristic 3

Suppose then F has characteristic 3 and consider $F(C_3 \times C_3)$ where the C_3 are generated respectively by g, h.

Define $\alpha = 1 + h(1 + g)$. Then $\alpha^3 = 0$. Define $\alpha_0 = 2 + 2h$. Then $\alpha_0^3 = 1$.

The matrix of α is $P = \begin{pmatrix} I & B & 0 \\ 0 & I & B \\ B & 0 & I \end{pmatrix}$ where I is the identity 3×3 matrix, 0 is the zero 3×3 matrix and $B = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$.

By row (block) operations P is equivalent to $\begin{pmatrix} I & 0 & -B^2 \\ 0 & I & B \\ 0 & 0 & 0 \end{pmatrix}$. Thus P has rank 6 and the matrix $Q = \begin{pmatrix} I & 0 & -B^2 \\ 0 & I & B \end{pmatrix}$ defines a block $(9,6)$ code. This block code has distance 3 which is maximum for a $(9,6)$ code over $\mathbb{Z}_3$.

Now define $\alpha_t = \alpha_0$ for some $0 < t < n$ and for $0 \leq i \leq n$ choose $\alpha_i = 0$ or $\alpha_i = \alpha$ for $i \neq t$. Define $f(z) = \sum_{i=0}^{n} \alpha_i z^i$. Then by Lemma 10.1, $f(z)^3 = z^{3t}$ and hence $f(z) \cdot (f(z)^2/z^{3t}) = 1$. Thus $f(z)$ may be used to define a convolutional code. Choose the first 6 rows of the α_i in $f(z)$ to define the code and thus we get a $(9,6)$ convolutional code. The generator matrix is $\hat{f}(z) = \sum_{i=0}^{n} \hat{\alpha_i} z^i$ where $\hat{\alpha_i}$ denotes the first 6 rows of α_i, considered as a matrix.

The control matrix is obtained from $f(z)^2/z^{3t}$ using the last 3 columns of the α_i.

As a specific example in characteristic 3 consider the following. Define $f(z) = \alpha + \alpha_0 z + \alpha z^2$. Then $\hat{f}(z) = \hat{\alpha} + \hat{\alpha_0} z + \hat{\alpha} z^2$ is a convolutional $(9,6)$ code of free distance 8.

Define $f(z) = \alpha + \alpha_0 z + \alpha z^3 + \alpha z^4$. Then $\hat{f}(z) = \hat{\alpha} + \hat{\alpha_0} z + \hat{\alpha} z^3 + \hat{\alpha} z^4$ defines a $(9,6)$ convolutional code which has free distance 11.

The free distances for these may be proved algebraically. It is useful to note that $\underline{x}\hat{\alpha_i} + \underline{y}\hat{\alpha_0}$ has distance at least 1 for 1×6 vectors $\underline{x}, \underline{y}$ with $\underline{y} \neq \underline{0}$, so that only a small number of combinations need be considered.

A result similar to Theorem 5.2 can also be proved.

Suppose now $\mathcal{C}$ is a cyclic (n, k, d_1) code over the field F of characteristic 3. Suppose also that the dual of $\mathcal{C}$, denoted $\hat{\mathcal{C}}$, is an $(n, n-k, d_2)$ code.

Let $d = \min(d_1, d_2)$. Suppose $f(g) = \sum_{i=0}^{r} \beta_i g^i$, with $\beta_i \in F, (\beta_r \neq 0)$, is a generating polynomial for $\mathcal{C}$. In $f(g)$, assume $\beta_0 \neq 0$.

Consider $f(z) = \sum_{i=1}^{r} \alpha_i z^i$ where now $\alpha_i = \beta_i \alpha$ with α as above in $F(C_3 \times C_3)$. Note that if $\beta_i = 0$ then $\alpha_i = 0$. Replace some α_i, say α_t, by α_0 (considered as members of $F(C_3 \times C_3)$).

So assume $f(z) = \sum_{i=0}^{r} \alpha_i z^i$ with this $\alpha_t = \alpha_0$ and other $\alpha_i = \beta_i \alpha$ (for $i \neq t$).

Then $f(z)^3 = \beta_t^3 z^{3t}$ gives that $f(z) \cdot (f(z)^2/(\beta_t^3 z^{3t}) = 1$. We now use $f(z)$ to generate a convolutional code by taking the first 6 rows of the α_i. Thus the generating matrix is $\hat{f}(z) = \sum_{i=0}^{r} \hat{\alpha}_i \beta_i z^i$ where $\hat{\alpha}_i$ consists of the first 6 rows of α for $i \neq t$ and $\hat{\alpha}_t$ consists of the first 6 rows of α_0.

For the following theorem assume the invertible element α_0 does not occur in the first or the last position of f. The proof of the following is similar to the proof of Theorem 5.2 and is omitted.

Theorem 10.1. *Let $\mathcal{C}$ denote the convolutional code with generator matrix $\hat{f}(z)$. Then the free distance of $\mathcal{C}$ is at least $d + 4$.*

11. Convolutional Codes with Hamming Matrices

Set $R = \mathbb{Z}_2(C_4 \times C_2)$. Suppose C_4 is generated by a and C_2 is generated by h. Consider $\alpha_0 = 1 + h(1 + a^2)$ and $\alpha_i = 1 + h(a + a^2 + a^3)$ or $\alpha_i = 0$ for $i > 0$. Then $\alpha_0^2 = 1$ and $\alpha_i^2 = 0$. Define $w(z) = \sum_{i=0}^{n} \alpha_i z^i$ in RC_∞. By Lemma 4.1, $w^2 = 1$.

Let $A = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$, $B = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$ and I is the identity 4×4 matrix. The matrix corresponding to α_0 is then $\begin{pmatrix} I & A \\ A & I \end{pmatrix}$ and the matrix corresponding to α_i, $i \neq 0$, is either $\begin{pmatrix} I & B \\ B & I \end{pmatrix}$ or the zero matrix.

Now specify that the first 4 rows of w formulate the generator matrix of a code and then the last four columns of w formulate the control matrix. This gives a convolutional code of length 8 and dimension 4. It is easy to transform

the resulting code into a systematic code.

The generator matrix is $G(z) = (I, A) + \delta_1(I, B)z + \delta_2(I, B)z^2 + \ldots + \delta_n(I, B)z^n$, where $\delta_i \in \{0, 1\}$. The control matrix is

$$H(z) = \begin{pmatrix} A \\ I \end{pmatrix} + \delta_1 \begin{pmatrix} B \\ I \end{pmatrix} z + \delta_2 \begin{pmatrix} B \\ I \end{pmatrix} z^2 \ldots + \delta_n \begin{pmatrix} B \\ I \end{pmatrix} z^n.$$

The (I, A) may be moved to the coefficient of any z^i in which case the (natural) control matrix will need to be divided by a power of z to get the true control matrix.

This convolutional code may be considered as a Hamming type convolutional code as (I, B) is a generator matrix of the Hamming $(8, 4)$ code.

For $n = 1$ the free distance turns out to be 6; this can be proved in a similar manner to Theorem 7.1.

Example. $G(z) = (I, B) + (I, A)z + (I, B)z^2$ with control matrix $H(z)/z^2$ where $H(z) = \begin{pmatrix} B \\ I \end{pmatrix} + \begin{pmatrix} A \\ I \end{pmatrix} z + \begin{pmatrix} B \\ I \end{pmatrix} z^2$ has free distance 10.

11.1. From Cyclic to Hamming Type

For $n \geq 2$, proceed as previously to define the polynomials by reference to corresponding cyclic linear polynomials. This will give convolutional codes of this type of increasing free distance. Note that (I, A) has distance 2, (I, B) (the Hamming code) has distance 4, any combination of (I, A) and (I, B) has distance ≥ 1.

Suppose now $\mathcal{C}$ is a cyclic (n, k, d_1) code over the field F of characteristic 2 and that the dual of $\mathcal{C}$, $\hat{\mathcal{C}}$, is an $(n, n-k, d_2)$ code. Let $d = \min(d_1, d_2)$. Assume $f(g) = \sum_{i=1}^{r} \beta_i g^i$ is a generator polynomial for $\mathcal{C}$. In $f(g)$, it is possible to arrange that $\beta_0 \neq 0$ and naturally assume that $\beta_r \neq 0$. Define $f(z) = \sum_{i=1}^{r} \alpha_i z^i$ with the $\alpha_i = \beta_i \alpha$, $i \neq t$ and $\alpha_t = \alpha_0$. Then $f(z)^2 = z^{2t}$ giving $f(z) \times f(z)/z^{2t} = 1$. Now use $f(z)$ to generate a convolutional code by taking just the first four rows of the α_i. Thus the generating matrix is $G = \sum_{i=0}^{r} \hat{\alpha}_i z^i$ where $\hat{\alpha}_i$ consists of the first four rows of the matrix of α_i. It may be shown similar to Theorem 5.2 that this code has free distance at least $d + 8$.

References

[1] Richard E. Blahut, *Algebraic Codes for Data Transmission*, Cambridge University Press (2003).

[2] Gluesing-Luerssen, Heide and Schmale, Wiland, On cyclic convolutional codes, *Acta Applicandae Mathematicae*, **82**, No. 2 (2004), 183-237.

[3] Paul Hurley, Ted Hurley, Codes from zero-divisors and units in group rings, *ArXiv:* 0710.5893, *IJICoT*, To Appear.

[4] Ted Hurley, Group rings and rings of matrices, *Inter. J. Pure and Appl. Math.*, **31**, No. 3 (2006), 319-335.

[5] Ted Hurley, Self-dual, dual-containing and related quantum codes from group rings, *ArXiv:* 0711.3983.

[6] Paul Hurley, Ted Hurley, Module codes in group rings, *ISIT2007*, Nice (2007), 1981-1985.

[7] J.H. van Lint, R.M. Wilson, *A Course in Combinatorics*, Cambridge University Press (2001).

[8] J.L. Massey, D.J. Costello, J. Justesen, Polynomial weights and code constructions, *IEEE Trans. Inform. Theory*, **19**, No. 1 (1973), 101-110.

[9] R.J. McEliece, The algebraic theory of convolutional codes, In: *Handbook of Coding Theory*, Volume I, North Holland, Elsevier Science (1998).

[10] César Milies, Sudarshan Sehgal, *An Introduction to Group Rings*, Klumer (2002).

[11] David J.C. MacKay, *Information Theory, Inference and Learning Algorithms*, Cambridge University Press (2003).

[12] J. Rosenthal, R. Smarandache, Maximum distance separable convolutional codes, *Appl. Algebra Engrg. Comm. Comput.*, **10**, No. 1 (1999), 15-37.

[13] R. Smarandache, H. Gluesing-Luerssen, J. Rosenthal, Constructions for MDS-convolutional codes, *IEEE Trans. Inform. Theory*, **47** (2001), 2045-2049.

[14] F.J. MacWilliams, N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North Holland (1977).