Locally Invertible m-Dimensional Convolutional Codes

R. Lobo
Dept. of Elec. and Comp. Eng.
North Carolina State University
Raleigh, NC 27695

D. Bitzer
Dept. of Comp. Sci.
North Carolina State University
Raleigh, NC 27695

M. Vouk
Dept. of Comp. Sci.
North Carolina State University
Raleigh, NC 27695

***Abstract*— A new approach for decoding multidimensional (m-D) nonsystematic convolutional codes is described. The possibility of coding at different rates along the dimensions of the sequence space is explored and the notion of ordering of information and encoded sequences based on coding rate is explained. Using this technique, a class of fast-decodeable, locally invertible m-D convolutional codes based on the concept of one-to-one mapping between information and encoded subsequences of equal size is defined. Error syndrome generation using orthogonal parity-check sums is described and methods for error correction using error syndromes are suggested.**

I. Introduction

One dimensional convolutional codes have been widely implemented and are well suited for error control applications. By encoding data that is recorded geometrically in dimensions higher than one, it is possible to make the encoding scheme shift-invariant with respect to axis of the dimensions. This would lead to a convolutional code represented as a polynomial ring in several variables.

Fornasini and Valcher [1] describe 2-D convolutional codes with finite support using the ring $R[z_1^{\pm 1}, z_2^{\pm 1}]$. They define the code as a submodule of the Laurent polynomial ring R^n, where the code can be viewed as a set of sequences indexed on the discrete plane $\mathbb{Z} \times \mathbb{Z}$. Weiner [2], [3], [4] defines a m-D convolutional code as a submodule of the ring $R[z_1, \dots, z_m]$ and the code can be viewed as a set of sequences indexed on the nonnegative integer lattice $\mathbb{N}^m$. The framework for studying m-D convolutional codes by the above authors uses a module-theoretic approach and considers polynomial encoders, equivalent generators, syndrome formers and polynomial inverters. In this paper we introduce the notion of ordering m-D sequences based on the coding rate and describe a new class of nonsystematic *m-D locally invertible convolutional codes* that can be encoded by convolving data at different rates along the m-dimensions. The error syndromes are generated using orthogonal parity vectors and decoding is performed as a separate step after error correction.

II. Notation and Definitions

Multidimensional information can be represented geometrically as a *sequence space* in m-dimensions (m-D) or *algebraically* as a polynomial in m variants. Let $\mathbb{F} = \mathbb{F}_q$ be a finite field with q elements and $R = \mathbb{F}[z_1, \dots, z_m]$ be a polynomial ring in m variants over $\mathbb{F}$. Let the R-module R^k, be the m-D information sequence space of length k over $\mathbb{F}$. If we encode elements of R^k into codewords contained in $R^n, n > k$, then decoding is possible only if the map from $R^k \longrightarrow R^n$ is injective.

Definition 1: [3], [2] Let $G \in R^{k \times n}$ be of rank k. If we consider the code $\mathcal{C} = \mathcal{C}(G)$ as a row space over R of the polynomial matrix G, then G is called the *encoder* or the *generator matrix* and the *rate* of $\mathcal{C}$ is k/n.

A. Code Rate and m-D Representation

It is interesting to note that when $m > 1$, there is no natural notion of causality between the code rate and rate of convolution along the m-dimensions. For example, when encoding a 2-D sequence with a code rate of $6/15$, it is not clear if the convolution proceeds at the rate of 6 in the horizontal (z_1) or vertical (z_2) direction. Furthermore it raises the question of having different convolution rates along the different dimensions. ie. rate 2 along z_1 and rate 3 along z_2 with an effective data rate of 6.

Proposition 1: When specifying the code rate R, the notation

$$R = k/n = k_1/n_1 \times k_2/n_2 \times \cdots \times k_m/n_m \qquad (1)$$

specifies the rate of convolution along the dimensions $z_1, \dots, z_m$ of the m-D sequence space, and the values $k_1, \dots, k_m$ and $n_1, \dots, n_m$ define the ordering of the data and encoded sequence space.

When encoding m-D information, the sequence space and algebraic representation depend on the code rate specified by proposition 1. In a finite sequence space $\mathcal{S}^k$, elements of $\mathbb{F}$ can be viewed as being attached to points of a m-D nonnegative integer lattice $\mathbb{N}^m$. $\mathcal{S}^k$ can be represented as a polynomial vector $u \in R^k$, by associating the points of $\mathbb{N}^m$ with monomials in R via the correspondence $(i_1, \dots, i_m) \longleftrightarrow z_1^{i_1} \cdot \ldots \cdot z_m^{i_m}$ and the j^{th} element of u, $j = 1, \dots, k$, is the summation of all the j^{th} terms of the sequence space $\mathcal{S}^k$.

Example 1: Consider a 2-D code over $\mathbb{F}_2$. Finite information u can be represented as follows along z_1 and z_2 where the top left point is $(0, 0) \in \mathbb{N}^2$.

- Using a rate $2/6 = 2/6 \times 1/1$ code,

$$u = \begin{array}{c} 01\ \ 10 \\ 00\ \ 01 \\ 00\ \ 10 \\ 10\ \ 01 \end{array} = \begin{bmatrix} z_1 + z_1 z_2^2 + z_2^3 \\ 1 + z_1 z_2 + z_1 z_2^3 \end{bmatrix}^T, \tag{2}$$

  where u is ordered using $k_1 = 2$ and $k_2 = 1$ along z_1 and z_2.
- Using a rate $2/6 = 1/1 \times 2/6$ code,

$$u = \begin{array}{cccc} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{array} = \begin{bmatrix} z_1 + z_1^2 + z_1^2 z_2 \\ z_1^3 + z_2 + z_1^3 z_2 \end{bmatrix}^T, \tag{3}$$

  where u is ordered using $k_1 = 1$ and $k_2 = 2$ along z_1 and z_2.

B. Multidimensional Encoding

In a linear system, the encoding operation involving polynomial multiplication can be replaced by sequence space operations involving convolution.

Example 2: Consider a 2-D convolutional code whose encoder is given by

$$G = \begin{bmatrix} z_1^2 + z_2 + z_1^2 z_2 & 1 + z_1 + z_1^2 + z_2 \\ 1 + z_1^2 + z_2 + z_1 z_2 + z_1^2 z_2 & z_1 + z_1 z_2 + z_1^2 z_2 \\ z_1^2 + z_2 + z_1 z_2 + z_1^2 z_2 & z_1 + z_1 z_2 + z_1^2 z_2 \\ 1 + z_1 + z_2 + z_1^2 z_2 & z_1 + z_1^2 + z_2 + z_1 z_2 + z_1^2 z_2 \\ z_1^2 + z_2 + z_1^2 z_2 & 1 + z_1 + z_2 + z_1 z_2 + z_1^2 z_2 \\ 1 + z_1 + z_1 z_2 + z_1^2 z_2 & 1 + z_1 z_2 \end{bmatrix}^T \tag{4}$$

Rank(G)=2, let $\mathcal{C}$ represent a rate $R = 2/6 = 1/2 \times 2/3$ code. The *generator sequences*[5] $g_i^{(j)}; i = 1 \ldots k, j = 1 \ldots n$ for the code obtained from the G are

$$\begin{aligned} g_1^{(1)} &= \begin{pmatrix} 001 \\ 101 \end{pmatrix} \quad g_2^{(1)} = \begin{pmatrix} 111 \\ 100 \end{pmatrix} \\ g_1^{(2)} &= \begin{pmatrix} 101 \\ 111 \end{pmatrix} \quad g_2^{(2)} = \begin{pmatrix} 010 \\ 011 \end{pmatrix} \\ g_1^{(3)} &= \begin{pmatrix} 001 \\ 111 \end{pmatrix} \quad g_2^{(3)} = \begin{pmatrix} 010 \\ 011 \end{pmatrix} \\ g_1^{(4)} &= \begin{pmatrix} 110 \\ 101 \end{pmatrix} \quad g_2^{(4)} = \begin{pmatrix} 011 \\ 111 \end{pmatrix} \\ g_1^{(5)} &= \begin{pmatrix} 001 \\ 101 \end{pmatrix} \quad g_2^{(5)} = \begin{pmatrix} 110 \\ 111 \end{pmatrix} \\ g_1^{(6)} &= \begin{pmatrix} 110 \\ 011 \end{pmatrix} \quad g_2^{(6)} = \begin{pmatrix} 100 \\ 010 \end{pmatrix} \end{aligned} \tag{5}$$

Using the rates $k_1 = 1$ and $k_2 = 2$ along z_1 and z_2, the composite generator sequence is represented as

$$g = \{g^{(j)}; j = 1 \ldots n\}$$
$$= \begin{pmatrix} 001 \\ 111 \\ 101 \\ 100 \end{pmatrix}, \begin{pmatrix} 101 \\ 010 \\ 111 \\ 011 \end{pmatrix}, \begin{pmatrix} 001 \\ 010 \\ 111 \\ 011 \end{pmatrix}, \begin{pmatrix} 110 \\ 011 \\ 101 \\ 111 \end{pmatrix}, \begin{pmatrix} 001 \\ 110 \\ 101 \\ 111 \end{pmatrix}, \begin{pmatrix} 110 \\ 100 \\ 011 \\ 010 \end{pmatrix} \tag{6}$$

Now consider the input data sequence u from equation 3. The output of the encoder G in the algebraic form is

$$v = u.G$$
$$= \begin{bmatrix} z_1^5 + z_2 + z_1^3 z_2 + z_1^4 z_2 + z_1^5 z_2 + z_2^2 + z_1^2 z_2^2 + z_1^3 z_2^2 + z_1^4 z_2^2 \\ z_1 + z_1^2 + z_1^3 + z_1^2 z_2 + z_1^5 z_2 + z_1 z_2^2 + z_1^3 z_2^2 + z_1^5 z_2^2 \\ z_1^3 + z_1^5 z_2 + z_1^2 z_2 + z_1^3 z_2^2 + z_1^5 z_2^2 \\ z_1 + z_1^3 + z_1^4 + z_1^5 + z_1^2 z_2 + z_1^3 z_2 + z_1^4 z_2 + z_2^2 + z_1 z_2^2 + z_1^3 z_2^2 + z_1^5 z_2^2 \\ z_2 + z_1^2 z_2 + z_1^3 z_2 + z_1^5 z_2 + z_2^2 + z_1 z_2^2 + z_1^3 z_2^2 + z_1^5 z_2 \\ z_1 + z_2 + z_1 z_2^2 + z_1^3 z_2^2 \end{bmatrix}^T \tag{7}$$

The output can also be obtained by using the composite generator sequences from equation 6

$$\begin{aligned} v^{(j)} &= u * g^{(j)}; j = 1 \ldots 6 & (8) \\ v &= (v^{(1)}, v^{(2)}, \ldots, v^{(6)}), & (9) \end{aligned}$$

where * denotes discrete 2-D convolution. The output of the convolution results in the map

$$\begin{array}{c} u \\ \begin{array}{cccc} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{array} \end{array} \xrightarrow{u*g} \begin{array}{c} v \\ \begin{array}{cccccc} 00 & 01 & 01 & 01 & 00 & 10 \\ 00 & 01 & 00 & 11 & 01 & 01 \\ 00 & 01 & 00 & 00 & 00 & 00 \\ \\ 10 & 00 & 01 & 10 & 10 & 11 \\ 00 & 00 & 01 & 01 & 01 & 10 \\ 11 & 00 & 10 & 10 & 00 & 10 \\ \\ 10 & 01 & 10 & 11 & 10 & 01 \\ 01 & 11 & 00 & 11 & 00 & 11 \\ 10 & 11 & 00 & 11 & 00 & 10 \end{array} \end{array},$$

which is equivalent to the sequence space representation of equation 7 ordered using the rates $n_1 = 2$ and $n_2 = 3$ along z_1 and z_2.

III. Locally Invertible Codes

In order to make decoding possible the codewords have to be the *image* of unique information sequences and the encoder map should be injective. Such an encoder is called *globally invertible*. Here we introduce a class of m-D encoders which are invertible in the *local* sense. Locally Invertible 1-D encoders were proposed by Bitzer, Vouk and Dholakia[6], [7], [8], [9]. Here we extend the theory of local invertibility to m-D convolutional codes.

A. One-to-one Mapping

When encoding data in m-D, in order to achieve local invertibility, we have to establish a one-to-one correspondence between *equal m-D subsequences* of data and encoded bits. Consider a rate $R = k_1/n_1 \times \cdots \times k_m/n_m$ code $\mathcal{C}$ with $n_1 > k_1, \ldots, n_m > k_m$. Let $L = l_1 \times \ldots \times l_m$ be the input constraint lengths along $z_1, \ldots, z_m$. Now consider production of encoded bits. The first L data bits produce n encoded bits with order $n_1, \ldots, n_m$ in the $z_1, \ldots, z_m$ dimensions. Each additional shift of k_i on the data bits along z_i will produce n more encoded bits with the same ordering. If we find the *mapping length*[6] as in the 1-D case along each of the dimensions we get

$$w_i = \frac{n_i(l_i - k_i)}{n_i - k_i}; i = 1 \ldots m \tag{10}$$

We now have a one-to-one correspondence of equal m-D subsequences $w = w^{(d)} = w^{(e)}$

$$w = w_1 \times \cdots \times w_m \tag{11}$$

where $w^{(d)}$ data bits map to $w^{(e)}$ encoded bits.

Example 3: Consider the 2-D code with rate $R = 2/6 = 1/2 \times 2/3$ shown in example 2. From the composite generator sequence representation we get $l_1 = 3$ and $l_2 = 4$. From proposition 1 we have $k_1 = 1, k_2 = 2, n_1 = 2$ and $n_2 = 3$. Equation 10 gives us $w_1 = 4, w_2 = 6$ and $w = 4 \times 6$. So we have 4×6 data bits mapping to 4×6 encoded bits.

Now the set of $w = 24$ tuple *basis* data elements as shown below, can be used to obtain the complete encoding map between the data and encoded bits through exclusive-OR($\otimes$) operations.

$$\begin{array}{c} u_1 \\ \begin{matrix} 1&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \end{matrix} \end{array} \xrightarrow{*g} \begin{array}{c} v_1 \\ \begin{matrix} 11 & 00 \\ 11 & 00 \\ 11 & 00 \\ 00 & 00 \\ 00 & 00 \\ 00 & 00 \end{matrix} \end{array}, \quad \ldots, \quad \begin{array}{c} u_{24} \\ \begin{matrix} 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&1 \end{matrix} \end{array} \xrightarrow{*g} \begin{array}{c} v_{24} \\ \begin{matrix} 00 & 00 \\ 00 & 00 \\ 00 & 00 \\ 00 & 10 \\ 00 & 00 \\ 00 & 11 \end{matrix} \end{array} \tag{12}$$

Note that the convolution $*g$ is carried out at the rate $k_1 = 1$ along z_1 and $k_2 = 2$ along z_2 and the map takes into account only the bits corresponding to the 2-D subsequence w obtained during the convolution.

The data sequence u shown below can now be encoded using $\otimes$ operations as follows

$$\begin{array}{c} u = u_1 \otimes u_{10} \otimes u_{24} \\ \begin{matrix} 1&0&0&0 \\ 0&0&0&0 \\ 0&1&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&1 \end{matrix} \end{array} \xrightarrow{*g} \begin{array}{c} v = v_1 \otimes v_{10} \otimes v_{24} \\ \begin{matrix} 11 & 11 \\ 10 & 10 \\ 10 & 10 \\ 01 & 01 \\ 10 & 11 \\ 01 & 00 \end{matrix} \end{array}$$

B. Reduced Encoding Matrix

Close examination of the basis map shown in equation 12 reveals interlaced generator sequences and can be used to define the *reduced encoding matrix* $G'_{w\times w}$.

$$G'_{w\times w} = \begin{bmatrix} v_1 & v_2 & \cdots & v_w \end{bmatrix}^T \tag{13}$$

where each row of $G'_{w\times w}$ is the corresponding basis map from equation 12 written as a $1 \times w$ row vector.

Definition 2: Rate $R = k_1/n_1 \times \cdots \times k_m/n_m$ m-D convolutional encoders with $n_1 > k_1, \ldots, n_m > k_m$ and input constraint length $L = l_1 \times \ldots \times l_m$ along $z_1, \ldots, z_m$ that have a *invertible* reduced encoding matrix $G'_{w\times w}$ are called *locally invertible encoders*.

The encoding and decoding equations can be represented as

$$v_{1\times w} = u_{1\times w} \cdot G'_{w\times w} \tag{14}$$
$$\hat{u}_{1\times w} = \hat{v}_{1\times w} \cdot G'^{-1}_{w\times w} \tag{15}$$

A sliding m-D subsequence of size $w = w_1 \times \ldots \times w_m$ can now be used to encode a w-bit data subsequence into a w-bit encoded subsequence. When the received sequence does not have errors, decoding is performed using equation 15. Note that a shift of $k_1, \ldots, k_m$ on the data sequence space corresponds to a shift of $n_1, \ldots, n_m$ on the encoded sequence space along the $z_1, \ldots, z_m$ dimensions.

C. Error Detection

An important property of encoding using one-to-one mapping is that it produces overlapping bits between successive w-bit subsequences[6]. When there are no errors in the received subsequence $\hat{v}$, the corresponding overlapping bits will be the *same* on the data subsequence $\hat{u}$ and this property can be exploited to detect errors. It should be noted that in m-D sequence space, $m > 1$, more than one overlap is possible.

Example 4: For the 2-D code with rate $R = 2/6 = 1/2 \times 2/3$ shown in example 2. A shift of $n_1 = 2$ and $n_2 = 3$ generates a $n_1 \times n_2$ bit overlap between two $w_1 \times w_2$ received subsequences, which corresponds to a $k_1 \times k_2$ shift on the data sequence space.

$$\begin{array}{c} \hat{v} \\ \begin{matrix} 11 & 11 & \\ 10 & 10 & \\ 10 & 10 & \\ 01 & 01 & 11 \\ 10 & 11 & 11 \\ 01 & 00 & 10 \\ & 10 & 10 \\ & 00 & 01 \\ & 11 & 01 \end{matrix} \end{array} \longrightarrow \begin{array}{c} \hat{u} \\ \begin{matrix} 1&0&0&0& \\ 0&0&0&0& \\ 0&1&0&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \\ 0&0&0&1&0 \\ &0&1&0&0 \\ &0&0&0&0 \end{matrix} \end{array}$$

If the received sequence $\hat{v}$ had an error then the corresponding overlapping bits of $\hat{u}$ would not be the same and thus indicate an error.

1) Syndrome Former (g-mask): From example 4 it is clear that an $(n_1 \times \cdots \times n_m)$-bit encoded subsequence with an encoded bit in error affects at most $(w_1+k_1)\times\cdots\times(w_m+k_m)$ data bits, and the decoding operation specified by equation 15 would yield corrupt data. Since $(w_1+n_1)\times\cdots\times(w_m+n_m)$ encoded bits are needed to produce $(w_1+k_1)\times\cdots\times(w_m+k_m)$ data bits, we need to correct all errors over a subsequence of size $(w_1+n_1)\times\cdots\times(w_m+n_m)$ encoded bits for correct decoding.

The error detection in one-to-one mapped locally invertible convolutional codes is based on the concept of orthogonal parity-check sums and is achieved using the *g-mask* h. A g-mask can be viewed as a finite m-D subsequence of size $(w_1+n_1)\times\cdots\times(w_m+n_m) \in \mathcal{C}^\perp$, where the module $\mathcal{C}^\perp$ is the orthogonal code of $\mathcal{C}$.

Proposition 2: Let $\mathcal{C}$ be a Locally Invertible m-D convolutional code with rate $R = k/n = k_1/n_1 \times k_2/n_2 \times \cdots \times k_m/n_m$. If the one-to-one mapping m-D subsequence is $w = w_1 \times \cdots \times w_m$. Then the maximum number of linearly independent g-masks that detect errors over an encoded sequence of size $(w_1+n_1)\times\cdots\times(w_m+n_m)$ bits is $((w_1+n_1)\times\cdots\times(w_m+n_m))-((w_1+k_1)\times\cdots\times(w_m+k_m))$.

Proof: To produce $v=((w_1+n_1)\times\cdots\times(w_m+n_m))$ encoded bits we need $u=((w_1+k_1)\times\cdots\times(w_m+k_m))$ data bits. Since the Locally Invertible map is injective, the u codewords of size v, generated from the ordered standard basis of data bits of size u, will be form a basis for the encoded space and hence be linearly independent.

Let A be a $u\times v$ matrix where each of the u codewords of size v form a row of A. Since a g-mask h lies in the null space of A, solving for $AX=0$ gives us the number of g-masks $\sharp h$

$$\begin{aligned}\sharp h &= \text{Nullity}(A)=v-\text{rank}(A)=v-u\\ &= ((w_1+n_1)\times\cdots\times(w_m+n_m))-\\ &\quad((w_1+k_1)\times\cdots\times(w_m+k_m)). \qquad (16)\end{aligned}$$

■

Example 5: Consider the code shown in example 3. We have $w=4\times 6$ with $w_1=4$ and $w_2=6$. From equation 16 we get

$$\sharp h=((4+2)\times(6+3))-((4+1)\times(6+2))=14.$$

The 14 g-masks are obtained by taking the $((4+1)\times(6+2))$ linearly independent code words of size $((4+2)\times(6+3))$ generated from the $((4+1)\times(6+2))$ standard data basis and solving for the null space.

2) Error Syndrome: Data independent finite syndromes $s=\{s^{(1)},\ldots,s^{(\sharp h)}\}$ that depend only on the error pattern can be used to detect and correct errors in the encoded stream. Each of the m-D syndrome subsequences is calculated as follows

$$s^{(j)}=\hat{v}*h^{(j)}; j=1\ldots\sharp h \qquad (17)$$

$\hat{v}\in\mathcal{C}$ if $s=0$. Note that since we are working on the encoded sequence space, the discrete convolution $*$ is carried out at the rates $n_1,\ldots,n_m$ along $z_1,\ldots,z_m$ respectively.

Since each g-mask $h^{(j)}$ is of fixed size $(w_1+n_1)\times\cdots\times(w_m+n_m)$, the size of each $s^{(j)}$ defined in 17 depends on the size of the encoded subsequence $\hat{v}$. While the minimum size of $\hat{v}$ is $(w_1+n_1)\times\cdots\times(w_m+n_m)$, *syndrome-extension* can be achieved by increasing the length of $\hat{v}$. See for example the *Table Based Decoding* approach discussed in [6], [7], [8], [9].

IV. Conclusion

In current literature, typical suggestions for the use of m-D convolutional codes have been to encode m-D information such as images, holograms, animated pictures etc. It has been shown in [3] that arbitrarily large distances can be achieved for $m>2$ codes that are row spaces of 1×1 polynomial matrices. The recursive data processing inherent in m-D, $m>1$ convolution provides the decoder with several different views of the received sequence. This seems to suggest that good error correction can be achieved by reordering 1-D information into m-dimensions and encoding it with a m-D convolutional code. When encoding m-D information, it is desirable to be able to choose the coding rate and generator matrix independent of the ordering of the m-D information (see for example observation 3 in [10]). In this paper we have introduced a method for ordering information and encoded sequences based on the coding rate and using this concept, described a simple method for decoding m-D convolutional codes. Locally Invertible codes have to satisfy certain criteria. In order to have a one-to-one mapping between data and encoded subsequences the code rate has to expressible as a factor of rates along the m-dimensions, with $n_1>k_1,\ldots,n_m>k_m$ and the input constraint lengths $L=l_1\times\ldots\times l_m$ have to be chosen such that the dimensions of the one-to-one mapping subsequence w defined in equation 10 are positive integers. Work on several aspects of Locally Invertible m-D codes including distance properties remains to be studied.

References

[1] E. Fornasini and M. Valcher, “Algebraic aspects of 2D convolutional codes,” *IEEE Trans. Inform. Theory*, vol. IT-40, no. 4, pp. 1068–1082, 1994.

[2] P. A. Weiner, “Multidimensional convolutional codes,” Ph.D. dissertation, University of Notre Dame, 1998. [Online]. Available: http://www.nd.edu/ rosen/preprints.html

[3] P. Weiner, “Basic properties of multidimensional convolutional codes,” in *Codes, Systems and Graphical Models*, ser. IMA Volumes in Mathematics and Its Applications, 123. New York: Springer-Verlag, 2001, ch. 4, pp. 397–414.

[4] H. Gluesing-Luerssen, J. Rosenthal, and P. A. Weiner, “Duality between multidimensional convolutional codes and systems,” in *Advances in Mathematical Systems Theory, A Volume in Honor of Diederich Hinrichsen*, F. Colonius, U. Helmke, F. Wirth, and D. Prätzel-Wolters, Eds. Boston: Birkhauser, 2000, pp. 135–150.

[5] S. Lin and D. J. Costello Jr., *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[6] D. Bitzer and M. Vouk, “A table-driven (feedback) decoder,” in *Proc. Intl. Phoenix Conf. on Comp. Comm.*, Phoenix, AZ, 1991, pp. 385–392.

[7] D. Bitzer, A. Dholakia, H. Koorapaty, and M. Vouk, “On locally invertible rate-1/n convolutional encoders,” *IEEE Transactions on Information Theory*, vol. 44, no. 1, pp. 420–422, January 1998.

[8] A. Dholakia, M. Vouk, and D. Bitzer, “Table based decoding of rate one-half convolutional codes,” *IEEE Trans. on Communications*, vol. 43, no. 2-4, pp. 681–686, 1995.

[9] A. Dholakia, “Locally invertible convolutional encoders, table-based decoding, and their applications to high speed communications,” Ph.D. dissertation, ECE, NCSU, 1993.

[10] C. Charoenlarpnopparut and S. Tantaratana, “Multidimensional convolutional code: progresses and bottlenecks,” in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS 2003)*, Bangkok, Thailand, May 2003, pp. III–686 to III–689.