BCH Convolutional Codes

Joachim Rosenthal, *Senior Member, IEEE*, and Eric V. York, *Member, IEEE*

***Abstract*—Using a new parity-check matrix, a class of convolutional codes with a designed free distance is introduced. This new class of codes has many characteristics of BCH block codes, therefore, we call these codes BCH convolutional codes.**

***Index Terms*—BCH codes, convolutional codes, cyclotomic sets, linear systems.**

I. Introduction

CONVOLUTIONAL codes having a large free distance and a low degree are often found by computer searches. Several authors have extended constructions known for block codes to convolutional codes. A survey of some of this work is provided in the book of Piret [17, Sec. 3.5] where more complete references can be found. Most of these constructions are based on cyclic or quasi-cyclic constructions of block codes. These techniques originate in work by Massey, Costello, and Justesen [13] where it is shown how the free distance of a convolutional code can be lower-bounded by the distance of a related cyclic code. In [7] and [8] Justesen refines the method and he constructs polynomial generator matrices of convolutional codes directly from the generator polynomials of cyclic codes. In these papers Justesen also presents a subfield code construction.

The paper by Tanner [26] uses a quasi-cyclic code to construct a polynomial parity-check matrix of a convolutional code. This work generalizes the methods by Justesen and further progress in this direction has recently been reported by Esmaeili *et al.* [1]. Also worth mentioning is the paper by Piret [18] where he constructed convolutional codes having a parity-check matrix of the form $H(z) = H_0 + zH_1$ with characteristics similar to those of a Reed–Solomon block code. All the referenced constructions have in common that they relate the polynomial representations of the cyclic codes with the polynomial representations of the convolutional codes.

In [23], the authors of this paper jointly with J. M. Schumacher showed that state-space representations commonly used in systems theory are very useful for the construction of convolutional codes. In [23], a construction of a convolutional code with a designed free distance was presented. This construction required that a controllability matrix associated with the state-space system was the parity-check matrix of a Reed–Solomon code. As in the construction of Reed–Solomon codes, large-signal alphabets were required. In [27], York showed how it is possible to do a subfield construction which leads to binary convolutional codes with a designed free distance.

In this paper, we will work systematically with linear state-space descriptions and we will generalize the binary construction presented in [27] to codes over arbitrary Galois fields. The code construction which we present is similar to the classical Bose–Chaudhuri–Hocquenghem (BCH) construction for block codes and this explains our choice of title. There is also some similarity to the work of Justesen [7], [8], and Tanner [26] who derived BCH-type binary constructions starting with the generator polynomial of a BCH block code. The main difference is that the code constructions presented below are much closer to the classical BCH code constructions. There is another advantage of our approach. The nature of the state-space description allows one to analyze the encoder state at each time instance. This knowledge leads to an algebraic decoding algorithm for convolutional codes which is particularly well suited for the BCH convolutional codes constructed in this paper. Details of this algorithm are given in [21].

The paper is structured as follows. Our starting point will be a state-space realization of a *rational* and *systematic encoder*. Using some classical ideas from linear systems theory we will analyze the algebraic structure of convolutional codes in Section II. In this section we will also provide a review of the relevant results from systems theory that will be used throughout the paper. In Section III, we present a general code construction technique which leads to convolutional codes with a designed free distance. As an immediate application of the derived results we obtain the Reed–Solomon-type construction presented in [23]. We show that codes constructed in this way have excellent free distance if the rate is high. In Section IV, we provide the main results of this paper, a detailed convolutional code construction similar to the BCH block code construction.

II. Definitions and Basic Properties

In this section, we will describe convolutional codes with the help of a classical systems theory approach. Let $\mathbb{F} = \mathbb{F}_q$ be the Galois field of q elements and consider the matrices $A \in \mathbb{F}^{\delta\times\delta}, B \in \mathbb{F}^{\delta\times k}, C \in \mathbb{F}^{(n-k)\times\delta}$, and $D \in \mathbb{F}^{(n-k)\times k}$. A rate k/n convolutional code $\mathcal{C}$ of degree δ can be described

Manuscript received November 23, 1997; revised March 7, 1999. This work was supported in part by NSF under Grant DMS-96-10389. The material in this paper was presented in part at the First Lincoln Workshop in Cryptology and Coding, Lincoln, NB, June 1–4, 1997 and at the 1997 IEEE International Symposium on Information Theory, Ulm, Germany, June 29–July 4, 1997.

J. Rosenthal is with EPFL SSC, Lausanne, Switzerland, on leave from the Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556-5683 USA.

E. V. York is with the Department of Defense, National Security Agency, Ft. Meade, MD 20755 USA.

Communicated by F. R. Kschischang, Associate Editor for Coding Theory.
Publisher Item Identifier S 0018-9448(99)06034-4.

by the linear system governed by the equations

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t, \\ y_t &= Cx_t + Du_t, \\ v_t &= \begin{pmatrix} y_t \\ u_t \end{pmatrix}, \qquad x_0 = 0. \end{aligned} \tag{2.1}$$

We call $x_t \in \mathbb{F}^\delta$ the *state vector*, $u_t \in \mathbb{F}^k$ the *information vector*, $y_t \in \mathbb{F}^{n-k}$ the *parity vector*, and $v_t \in \mathbb{F}^n$ the *code vector*, each at time t. In the systems literature, representation (2.1) is known as the *input state output* representation. The integer δ describes the McMillan degree of the linear system (2.1). The McMillan degree is equal to the dimension of the state space $\mathbb{F}^\delta$. In terms of coding theory, (2.1) describes the state-space realization of a rational and systematic convolutional encoder and we will explain this in detail at the end of this section.

Remark 2.1: The state-space realization (2.1) is different from a realization often found in the coding literature. In the coding literature, convolutional codes are usually represented by a *driving variable representation*

$$\begin{aligned} x_{t+1} &= \mathcal{A}x_t + \mathcal{B}m_t \\ v_t &= \mathcal{C}x_t + \mathcal{D}m_t \end{aligned} \tag{2.2}$$

with $m_t \in \mathbb{F}^k$ the *message vector* and $v_t \in \mathbb{F}^n$, $x_t \in \mathbb{F}^\delta$ as above. Representation (2.2) was used by Massey and Sain [14, Theorem 1] and became the standard way in which convolutional codes were presented in terms of linear systems. (Compare with [15].) The difference of (2.1) compared to (2.2) is best seen when the degree $\delta = 0$, which is the case when the convolutional code is memoryless. For this denote by I_{n-k} the $(n-k) \times (n-k)$ identity matrix. Equations (2.1) reduce to the parity-check equation

$$[I_{n-k} - D] \begin{bmatrix} y_t \\ u_t \end{bmatrix} = 0. \tag{2.3}$$

In contrast to this, (2.2) reduces to

$$\begin{bmatrix} y_t \\ u_t \end{bmatrix} = v_t = \mathcal{D}m_t. \tag{2.4}$$

For the purpose of constructing convolutional codes we feel that (2.1) is the better choice.

One of our design objectives will be the construction of convolutional codes with a large free distance. In terms of the state-space description (2.1) we immediately have the characterization of the free distance through

$$d_f(\mathcal{C}) = \min \left(\sum_{t=0}^{\infty} \mathrm{wt}\,(u_t) + \sum_{t=0}^{\infty} \mathrm{wt}\,(y_t) \right) \tag{2.5}$$

where the minimum has to be taken over all possible nonzero codewords and where wt denotes the Hamming weight.

The set of codewords are by definition equal to the set of trajectories $\left\{ \binom{y_t}{u_t} \right\}_{t \geq 0}$ of the dynamical system (2.1). The following Proposition characterizes those trajectories.

Proposition 2.2 (Local Description of Trajectories): Let $\tau, \gamma \in \mathbb{Z}_+$ be positive integers with $\tau < \gamma$. Assume that the encoder is at state x_τ at time $t = \tau$. Then any code sequence $\left\{ \binom{y_t}{u_t} \right\}_{t \geq 0}$ governed by the dynamical system (2.1) must satisfy

$$\begin{pmatrix} y_\tau \\ y_{\tau+1} \\ \vdots \\ \vdots \\ y_\gamma \end{pmatrix} = \begin{pmatrix} C \\ CA \\ \vdots \\ \vdots \\ CA^{\gamma-\tau} \end{pmatrix} x_\tau + \begin{pmatrix} D & 0 & \cdots & & 0 \\ CB & D & \ddots & & \vdots \\ CAB & CB & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & 0 \\ CA^{\gamma-\tau-1}B & CA^{\gamma-\tau-2}B & \cdots & CB & D \end{pmatrix} \cdot \begin{pmatrix} u_\tau \\ u_{\tau+1} \\ \vdots \\ \vdots \\ u_\gamma \end{pmatrix}.$$

Moreover, the evolution of the state vector x_t is given over time as

$$x_t = A^{t-\tau} x_\tau + (A^{t-\tau-1}B \cdots B) \begin{pmatrix} u_\tau \\ \vdots \\ u_{t-1} \end{pmatrix}; \qquad t = \tau+1, \tau+2, \cdots, \gamma+1. \tag{2.6}$$

Proof: This follows easily by iterating the equations that define the system. □

In this paper we will construct codes with large free distance. For algebraic reasons it will be desirable to restrict ourselves to finite-weight codewords:

Definition 2.3: A sequence $\{\binom{y_t}{u_t} \in \mathbb{F}^n | t = 0, 1, 2, \cdots\}$ represents a *finite-weight codeword* if

1) equation (2.1) is satisfied for all $t \in \mathbb{Z}_+$, where $\mathbb{Z}_+$ denotes the set of positive integers;
2) there is an integer γ such that $x_{\gamma+1} = 0$ and $u_t = 0$ for $t \geq \gamma + 1$.

The definition implies that $y_t = 0$ for $t \geq \gamma + 1$ and the code sequence, therefore, has finite weight. For a finite-weight codeword it is, therefore, required that both the input sequence and the state sequence (and hence the output sequence) have finite support. The set of finite-weight codewords can be characterized through a natural parity-check matrix. This matrix will be of central importance in the construction of codes of this paper.

Proposition 2.4 (Global Description of Trajectories): $\{\binom{y_t}{u_t} \in \mathbb{F}^n | t = 0, \cdots, \gamma\}$ represents a finite-weight codeword

if and only if

$$\left(\begin{array}{ccc|ccccc} 0 & \cdots & 0 & A^{\gamma}B & A^{\gamma-1}B & \cdots & AB & B \\ \hline & & & D & & & & \\ & & & CB & D & & & \\ & -I & & CAB & CB & \ddots & & \\ & & & \vdots & & \ddots & \ddots & \\ & & & CA^{\gamma-1}B & CA^{\gamma-2}B & \cdots & CB & D \end{array}\right) \cdot \left(\begin{array}{c} y_0 \\ y_1 \\ \vdots \\ y_{\gamma} \\ \hline u_0 \\ u_1 \\ \vdots \\ u_{\gamma} \end{array}\right) = 0. \quad (2.7)$$

Proof: Setting $\tau = 0$ in Proposition 2.2 gives the bottom portion of the matrix. Since $x_{\gamma+1} = 0$, the top part of the matrix follows from (2.6) in Proposition 2.2. □

Observe that the sequence of information symbols $u_0, \cdots, u_{\gamma}$ in (2.7) is restricted by some algebraic constraints. These constraints simply guarantee that the state vector $x_{\gamma+1} = 0$.

In what follows, we will use the local and global systems theoretic properties described above to give code constructions with designed free distance. These representations were also crucial in the decoding algorithm [21].

The set of finite-weight codewords has a natural module structure over the polynomial ring $\mathbb{F}[z]$. For this consider a finite-weight codeword $\{\binom{y_t}{u_t}\}_{0 \le t \le \gamma}$ with corresponding state sequence $\{x_t\}_{0 \le t \le \gamma}$. Define

$$x(z) = x_0 z^{\gamma} + x_1 z^{\gamma-1} + \cdots + x_{\gamma}, \qquad x_t \in \mathbb{F}^{\delta},\ t = 0, \cdots, \gamma$$
$$u(z) = u_0 z^{\gamma} + u_1 z^{\gamma-1} + \cdots + u_{\gamma}, \qquad u_t \in \mathbb{F}^{k},\ t = 0, \cdots, \gamma$$

and let

$$y(z) = y_0 z^{\gamma} + y_1 z^{\gamma-1} + \cdots + y_{\gamma}, \qquad y_t \in \mathbb{F}^{n-k},\ t = 0, \cdots, \gamma.$$

One immediately verifies that $\{x_t, u_t, y_t\}$ satisfy (2.1) if and only if

$$\begin{bmatrix} zI - A & 0_{\delta \times (n-k)} & -B \\ -C & I_{n-k} & -D \end{bmatrix} \begin{bmatrix} x(z) \\ y(z) \\ u(z) \end{bmatrix} = 0. \quad (2.8)$$

Moreover, the set of polynomial vectors

$$\begin{bmatrix} y(z) \\ u(z) \end{bmatrix} \in \mathbb{F}^n[z]$$

which satisfy (2.8) for some polynomial vector $x(z) \in \mathbb{F}^{\delta}[z]$ forms a $\mathbb{F}[z]$-submodule of the free module $\mathbb{F}^n[z]$. By abuse of notation we will denote this module by $\mathcal{C}(A, B, C, D)$ and we will call this module the *finite-weight convolutional code* generated by the matrices A, B, C, D. The code $\mathcal{C}(A, B, C, D)$ will be the main focus of our investigation. At the end of this section we will relate the properties of this code with the standard literature on convolutional codes [2], [5], [6], [15], and [17].

Since $\mathbb{F}[z]$ is a principal ideal domain, $\mathcal{C}(A, B, C, D)$ is a free module of rank k (see [4. Ch. IV, Theorem 6.1]) and there exists an $n \times k$ polynomial matrix $G(z)$ such that

$$\mathcal{C}(A, B, C, D) = \{v(z) \in \mathbb{F}^n[z] | \exists m(z) \in \mathbb{F}^k[z] \colon v(z) = G(z)m(z)\}.$$

We will call $G(z)$ a *polynomial encoder* of the finite-weight convolutional code $\mathcal{C}(A, B, C, D)$. The following lemma provides a way to compute a polynomial encoder $G(z)$.

Lemma 2.5: There exist polynomial matrices $X(z), Y(z)$, and $U(z)$ of size $\delta \times k$, $(n-k) \times k$, and $k \times k$, respectively, such that

$$\ker \begin{bmatrix} zI - A & 0 & -B \\ -C & I & -D \end{bmatrix} = \operatorname{im} \begin{bmatrix} X(z) \\ Y(z) \\ U(z) \end{bmatrix}. \quad (2.9)$$

Moreover, the polynomial matrix

$$G(z) = \begin{bmatrix} Y(z) \\ U(z) \end{bmatrix}$$

describes a polynomial encoder.

Proof: The matrix on the left-hand side of (2.9) has size $(\delta + n - k) \times (\delta + n)$ and it has full rank over the field of rationals $\mathbb{F}(z)$. The kernel over the field $\mathbb{F}(z)$ has, therefore, dimension k. This kernel has a minimal polynomial basis in the sense of Forney [3]. Choosing such a minimal basis results in the matrix on the right-hand side of (2.9).

If $v(z) \in \mathbb{F}^n[z]$ is a finite-weight codeword then there exists a polynomial vector $x(z) \in \mathbb{F}^{\delta}[z]$ such that

$$\begin{bmatrix} x(z) \\ v(z) \end{bmatrix} \in \operatorname{im} \begin{bmatrix} X(z) \\ G(z) \end{bmatrix}.$$

In other words, $G(z)$ is a polynomial encoder for the convolutional code $\mathcal{C}(A, B, C, D)$. □

Clearly, not every 4-tuple of matrices (A, B, C, D) having sizes $\delta \times \delta$, $\delta \times k$, $(n-k) \times \delta$, and $(n-k) \times k$, respectively, results in a "desirable" finite-weight convolutional code. In addition, the description (2.1) is in general not unique. The following lemma addresses the nonuniqueness of the description (2.1). We omit the simple proof.

Lemma 2.6: If $S \in Gl_{\delta}(\mathbb{F})$ is an invertible matrix then

$$\mathcal{C}(SAS^{-1}, SB, CS^{-1}, D) = \mathcal{C}(A, B, C, D).$$

The transformation $S \in Gl_{\delta}(\mathbb{F})$ has no affect on the degree δ. Sometimes it is possible to describe the code $\mathcal{C}(A, B, C, D)$ using matrices A_1, B_1, C_1, D_1 which are in size smaller than the matrices A, B, C, D. If the matrices A, B, C, D have the smallest possible size we say (2.1) is a *minimal* description for the code $\mathcal{C}(A, B, C, D)$. In order to describe the class of matrices (A, B, C, D) which describe noncatastrophic convolutional encoders in a minimal way we will have to recall some systems-theoretic concepts. We will start with some notation which will be convenient throughout the paper.

Let A, B, C be scalar matrices over $\mathbb{F}$ of size $\delta \times \delta$, $\delta \times k$, and $(n-k) \times \delta$, respectively. Let j be a positive integer and define

$$\Phi_j(A,B) := (B \quad AB \cdots A^{j-2}B \quad A^{j-1}B) \tag{2.10}$$

$$\Omega_j(A,C) := \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{j-1} \end{pmatrix}. \tag{2.11}$$

The matrices $\Phi_j(A,B)$ and $\Omega_j(A,C)$ will be of central importance in the rest of this paper. With this notation we define (compare, e.g., with [9, p. 356])

Definition 2.7: Let A, B be matrices of size $\delta \times \delta$ and $\delta \times k$, respectively. Then (A,B) is called a *controllable pair* if

$$\operatorname{rank} \Phi_\delta(A,B) = \delta. \tag{2.12}$$

If (A,B) is a controllable pair then we call the smallest integer κ having the property that $\operatorname{rank} \Phi_\kappa(A,B) = \delta$ the *controllability index* of (A,B).

In a similar fashion we define:

Definition 2.8: Let A, C be matrices of size $\delta \times \delta$ and $(n-k) \times \delta$, respectively. Then (A,C) is called an *observable pair* if

$$\operatorname{rank} \Omega_\delta(A,C) = \delta. \tag{2.13}$$

If (A,C) is an observable pair then we call the smallest integer ν having the property that $\operatorname{rank} \Omega_\nu(A,C) = \delta$ the *observability index* of (A,C).

Let us now explain what happens if either (A,B) is not a controllable pair or (A,C) is not an observable pair.

If (A,B) is not a controllable pair then there is an integer r with $\operatorname{rank} \Phi_\delta(A,B) = r < \delta$. One shows the existence of an invertible matrix $S \in Gl_\delta(\mathcal{F})$ such that

$$(SAS^{-1}, SB, CS^{-1}) = \left(\begin{bmatrix} A_1 & A_2 \\ 0 & A_3 \end{bmatrix}, \begin{bmatrix} B_1 \\ 0 \end{bmatrix}, [C_1, C_2] \right) \tag{2.14}$$

where A_1, B_1, C_1 are matrices of size $r \times r$, $r \times k$, and $(n-k) \times r$, respectively, and where (A_1, B_1) forms a controllable pair. The partitioning appearing in (2.14) is often referred to as Kalman's normal form and the existence of such partitioning is easily established.

Theorem 2.9: Assume (A,B) is not a controllable pair and let (SAS^{-1}, SB, CS^{-1}) be the Kalman normal form as in (2.14). Then

$$\mathcal{C}(A,B,C,D) = \mathcal{C}(A_1, B_1, C_1, D).$$

Proof: By Lemma 2.6 we know that

$$\mathcal{C}(SAS^{-1}, SB, CS^{-1}, D) = \mathcal{C}(A,B,C,D).$$

The theorem now follows from the identity

$$\ker \begin{bmatrix} zI_r - A_1 & -A_2 & 0 & -B_1 \\ 0 & zI_{\delta-r} - A_3 & 0 & 0 \\ -C_1 & -C_2 & I_{n-k} & -D \end{bmatrix} = \ker \begin{bmatrix} zI_r - A_1 & 0 & 0 & -B_1 \\ 0 & I_{\delta-r} & 0 & 0 \\ -C_1 & 0 & I_{n-k} & -D \end{bmatrix}. \tag{2.15}$$

□

The theorem simply states that if (A,B) is not a controllable pair then the finite-weight convolutional code $\mathcal{C}(A,B,C,D)$ is not described in a minimal way. Because of this we will now assume that (A,B) forms a controllable pair. The following theorem is due to Popov [19, Theorem 2].

Theorem 2.10: If (A,B) forms a controllable pair then there exist positive integers $\kappa_1 \geq \cdots \geq \kappa_k$ only dependent on the Gl_δ equivalence class of (A,B) having the following properties.

1) $\kappa_1 = \kappa$, the controllability index of (A,B).
2) $\sum_{i=1}^{k} \kappa_i = \delta$, the size of the matrix A.
3) There exist polynomial matrices $X(z), Y(z), U(z)$ satisfying (2.9) and having the property that the ith column degree of

$$G(z) = \begin{bmatrix} Y(z) \\ U(z) \end{bmatrix}$$

is equal to κ_i, and the ith column degree of $X(z)$ is equal to $\kappa_i - 1$ for $i = 1, \cdots, k$.

The indices $\kappa_1 \geq \cdots \geq \kappa_k$ are often referred to as the controllability indices of the pair (A,B). (See [9] for more details.) In the coding literature [6, Sec. 2.5] the integers $\kappa_1, \cdots, \kappa_k$ are referred to as the *constraint indices* and κ is called the *memory* of the encoder $G(z)$. We would like to note that those indices are invariants of the column module $\mathcal{C}(A,B,C,D)$ of $G(z)$ and that they are, in general, different from the minimal polynomial indices (in the sense of Forney [3]) of the rational vector space spanned by the columns of $G(z)$. Details about those differences are given in [16].

Next we are interested in conditions on the matrices A, B, C, D which guarantee that the induced polynomial encoder $G(z)$ is noncatastrophic. First assume that $G_1(z)$ and $G_2(z)$ are two polynomial encoders of the finite-weight convolutional code $\mathcal{C}(A,B,C,D)$. Since the columns of $G_1(z)$ and the columns of $G_2(z)$ both form a $\mathbb{F}[z]$-basis of the free module $\mathcal{C}(A,B,C,D)$ there exists a $k \times k$ unimodular matrix $V(z)$ such that $G_2(z) = G_1(z)V(z)$. It follows that $G_1(z)$ describes a noncatastrophic encoder (i.e., $G_1(z)$ is right prime) if and only if $G_2(z)$ describes a noncatastrophic encoder. For finite-weight convolutional codes one therefore has a notion of noncatastrophicity. In order to avoid any confusion with the existing literature and in light of the next lemma, we call $\mathcal{C}(A,B,C,D)$ an *observable convolutional code* (compare with [27], [23]) if one and hence every encoder $G(z)$ of $\mathcal{C}(A,B,C,D)$ describes a noncatastrophic encoder.

The following result identifies the observable convolutional codes.

Lemma 2.11: Assume the matrices (A, B) form a controllable pair. The convolutional code $\mathcal{C}(A, B, C, D)$ defined through (2.1) represents an observable convolutional code if and only if (A, C) forms a observable pair.

Proof: Notation as in Lemma 2.5. By [23, Lemma 3.2] $G(z)$ is right prime if and only if the matrix pencil (i.e., polynomial matrix of degree one)

$$\begin{bmatrix} zI - A \\ -C \end{bmatrix}$$

is right prime and by the well-known Popov–Belevitch–Hautus test [9, Theorem 6.2-6] this is the case if and only if (A, C) forms an observable pair. □

An analogous result for the driving variable representation does not exist.

Example 2.12 ([27]): Let $\mathcal{C}$ be the rate $\frac{1}{2}$ convolutional code over $\mathbb{F}_2$ having a catastrophic generator matrix

$$G(z) = \begin{pmatrix} z^2 + 1 \\ z + 1 \end{pmatrix}.$$

The driving variable representation for this system is given by

$$x_{t+1} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} x_t + \begin{pmatrix} 1 \\ 0 \end{pmatrix} m_t$$

$$v_t = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} x_t + \begin{pmatrix} 1 \\ 1 \end{pmatrix} m_t.$$

Despite the fact that $G(z)$ is catastrophic the matrix pair

$$\left(\begin{pmatrix} 0\,0 \\ 1\,0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right)$$

forms a controllable pair and the pair

$$\left(\begin{pmatrix} 0\,0 \\ 1\,0 \end{pmatrix}, \begin{pmatrix} 0\,1 \\ 1\,0 \end{pmatrix} \right)$$

forms an observable pair.

More generally, one can show that any catastrophic polynomial encoder $G(z)$ has a driving variable representation $(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D})$ as described in (2.2) whose matrix pairs $(\mathcal{A}, \mathcal{B})$ and $(\mathcal{A}, \mathcal{C})$ are both controllable and observable, respectively, thus making it difficult to work with this representation of a code. There is yet another peculiarity of the driving variable representation. If $G(z)$ is a polynomial encoder, then the matrix $\mathcal{A}$ appearing in the driving variable representation (2.2) is necessarily nilpotent. For these reasons we feel it is preferable to work with the input state output representation (2.1) of a code.

Up to now we have concentrated our efforts on properties of finite-weight convolutional codes of the form $\mathcal{C}(A, B, C, D)$. In the coding literature [2], [5], [6], [15], [17] it is customary to define a convolutional code as a $\mathcal{F}$-linear subspace of $\mathcal{F}^n$, where $\mathcal{F}$ represents either the field of rational functions $\mathbb{F}(z)$ or the field of formal Laurent series $\mathbb{F}((z))$. If $G(z)$ is a polynomial encoder of $\mathcal{C}(A, B, C, D)$ then $G(z)$ induces a convolutional code $\hat{\mathcal{C}} = \hat{\mathcal{C}}(A, B, C, D) \subset \mathcal{F}^n$ by simply defining $\hat{\mathcal{C}}$ as the $\mathcal{F}$-linear span of the columns of $G(z)$. Note that this definition is independent of the particular convolutional encoder $G(z)$ of $\mathcal{C}(A, B, C, D)$.

The free distance of the convolutional code $\hat{\mathcal{C}}$ is defined [6], [17] as the minimal value d_f in (2.5), where the minimization is taken over all possible nonzero codewords in $\hat{\mathcal{C}}$. The next lemma shows that for constructions of convolutional codes with a certain designed distance it is sufficient to consider the finite-weight convolutional code $\mathcal{C}(A, B, C, D)$.

Lemma 2.13: Let $\hat{\mathcal{C}}(A, B, C, D) \subset \mathcal{F}^n$ be a convolutional code defined by the matrices A, B, C, D. Assume (A, C) forms an observable pair. If one minimizes (2.5) over all nonzero codewords inside $\hat{\mathcal{C}}$ then the minimum value is attained at a codeword inside the finite-weight convolutional code $\mathcal{C}(A, B, C, D)$.

Proof: Let $\{\binom{y_t}{u_t} \in \mathbb{F}^n | t = 0, 1, 2, \cdots\}$ be a nonzero code sequence which results in the minimal value d_f in (2.5). By definition, this sequence has finite weight d_f. Assume, by contradiction, that this sequence does not belong to $\mathcal{C}(A, B, C, D)$, i.e., the state sequence $\{x_t \in \mathbb{F}^\delta | t = 0, 1, 2, \cdots\}$ does not have finite support. Under this condition, there exists a positive integer τ such that $x_\tau \neq 0$ and $u_t = 0$ and $y_t = 0$ for $t \geq \tau$. Since (A, C) forms an observable pair this contradicts the local description of the trajectories as given in Proposition 2.2. □

If (A, C) does not form an observable pair then the minimization over the nonzero codewords in $\hat{\mathcal{C}}(A, B, C, D)$ is, in general, smaller than the minimization over the nonzero codewords in $\mathcal{C}(A, B, C, D)$. For this, consider the parity-check matrix appearing in (2.7). For each positive integer γ, let d_γ be the distance of the block code defined by (2.7), then d_γ is equal to the minimal weight of a nonzero trajectory of length $\gamma+1$ which starts from and returns to the zero state. The integers d_γ form a nonincreasing sequence and they are related but not equal to the γth-order row distance of an encoder [6, Sec. 3.1]. The limit $d_\infty := \lim_{\gamma \mapsto \infty} d_\gamma$ is equal to the minimal weight of a nonzero trajectory which starts from and returns to the all-zero state. This integer is also equal to the minimal value d_f in (2.5) where the minimization is taken over all possible nonzero codewords $\mathcal{C}(A, B, C, D)$. The book of Lin and Costello [11, Sec. 10.3] defines d_∞ as the free distance of a convolutional encoder.

On the side of d_∞ there is a second important distance measure called the *jth-order column distance* [6], [11], [17] of $\mathcal{C}$, defined as

$$c_j = \min \left(\sum_{t=0}^{j} \mathrm{wt}\,(u_t) + \sum_{t=0}^{j} \mathrm{wt}\,(y_t) \right) \tag{2.16}$$

where the minimum has to be taken over all possible (truncated) nonzero codewords of $\hat{\mathcal{C}}(A, B, C, D)$. For any positive integers j and γ one has that $c_j \leq d_\gamma$. $c_\infty := \lim_{j \mapsto \infty} c_j$ is equal to the minimal weight of a nonzero trajectory which starts from the all-zero state, but does not necessarily return to the all-zero state. This integer is also equal to the minimal value d_f in (2.5) where the minimization is taken over all possible nonzero codewords $\hat{\mathcal{C}}(A, B, C, D)$. The books of

Johannesson and Zigangirov [6, Ch. 3] and Piret [17, p. 67] define c_∞ as the free distance of the convolutional $\hat{\mathcal{C}}$. By Lemma 2.13 (compare also with [6, Theorem 3.6]) one has the equality $c_\infty = d_\infty$ when the code is observable.

Our last result of this section will show that the state-space description (2.1) describes in a natural way the dynamics of a rational and systematic encoder. For this, recall that a rational function $(p(z)/q(z)) \in \mathbb{F}(z)$ is called proper if $\deg q \geq \deg p$. A matrix with entries in $\mathbb{F}(z)$ is called a *proper transfer function* if each entry of $R(z)$ is a proper rational function.

Lemma 2.14: Notation as in Lemma 2.5. The matrices A, B, C, D appearing in (2.9) form a state-space realization of the transfer function $Y(z)U(z)^{-1}$, i.e., one has the relation:

$$C(zI - A)^{-1}B + D = Y(z)U(z)^{-1}. \tag{2.17}$$

In particular, $Y(z)U(z)^{-1}$ describes a proper transfer function.

Proof: Equation (2.9) is equivalent to the equations

$$X(z)U(z)^{-1} = (zI - A)^{-1}B$$

and

$$YU(z)^{-1} = CX(z)U(z)^{-1} + D$$

which, in turn, is equivalent to (2.17). Using Cramer's rule it follows from (2.17) that $Y(z)U(z)^{-1}$ is proper. □

Remark 2.15: If the polynomial matrices $U(z), Y(z)$ have the property that $Y(z)U(z)^{-1}$ describes a proper transfer function then there always exist matrices A, B, C, D satisfying (2.17). The dynamical system (2.1) is then called a *state-space realization* of the transfer function $Y(z)U(z)^{-1}$. If $U(z)$ is either not invertible or if $Y(z)U(z)^{-1}$ is not proper then a more general state-space description such as the "K, L, M" description [23, Theorem 3.1] will be needed. A particular simple algorithm for computing both traditional A, B, C, D realizations as well as more general realizations was recently given in [22].

We can view the transfer function $Y(z)U(z)^{-1}$ in two ways. In the coding literature it is customary to consider the code $\hat{\mathcal{C}}(A, B, C, D) \subset \mathcal{F}^n$, the $\mathcal{F}$-linear subspace spanned by the columns of $G(z)$. As an encoder over $\mathcal{F}$, $G(z)$ is equivalent to the systematic encoder

$$\begin{bmatrix} Y(z)U(z)^{-1} \\ I_k \end{bmatrix}$$

and one can view the encoding as a linear map

$$\varphi\colon \mathcal{F}^k \to \mathcal{F}^{n-k}, u(z) \mapsto y(z) = Y(z)U(z)^{-1}u(z).$$

Under this point of view there are no restrictions on $u(z) \in \mathcal{F}^k$.

Alternatively, $Y(z)U(z)^{-1}$ describes a module homomorphism from the column module of $U(z)$ to the column module of $Y(z)$. From this point of view, the information vector $u(z)$ is assumed to be in the column module of $U(z)$ and this restriction will guarantee that the sequence of state vectors $x_0, x_1, x_2, \cdots$ reaches the all-zero state in finite time. (Compare with Proposition 2.4.)

The major reason we have developed a theory for finite-weight convolutional codes (i.e., modules) of the form $\mathcal{C}(A, B, C, D)$ is Proposition 2.4 together with the convenient parity-check matrix appearing in (2.7). This matrix gives a nice algebraic criterion for characterizing the distance d_f as defined in (2.5) and it is also very useful if one is interested in algebraic methods for decoding convolutional codes [21]. At the same time, there seems to exist little engineering reason why infinite-weight codewords have to be part of the theory. In fact, McEliece [15, Sec. 2] points out that finite-weight codewords are the only ones that can occur in engineering practice. From a more mathematical point of view there are some other beneficial points. The set of all submodules of $\mathbb{F}^n[z]$ is in one-to-one correspondence with the the set of all linear, shift-invariant, and complete behavior of $\mathbb{F}^n[[z]]$ by a categorical duality. (See [23, Theorem 2.6].) This allows one to simply carry over the representations from systems theory to convolutional coding theory and we have done this in this section. Finally, we would like to mention that the set of all rank k submodules $\mathcal{C} \subset \mathbb{F}^n[z]$ of degree at most δ has in a natural way the structure of a smooth projective variety $X^\delta_{k,n}$ [20]. The set of k-dimensional subspaces $\hat{\mathcal{C}} \subset \mathcal{F}^n$ of degree at most δ corresponds to the observable finite-weight convolutional codes and inside the variety $X^\delta_{k,n}$ this subset forms a proper Zariski open set. The "missing points" inside the closure of $X^\delta_{k,n}$ are the nonobservable convolutional codes.

In the next section, we will use the algebraic description of Proposition 2.4 to construct observable convolutional codes of the form $\mathcal{C}(A, B, C, D)$ having a fixed rate and degree. Because of this proposition we will work with finite-weight codewords and the free distance that we compute corresponds to the smallest possible weight of a codeword whose state starts and terminates in the all-zero state. Because of Lemmas 2.11 and 2.13, the distance bounds for these codes also hold if one prefers to consider infinite sequences whose state does not terminate in the all-zero state.

III. A General Code Construction Technique and Reed–Solomon-Type Convolutional Codes

How do we go about choosing A, B, C, and D matrices to obtain observable convolutional codes with large free distance? We showed in Lemma 2.11 that a code is observable as soon as the matrix pair (A, C) forms an observable pair. The code description is only in a minimal form if (A, B) forms a controllable pair. Hence, two obvious conditions for the matrices chosen are that

$$\operatorname{rank} \Phi_\delta(A, B) = \delta \quad \text{and} \quad \operatorname{rank} \Omega_\delta(A, C) = \delta.$$

What are some other conditions? Propositions 2.2 and 2.4 tell us that the $y(z)$ part of the trajectories depends locally on $\Omega(A, C)$, while the $u(z)$ part depends globally on $\Phi(A, B)$. Using this insight, we show that by choosing A, B, and C properly, we can control the trajectories enough to give a lower bound on the free distance of the corresponding code. Note that we have complete freedom in choosing the matrix D and there is no concern about the nilpotency of the matrix A. The following theorem is in essence [27, Theorem 6.2.1].

Theorem 3.1: Let $\mathcal{C}$ be an observable, rate k/n, degree δ, convolutional code defined through the matrices A, B, C, and D. Let ν be the observability index of the pair (A, C) and

suppose that there exists $d \in \mathbb{Z}_+$ such that $\Phi_{d\nu}(A,B)$ forms the parity-check matrix of a block code of distance d. Then the free distance of $\mathcal{C}$ is greater than or equal to d.

Proof: Let $v(z) \in \mathcal{C}$ and suppose that $\deg\, v(z) = \gamma$. Without loss of generality we assume that $u_0 \neq 0$. If $\gamma < d\nu$ then, by Proposition 2.4 and our assumption on $\Phi_{d\nu}(A,B)$, we obtain $\mathrm{wt}\,(u(z)) \geq d$, which implies that $\mathrm{wt}\,(v(z)) \geq d$. Suppose now that $\gamma \geq d\nu$ and that $\mathrm{wt}\,(u(z)) = b < d$ (note that if $\mathrm{wt}\,((u(z)) = b \geq d$ we would be done). By the pigeonhole principle, there must be at least $d-b$ length ν sequences of all-zero input vectors occurring before time $d\nu - 1$. Let

$$u_{i+1} = u_{i+2} = \cdots = u_{i+\nu} = 0$$

be one such sequence.

We claim that $x_{i+1} \neq 0$. To see this, note that if $x_{i+1} = 0$, then we could choose $u_t = 0$ for all $t > i$ and we would obtain a trajectory $\hat{v}(z) \in \mathcal{C}$ with $\deg\, \hat{v}(z) \leq i < d\nu$. Proposition 2.4 implies $\Phi_i(A,B)\hat{u} = 0$ with

$$\mathrm{wt}\,(\hat{u}_0, \hat{u}_1, \cdots \hat{u}_i) < d$$

which contradicts our choice of A, B in the statement of the theorem. Hence, $x_{i+1} \neq 0$. Using Proposition 2.2, and the fact that $\mathrm{rank}\ \Omega_\nu(A,C) = \delta$, we see that $\mathrm{wt}\,(y_{i+1}, y_{i+2}, \cdots, y_{i+\nu}) > 0$. Since there are at least $d-b$ such sequences, we obtain

$$\mathrm{wt}\,((v_0, v_1, v_2, \cdots, v_{(d\nu-1)})) \geq d - b + b = d$$

which implies that $\mathrm{wt}\,(v(z)) \geq d$. □

According to this theorem one way to construct convolutional codes having rate k/n, degree δ, and designed distance d is to ensure that the matrix $\Phi_{d\nu}(A,B)$ defines a parity-check matrix for a "good block code." This was accomplished in [23] when the finite field $\mathbb{F}$ had sufficiently many elements.

Corollary 3.2: Let $r := \max\{n-k, k\}$ and let α be a primitive of the field $\mathbb{F}_q$, i.e., a generator of the cyclic group $\mathbb{F}_q^*$. Assume $|\mathbb{F}_q| = q > \delta r\lceil \delta/(n-k)\rceil$ and let $\epsilon = \max\{n - 2k + 1, 0\}$. Let

$$A := \begin{pmatrix} \alpha^r & 0 & \cdots & 0 \\ 0 & \alpha^{2r} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \alpha^{\delta r} \end{pmatrix}$$

$$B := \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{k-1} \\ 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(k-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha^\delta & \alpha^{2\delta} & \cdots & \alpha^{\delta(k-1)} \end{pmatrix}$$

$$C := \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha & \alpha^2 & \cdots & \alpha^\delta \\ \alpha^2 & \alpha^4 & \cdots & \alpha^{2\delta} \\ \vdots & \vdots & & \vdots \\ \alpha^{n-k-1} & \alpha^{2(n-k-1)} & \cdots & \alpha^{\delta(n-k-1)} \end{pmatrix}$$

$$D := \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha & \alpha^2 & \cdots & \alpha^k \\ \vdots & \vdots & & \vdots \\ \alpha^{(n-k-1)} & \alpha^{2(n-k-1)} & \cdots & \alpha^{k(n-k-1)} \end{pmatrix}.$$

Then the convolutional code $\mathcal{C}$ defined by the matrices A, B, C, D is an observable, rate k/n convolutional code with degree δ and free distance

$$d_f(\mathcal{C}) \geq \delta + 1 + \epsilon. \tag{3.1}$$

Proof: The observability index in this situation is $\nu = \lceil \delta/(n-k)\rceil$. The size of the finite field guarantees that $\Phi_{\delta\nu}(A,B)$ forms the parity-check matrix of a maximum-distance separable (MDS) code, in particular the distance of this block code is $d = \delta + 1$. Because of Theorem 3.1, the free distance of the convolutional code $\mathcal{C}(A,B,C,D)$ is, therefore, at least $\delta + 1$. This establishes the claim in the case when the convolutional code has a "high rate."

When $\epsilon = \max\{n - 2k + 1, 0\} > 0$ (i.e., the code has low rate) one can improve on the free distance estimate. One observes that the $(n-k) \times n$ matrix $(I_{n-k}\ D)$ defines the parity-check matrix of an MDS block code. The weight of the code component (u_0, y_0), therefore, has to be at least $n-k+1$. The weight of y_0 is therefore at least $n-2k+1$. This completes the proof. □

It is in general straightforward to compute a generator matrix from the 4-tuple of matrices (A,B,C,D) as given in Corollary 3.2 and this is the case even if the free distance is fairly large. The following example was given in [27, Example 6.3.4].

Example 3.3: Let $n = 3, k = 2, q = 1801, \alpha = 11, \delta = 30$, and $D = (0\ 0)$. Then $r = 2$. Let (A,B,C,D) be defined as in Corollary 3.2. Using the identity (2.9) one readily computes the 3×2 generator matrix $G(z)$ having entries

$$\begin{aligned}
g_{1,1}(z) &= 315z + 749 + 75z^{10} + 897z^2 + 639z^3 - 610z^4 \\
&\quad + 872z^5 - 133z^6 + 40z^7 - 431z^8 + 565z^9 \\
&\quad + 247z^{11} + 408z^{12} + 674z^{13} - 11z^{14} - 783z^{15} \\
g_{1,2}(z) &= 935z + 104z^{10} + 838z^2 + 410z^3 + -340z^4 \\
&\quad - 376z^5 - 141z^6 + 995z^7 + 322z^8 - 258z^9 \\
&\quad - 529z^{11} - 193z^{12} - 507z^{13} - 746z^{14} \\
&\quad - 552z^{15} + 559 \\
g_{2,1}(z) &= 825z + 418z^{10} + 82z^2 + 830z^3 + 47z^4 + 850z^5 \\
&\quad + 449z^6 - 741z^7 + 601z^8 + 306z^9 + 452z^{11} \\
&\quad + 524z^{12} + 310z^{13} + 235z^{14} - 708z^{15} \\
g_{2,2}(z) &= 1 - 442z + 672z^{10} + 756z^2 - 586z^3 + 909z^4 \\
&\quad + 224z^5 - 457z^6 + 661z^7 - 532z^8 - 300z^9 \\
&\quad + 385z^{11} - 98z^{12} - 627z^{13} + 281z^{14} \\
g_{3,1}(z) &= 858z - 424z^{10} - 185z^2 - 91z^3 + 928z^4 + 988z^5 \\
&\quad - 570z^6 - 593z^7 + 640z^8 - 631z^9 + 750z^{11} \\
&\quad + 175z^{12} + 647z^{13} + 895z^{14} - 22z^{15} \\
g_{3,2}(z) &= -22z + 907z^{10} + 812z^2 + 550z^3 - 615z^4 \\
&\quad + 324z^5 - 105z^6 + 435z^7 + 559z^8 - 679z^9 \\
&\quad - 336z^{11} - 472z^{12} + 544z^{13} - 273z^{14} + 233z^{15}.
\end{aligned}$$

$G(z)$ defines a rate $2/3$ convolutional code whose free distance is at least 31. The memory of this code is 15 and the degree is 30.

In general, the computation of the generator matrix requires the solution of a system of linear equations having fewer than $\delta^2+n(\delta+k)$ unknowns. For this, observe that the computation of the kernel in (2.9) is a linear problem in the coefficients of the polynomial matrices $X(z), Y(z), U(z)$. By Popov's Theorem 2.10 we know that the ith-column degree of $X(z)$ is $\kappa_i - 1$. The number of coefficients in $X(z)$ is, therefore, $\delta \ \sum_{i=1}^{k} \kappa_i = \delta^2$. Similarly, the number of coefficients in $Y(z), U(z)$ is $n \ \sum_{i=1}^{k} (\kappa_i + 1) = n(\delta + k)$.

If one writes down the linear system that the coefficients of $X(z), Y(z), U(z)$ do satisfy, one observes that this system is in a fairly sparse form. Because of this, one can compute generator matrices with a designed free distance of over 1000. Of course, codes with such a large free distance will require very large finite fields and we will explain in the next section how to overcome this obstacle.

In the remainder of this section we analyze how the free distance of the presented codes does compare with the best possible free distance among all codes with the same rate and the same degree.

Let us first discuss Example 3.3: A rate $2/3$ code of memory 15 and degree 30 can have a distance of at most $3(15+1) = 48$. Lin and Costello [11, Table 11.1] give the best rate $2/3$ binary codes with degree $\delta \leq 10$. For example, the best $2/3$ binary code of degree $\delta = 10$ has a distance of $d_f = 10$. Since these results were obtained by computer search, no comparable results for higher degree and larger field sizes are available in [11].

In general, we know from (3.1) that for the code described in Corollary 3.2 the following estimate holds:

$$\frac{d_f(\mathcal{C})}{\delta} \geq 1.$$

It follows that the presented codes are "asymptotically good" in the sense that

$$\lim_{\delta \to \infty} \frac{d_f(\mathcal{C})}{\delta} > 0. \tag{3.2}$$

How are they compared to the "best possible codes?" For this we first derive a simple bound for a convolutional code having a certain rate and a certain degree:

Lemma 3.4: Suppose that $\mathcal{C}$ is a rate k/n code with degree δ. Then

$$d_f(\mathcal{C}) \leq n\left\lfloor \frac{\delta}{k} + 1 \right\rfloor =: d_{\max}. \tag{3.3}$$

Proof: The smallest column degree of a generator matrix $G(z)$ is given by $\kappa_k \leq \lfloor \delta/k \rfloor$. The weight of the corresponding column vector is, therefore, at most $d_{\max}$. □

Inequality (3.3) in particular implies that $\delta \geq (k/n)d_{\max} - k$. Using these estimates we obtain for the codes constructed above

$$\lim_{\delta \to \infty} \frac{d_f(\mathcal{C})}{d_{\max}} \geq \frac{k}{n}.$$

Hence, for very high rates, the codes constructed above are near-maximal. However, we note that very large fields are needed in order to construct these codes. For low rates some constructions were provided by the first author and Smarandache [24] which result in better free distances than $\frac{k}{n}d_{\max}$. For the rate $\frac{1}{n}$ Justesen [8] constructed codes with maximal possible free distance $d_f = n(\delta+1)$. All of these constructions require large field sizes.

It is interesting to observe that the construction that we provided in Theorem 3.1 is near-optimal for high rates whereas the construction of Justesen [7], [8] and the extensions of Tanner [26] are best for low rates. In [25], Smarandache together with the first author showed how the result of Justesen [8] can be obtained by choosing the matrices A, B as in Corollary 3.2 and adding in a clever way a matrix C different from the one provided in Corollary 3.2. Unfortunately, this method works at this point only for rate $1/n$ and the construction of a C matrix resulting in better distances seems to be difficult in general.

We conclude the section with an example which explains the properties of the provided codes.

Example 3.5: Let $G(z)$ be the generator matrix of a rate $9/10$ convolutional code of degree $\delta = 80$. The smallest controllability index (compare with Theorem 2.10) of $G(z)$ is then at most 8 and the free distance is hence at most $d_{\max} = 10(8+1) = 90$. Corollary 3.2 shows that there exists a rate $9/10$ code of distance 81.

IV. BCH-Type Convolutional Codes

In this section, we will give techniques for constructions over arbitrary finite fields $\mathbb{F}_q$.

The generalization of the Reed–Solomon codes in the theory of block codes are the Bose–Chaudhuri–Hocquenghem (BCH) codes (see [12]). In the sequel, we explain how the construction of the last section can be generalized to arrive at a BCH type of convolutional code over arbitrary finite fields $\mathbb{F}_q$. The case where $q = 2$ was first presented in [27]. First, we review some of the ingredients of the BCH construction for block codes.

Definition 4.1: Let $\mathbb{F}_q$ be an arbitrary finite field, let b, d, N be positive integers, let N satisfy $(N, q) = 1$, and let α be a primitive Nth root of unity. Let $\mathbb{F}_{q^m}$ be the splitting field of $x^N - 1$, i.e., $\mathbb{F}_{q^m} = \mathbb{F}_q(\alpha)$. The BCH code over $\mathbb{F}_q$ of design distance d is defined as the $\mathbb{F}_q$ kernel of the matrix

$$H := \begin{pmatrix} 1 & \alpha^{b} & \alpha^{2b} & \cdots & \cdots & \alpha^{(N-1)b} \\ 1 & \alpha^{b+1} & \alpha^{2(b+1)} & \cdots & \cdots & \alpha^{(N-1)(b+1)} \\ \vdots & \vdots & \vdots & & & \vdots \\ 1 & \alpha^{b+d-2} & \alpha^{2(b+d-2)} & \cdots & \cdots & \alpha^{(N-1)(b+d-2)} \end{pmatrix}. \tag{4.1}$$

We will denote this code by $\mathrm{BCH}_q(N, d)$.

Note that $\mathrm{BCH}_q(N, d)$ is a linear subspace of $\mathbb{F}_q^N$. If $N = q^m - 1$, i.e., if α is a primitive of the splitting field $\mathbb{F}_{q^m}$, the code $\mathrm{BCH}_q(N, d)$ is often referred to as a *primitive* BCH *code* and if $b = 1$ then one speaks of a *narrow-sense* BCH *code*.

The following result is well known (see, e.g., [12, Ch.7, Sec. 6, Theorem 10]) and easy to verify.

Theorem 4.2: $\mathrm{BCH}_q(N, d) \subset \mathbb{F}_q^N$ is a cyclic code and it has designed distance at least d and dimension at least $N - m(d-1)$.

For the BCH construction which will follow it will be of importance that we determine the exact dimension of $\mathrm{BCH}_q(N,d)$. This will be established in the sequel.

We can identify the BCH code $\mathrm{BCH}_q(N,d) \subset \mathbb{F}_q^N$ with the set of polynomials $c(x) \in \mathbb{F}_q[x]$ of degree $\deg c(x) < N$ and having the property that $c(x)$ has roots at $(\alpha^b, \alpha^{b+1}, \cdots, \alpha^{b+d-2})$. Let $M_i(x) \in \mathbb{F}_q[x]$ be the minimal polynomial of α^i. The generator polynomial of the cyclic code $\mathrm{BCH}_q(N,d)$ is then equal to (see, e.g., [12, Ch.7, Sec. 6])

$$g(x) = \text{l.c.m.}\{M_b(x), M_{b+1}(x), \cdots, M_{b+d-2}(x)\}.$$

The following theorem describes now the dimension of $\mathrm{BCH}_q(N,d)$.

Theorem 4.3:

$$\dim_{\mathbb{F}_q} \mathrm{BCH}_q(N,d) = N - \deg g(x) \geq N - m(d-1).$$

Moreover, there exist integers $b \leq j_1 < \cdots < j_\ell \leq b+d-2$ such that

$$g(x) = M_{j_1}(x) \cdot \ldots \cdot M_{j_\ell}(x).$$

Proof: The dimension formula is given in [12, Ch. 7, Sec. 3, Theorem 1]. The selection of the indices $j_1, \cdots, j_\ell$ is accomplished by omitting any repetition among the irreducible factors of $\{M_b(x), M_{b+1}(x), \cdots, M_{b+d-2}(x)\}$. □

Let $\alpha^{j_1}, \cdots, \alpha^{j_\ell}$ be roots of $M_{j_1}(x), \cdots, M_{j_\ell}(x)$, respectively. Define

$$\mathcal{H} := \begin{pmatrix} 1 & \alpha^{j_1} & \alpha^{2j_1} & \cdots & \cdots & \alpha^{(N-1)j_1} \\ 1 & \alpha^{j_2} & \alpha^{2j_2} & \cdots & \cdots & \alpha^{(N-1)j_2} \\ \vdots & & \vdots & \vdots & & \vdots \\ 1 & \alpha^{j_\ell} & \alpha^{2j_\ell} & \cdots & \cdots & \alpha^{(N-1)j_\ell} \end{pmatrix}. \tag{4.2}$$

$\mathcal{H}$ is obtained from H by omitting a set of rows. The $\mathbb{F}_q$ kernel of $\mathcal{H}$ is equal to the $\mathbb{F}_q$ kernel of H by construction. It is also clear that no further rows can be omitted from $\mathcal{H}$ without changing the kernel.

For the BCH construction of this section it will be necessary to show that we can write the parity-check matrix $\mathcal{H}$ as a controllability matrix:

Lemma 4.4: Let k be a positive integer, let $\alpha, j_1, \cdots, j_\ell$ be as above, and let

$$A := \begin{pmatrix} \alpha^{kj_1} & 0 & \cdots & 0 \\ 0 & \alpha^{kj_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \alpha^{kj_\ell} \end{pmatrix}$$

$$B := \begin{pmatrix} 1 & \alpha^{j_1} & \alpha^{2j_1} & \cdots & \alpha^{(k-1)j_1} \\ 1 & \alpha^{j_2} & \alpha^{2j_2} & \cdots & \alpha^{(k-1)j_2} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha^{j_\ell} & \alpha^{2j_\ell} & \cdots & \alpha^{(k-1)j_\ell} \end{pmatrix}.$$

Then (A,B) forms a controllable pair and for any positive integer γ with $\ell < \gamma k \leq N$ one has that the $\mathbb{F}_q$ kernel of

$$\Phi_\gamma(A,B) = (B \; AB \cdots A^{\gamma-1}B)$$

forms a block code of length γk with designed distance d.

Proof: Direct consequence from the fact that $\Phi_\gamma(A,B)$ coincides with the first γk columns of $\mathcal{H}$. □

At this point, the matrices A, B are still defined over $\mathbb{F}_{q^m}$ and what we really need are matrices defined over the base field $\mathbb{F}_q$. As is well known, we can identify the field $\mathbb{F}_{q^m}$ with the vector space $(\mathbb{F}_q)^m$ and in this way we will have a way of rewriting the matrices A, B as enlarged matrices over $\mathbb{F}_q$. Unfortunately, the situation is not so easy since we will lose in general the controllability of this "blown-up matrix pair" $\hat{A}, \hat{B}$.

If one does the process of field extension carefully and takes into consideration the degrees of the minimal polynomials $M_{j_s}(x)$ of each element $\alpha^{j_s}, 1 \leq s \leq \ell$, it is possible to arrive at a controllable pair $\hat{A}, \hat{B}$ defined over $\mathbb{F}_q$ whose controllability matrix has designed distance d. For reasons of readability we choose not to work in this full generality and we prefer to make certain restrictions which will ultimately guarantee that all irreducible polynomials $\{M_{j_1}(x), M_{j_2}(x), \cdots, M_{j_\ell}(x)\}$ have degree m.

Assume α is a primitive of $\mathbb{F}_{q^m}$. It is well known that if $\alpha^i \in \mathbb{F}_{q^m}$ is a root of some polynomial $c(x)$ then

$$\alpha^{qi}, \alpha^{q^2 i}, \cdots, \alpha^{q^m i} = \alpha^i \tag{4.3}$$

are roots as well. The set $\{i, qi, q^2 i, \cdots, q^{m-1} i\}$ is often referred to as a *cyclotomic coset*. The cardinality of the set of roots given in (4.3) is simply the degree of the minimal polynomial $M_i(x)$ and in general it is not true that this degree is m. The following Lemma provides a simple sufficient condition. This Lemma is a straightforward generalization of [12, Ch. 9, Sect. 3].

Lemma 4.5: Assume that

$$0 < i < q^{\lceil m/2 \rceil}.$$

Then the m numbers of $\mathbb{F}_{q^m}$ described in (4.3) are pairwise-different.

Proof: Write the integer i to the base q as

$$i = i_{m-1} q^{m-1} + \cdots + i_1 q + i_0.$$

In this way we can identify the integer i with the m-vector $(i_{m-1}, \cdots, i_0)$. The multiplication by q modulo $q^m - 1$ corresponds then to a cyclic left shift of the vector. Under the assumption of the integer i we know that the first $\lfloor m/2 \rfloor$ components of the corresponding m-vector are zero. Therefore, there will be m cyclic shifts needed until the vector repeats itself for the first time. □

This lemma will allow us to determine the dimension of $\mathrm{BCH}_q(N,d)$ more exactly under certain technical conditions.

Lemma 4.6: Consider the BCH code defined by (4.1). If

$$b + d - 2 < q^{\lceil m/2 \rceil}$$

then the irreducible polynomials $\{M_{j_1}(x), \cdots, M_{j_\ell}(x)\}$ all have degree m. In particular, the dimension

$$\dim_{\mathbb{F}_q} \mathrm{BCH}_q(N,d) = N - m\ell.$$

Remark 4.7: The Lemma is a generalization of [12, Ch. 9, Sec. 3, Corollary 8].

Proof: Because of Lemma 4.5 each irreducible factor must have degree m. Since the generator polynomial of $\mathrm{BCH}_q(N,d)$ is equal to $g(x) = M_{j_1}(x) \cdot \ldots \cdot M_{j_\ell}(x)$ the claimed dimension formula is established. □

For narrow-sense BCH codes we can give a more exact dimension estimate:

Lemma 4.8: Consider the BCH code defined by (4.1) having $b = 1$. If

$$d - 2 < q^{\lceil m/2 \rceil} - 1$$

then the dimension

$$\dim_{\mathbb{F}_q} \mathrm{BCH}_q(N,d) = N - m\left(d - 1 - \left\lfloor \frac{d-1}{q} \right\rfloor\right).$$

Proof: The indices $j_1 < \cdots < j_\ell$ generating the different cyclotomic cosets are in this case given by $1, 2, \cdots, q-1, q+1, \cdots$. By assumption they are all different and have cardinality m. □

With this preparation we will now be able to assume that under certain conditions all minimal polynomials $M_{j_s}(x)$ have cardinality m. As is done in the classical BCH construction, we can identify $\mathbb{F}_{q^m}$ with the vector space F_q^m. For this note that $\mathbb{F}_{q^m} = \mathbb{F}_q[\alpha]$ and that $1, \alpha, \cdots, \alpha^{m-1}$ forms a $\mathbb{F}_q$-basis. We will identify this $\mathbb{F}_q$-basis with the standard basis of F_q^m, i.e., we make the identification with the following column vectors:

$$1 \leftrightarrow \begin{pmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix}, \alpha \leftrightarrow \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \cdots, \alpha^{m-1} \leftrightarrow \begin{pmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{pmatrix}. \tag{4.4}$$

If $\beta \in \mathbb{F}_{q^m}$ is a particular element then denote with $[\beta]$ the corresponding column vector in F_q^m under above identification. Clearly, addition inside $\mathbb{F}_{q^m}$ corresponds to vector addition inside F_q^m. What about multiplication by the element α? For this, note that multiplication by α is a $\mathbb{F}_q$ linear transformation inside $\mathbb{F}_{q^m}$. This suggests that there exists an invertible linear transformation $L_\alpha \in GL_m(\mathbb{F}_q)$ describing this multiplication. The following lemma makes this precise.

Lemma 4.9: Let

$$f(x) = f_0 + f_1 x + f_2 x^2 + \cdots + f_{m-1} x^{m-1} + x^m \in \mathbb{F}_q[x]$$

be an irreducible monic polynomial of degree m. Let α be a root of $f(x)$, and let L_α be the companion matrix for $f(x)$ defined by

$$L_\alpha = \begin{pmatrix} 0 & \cdots & \cdots & 0 & -f_0 \\ 1 & \ddots & & \vdots & -f_1 \\ 0 & 1 & \ddots & \vdots & -f_2 \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & 1 & -f_{m-1} \end{pmatrix}. \tag{4.5}$$

Then multiplication by α inside $\mathbb{F}_{q^m}$ corresponds to left multiplication by L_α inside $\mathbb{F}_q^m$.

Proof: It is enough to verify the statement for the basis elements $1, \alpha, \cdots, \alpha^{m-1}$. For these elements it is clear that multiplication of α^i by α corresponds to multiplication of the vector in (4.4) by L_α. □

Remark 4.10: Since $\mathbb{F}_{q^m} = \mathbb{F}_q[\alpha]$ the assignment $\alpha \mapsto L_\alpha$ extends to an isomorphism $\mathbb{F}_q[\alpha] \cong \mathbb{F}_q[L_\alpha]$. In this way, we obtain a known embedding (compare with [10, Ch. 2.5]) of $\mathbb{F}_{q^m}$ into the matrix ring $GL_m(\mathbb{F}_q)$.

Theorem 4.11: Consider the BCH code defined by (4.1). If $b + d - 2 < q^{\lceil m/2 \rceil}$ then the matrices

$$\hat{A} := \begin{pmatrix} (L_\alpha)^{kj_1} & 0 & \cdots & 0 \\ 0 & (L_\alpha)^{kj_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & (L_\alpha)^{kj_\ell} \end{pmatrix}$$

$$\hat{B} := \begin{pmatrix} [1] & [\alpha^{j_1}] & [\alpha^{2j_1}] & \cdots & [\alpha^{(k-1)j_1}] \\ [1] & [\alpha^{j_2}] & [\alpha^{2j_2}] & \cdots & [\alpha^{(k-1)j_2}] \\ \vdots & \vdots & \vdots & & \vdots \\ [1] & [\alpha^{j_\ell}] & [\alpha^{2j_\ell}] & \cdots & [\alpha^{(k-1)j_\ell}] \end{pmatrix}$$

having sizes $m\ell \times m\ell$ and $m\ell \times k$, respectively, define a controllable pair. Moreover, if the integer γ satisfies $m\ell < k\gamma \leq N$ then the block code defined by the parity-check matrix

$$\Phi_\gamma(\hat{A}, \hat{B}) = \begin{pmatrix} \hat{B} & \hat{A}\hat{B} \cdots \hat{A}^{\gamma-1}\hat{B} \end{pmatrix}$$

has designed distance at least d.

Proof: Direct consequence of Theorem 4.2 and Lemma 4.4. □

As soon as we can exhibit an $(n-k) \times m\ell$ matrix $\hat{C}$ such that $(\hat{A}, \hat{C})$ forms an observable pair we will have constructed an observable convolutional code of designed distance d and degree $m\ell \leq m(d-1)$ as we will show in a moment. We will show first how to construct such a matrix $\hat{C}$.

Let

$$\tilde{C} := \begin{pmatrix} [1] & [\alpha^{j_1}] & [\alpha^{2j_1}] & \cdots & [\alpha^{(n-k-1)j_1}] \\ [1] & [\alpha^{j_2}] & [\alpha^{2j_2}] & \cdots & [\alpha^{(n-k-1)j_2}] \\ \vdots & \vdots & \vdots & & \vdots \\ [1] & [\alpha^{j_\ell}] & [\alpha^{2j_\ell}] & \cdots & [\alpha^{(n-k-1)j_\ell}] \end{pmatrix}.$$

Lemma 4.12: If $n - k \geq \gcd(k, q^m - 1)$ then $(\hat{A}, \tilde{C})$ forms a controllable pair, in particular, $(\hat{A}^t, \tilde{C}^t)$ forms an observable pair.

Proof: Let γ be an integer satisfying $m\ell < k\gamma \leq N$. Then the matrix $\Phi_\gamma(\hat{A}, \hat{B})$ appearing in Theorem 4.11 has full rank $m\ell$. We claim that every column of $\Phi_\gamma(\hat{A}, \hat{B})$ also appears in $\Phi_i(\hat{A}, \tilde{C})$ for a sufficiently large integer i. This would establish that $(\hat{A}, \tilde{C})$ forms a controllable pair and hence the lemma.

If $n - k \geq k$, the claim is trivially true. Otherwise, the exponents of the first row of $\Phi_i(\hat{A}, \tilde{C})$ have the form $0, j_1, 2j_1, \cdots, (n-k+1)j_1, kj_1, (k+1)j_1, \cdots$. Although some integers seem to be missing, we observe that "modulo

$(q^m - 1)$" all integers of the top row of $\Phi_\gamma(\hat{A}, \hat{B})$ indeed appear. For this, consider the factor ring

$$\mathbb{Z}_{q^m-1} = \mathbb{Z}/(q^m - 1)\mathbb{Z}.$$

Under the condition $n - k \geq \gcd(k, q^m - 1)$ it follows that

$$\{j + k\zeta \in \mathbb{Z}_{q^m-1} | 0 \leq j \leq n-k-1, \zeta \in \mathbb{Z}\} = \mathbb{Z}_{q^m-1}. \quad (4.6)$$

This establishes the claim. □

The matrices $\hat{A}$ and $\hat{A}^t$ are similar and therefore an invertible matrix S exists such that $\hat{A}^t = S\hat{A}S^{-1}$. Let

$$\hat{C} := \tilde{C}^t S^{-1}. \quad (4.7)$$

The main theorem then states.

Theorem 4.13: Let b, k, n, d be fixed positive integers with $k < n$. Choose m such that $q^m > km(d^2 - d)$ and $q^{\lceil m/2 \rceil} > b + d - 2$ and such that $n - k \geq \gcd(k, q^m - 1)$. Let $N = q^m - 1$ and let α be a primitive Nth root of unity. Let $\hat{A}$, $\hat{B}$ be defined as in Theorem 4.11 and let $\hat{C}$ be defined as in (4.7). Finally, let $\hat{D}$ be any $(n-k) \times k$ constant matrix. Then the 4-tuple of matrices $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$ defines an observable convolutional code of designed distance at least d and degree at most $m(d-1)$.

Proof: By Theorem 4.11, $(\hat{A}, \hat{B})$ is a controllable pair. Since $n - k \geq \gcd(k, q^m - 1)$, we have that $(\hat{A}^t, \tilde{C}^t)$ forms an observable pair. It follows that for every invertible matrix $S \in Gl_{m\ell}$ also $(S\hat{A}^t S^{-1}, \tilde{C}^t S^{-1})$ forms an observable pair, in particular $(\hat{A}, \hat{C})$ is an observable pair.

We will apply Theorem 3.1 to show that $\mathcal{C}(\hat{A}, \hat{B}, \hat{C}, \hat{D})$ has distance at least d. Since $(\hat{A}, \hat{C})$ is an observable pair it follows that the observability index ν of $(\hat{A}, \hat{C})$ is at most $m\ell$, the size of the matrix $\hat{A}$. The matrix $\Phi_{dm\ell}(\hat{A}, \hat{B})$ has

$$kdm\ell \leq km(d^2 - d) \leq N$$

columns. By Theorem 4.11, $\Phi_{dm\ell}(\hat{A}, \hat{B})$ defines the parity-check matrix of a block code of distance at least d. Therefore, the theorem follows directly from Theorem 3.1. □

For the particular construction of a convolutional code it is important to compute a transformation matrix S in an explicit form. One way of doing this was shown in [27].

Again, the question of how good these codes are arises. This requires that we be more specific about the degree of the constructed code. Over the binary field we can do this in a precise form:

Lemma 4.14: Assumptions as in Theorem 4.13 and $q = 2$. Then the code defined by $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$ has degree $\delta = \lceil m/2 \rceil (d-1)$.

Proof: Apply Lemma 4.6 in the situation $q = 2$. □

In contrast to the codes presented in Section III we cannot say if the codes are asymptotically good in the sense of (3.2). In case the codes have a distance of not much more than the designed distance d then it seems that

$$\lim_{\delta \to \infty} \frac{d_f(\mathcal{C})}{\delta} = 0 \quad (4.8)$$

and, in analogy to the block code situation, the presented BCH convolutional codes would be asymptotically bad. At this point, however, we cannot prove such a result.

We conclude the paper with an illustrative example:

Example 4.15: Continuing with Example 3.5 we want to design a code of rate $9/10$ and distance 81 over the binary field. We choose $b = 1$. By Theorem 4.13 we have to find m such that $2^m > 9m(81^2 - 81)$ and $q^{\lceil m/2 \rceil} > 80$ are both satisfied. The smallest integer which satisfies these inequalities is $m = 21$.

The numbers $j_1, \cdots, j_\ell$ appearing in Theorem 4.11 are, therefore, equal to $1, 3, 5, \cdots, 21$ and $\ell = 11$. The calculated 4-tuple of matrices $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$, therefore, defines an observable convolutional code of rate $9/10$, designed distance 81 and degree $\delta = \ell(d-1) = 880$. The individual polynomial entries are, therefore, in the range of degree 100 which corresponds to the memory.

The decoding algorithm as presented in [21] can be applied, provided the Berlekamp–Massey algorithm for a BCH code with 880 syndromes can be performed.

ACKNOWLEDGMENT

The authors wish to thank B. Allen, H. Glüsing-Lüerssen, A. Loeliger, R. Smarandache, and P. Weiner for helpful discussions during the preparation of this paper. Helpful comments by the anonymous referees are acknowledged as well.

REFERENCES

[1] M. Esmaeili, T. A. Gulliver, N. P. Secord, and S. A. Mahmoud, "A link between quasicyclic codes and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 44, pp. 431–435, Jan. 1998.
[2] G. D. Forney, Jr., "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 720–738, Sept. 1970.
[3] ——, "Minimal bases of rational vector spaces, with applications to multivariable linear systems," *SIAM J. Contr. Optim.*, vol. 13, no. 3, pp. 493–520, 1975.
[4] T. W. Hungerford, *Algebra*. New York: Springer, 1980.
[5] R. Johannesson and Z. Wan, "A linear algebra approach to minimal convolutional encoders," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1219–1233, July 1993.
[6] R. Johannesson and K. Sh. Zigangirov, *Fundamentals of Convolutional Coding*. New York: IEEE Press, 1999.
[7] J. Justesen, "New convolutional code constructions and a class of asymptotically good time-varying codes," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 220–225, Mar. 1973.
[8] ——, "An algebraic construction of rate $1/\nu$ convolutional codes," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 577–580, Jan. 1975.
[9] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1980.
[10] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Their Applications*. Cambridge, London, U.K.: Cambridge Univ. Press, 1994, revised edition.
[11] S. Lin and D. Costello, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ, Prentice-Hall, 1983.
[12] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North Holland, 1977.
[13] J. L. Massey, D. J. Costello, and J. Justesen, "Polynomial weights and code constructions," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 101–110, Jan. 1973.
[14] J. L. Massey and M. K. Sain, "Codes, automata, and continuous systems: Explicit interconnections," *IEEE Trans. Automat. Contr.*, vol. AC-12, no. 6, pp. 644–650, 1967.

[15] R. J. McEliece, "The algebraic theory of convolutional codes," in *Handbook of Coding Theory*, R. Brualdi, W. C. Huffman, and V. Pless, Eds. Amsterdam, The Netherlands: Elsevier, 1998.
[16] H. F. Münzner and D. Prätzel-Wolters, "Minimal bases of polynomial modules, structural indices and Brunovsky-transformations," *Int. J. Contr.*, vol. 30, pp. 291–318, 1979.
[17] Ph. Piret, *Convolutional Codes, an Algebraic Approach*. Cambridge, MA: MIT Press, 1988.
[18] ——, "A convolutional equivalent to Reed-Solomon codes," *Philips J. Res.*, vol. 43, no. 3-4, pp. 441–458, 1988.
[19] V. M. Popov, "Invariant description of linear time-invariant controllable systems," *SIAM J. Contr. Optim.*, vol. 10, pp. 252–264, 1972.
[20] M. S. Ravi and J. Rosenthal, "A smooth compactification of the space of transfer functions with fixed McMillan degree," *Acta Appl. Math*, vol. 34, pp. 329–352, 1994.
[21] J. Rosenthal, "An algebraic decoding algorithm for convolutional codes," in *Dynamical Systems, Control, Coding, Computer Vision: New Trends, Interfaces, and Interplay*, G. Picci and D. S. Gilliam, Eds. Boston-Basel-Berlin: Birkäuser, 1999, pp. 343–360.
[22] J. Rosenthal and J. M. Schumacher, "Realization by inspection," *IEEE Trans. Automat. Contr.*, vol. AC-42, pp. 1257–1263, Sept. 1997.
[23] J. Rosenthal, J. M. Schumacher, and E. V. York, "On behaviors and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1881–1891, Nov. 1996.
[24] J. Rosenthal and R. Smarandache, "Construction of convolutional codes using methods from linear systems theory," in *Proc. 35th Annu. Allerton Conf. Communication, Control, and Computing*, 1997, pp. 953–960.
[25] ——, "A state space approach for constructing MDS rate $1/n$ convolutional codes," in *Proc. 1998 IEEE Information Theory Workshop on Information Theory* (Killarney, Kerry, Ireland, June 1998), pp. 116–117.
[26] R. M. Tanner, "Convolutional codes from quasicyclic codes: A link between the theories of block and convolutional codes," Computer Res. Lab., Tech. Rep., USC-CRL-87-21, Nov. 1987.
[27] E. V. York, "Algebraic description and construction of error correcting codes, a systems theory point of view," Ph.D. dissertation, Univ. Notre Dame, 1997. [Online]. Available WWW at: http://www.nd.edu/~rosen/preprints.html.