LUND UNIVERSITY

A linear algebra approach to minimal convolutional encoders

Johannesson, Rolf; Wan, Zhe-Xian

Published in:
IEEE Transactions on Information Theory

DOI:
10.1109/18.243440

1993

Link to publication

Citation for published version (APA):
Johannesson, R., & Wan, Z-X. (1993). A linear algebra approach to minimal convolutional encoders. *IEEE Transactions on Information Theory*, *39*(4), 1219-1233. https://doi.org/10.1109/18.243440

Total number of authors:
2

General rights
Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.
• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: https://creativecommons.org/licenses/

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

A Linear Algebra Approach to Minimal Convolutional Encoders

Rolf Johannesson, *Member, IEEE*, and Zhe-xian Wan

Abstract— **This semitutorial paper starts with a review of some of Forney's contributions on the algebraic structure of convolutional encoders on which some new results on minimal convolutional encoders rest. An example is given of a basic convolutional encoding matrix whose number of abstract states is minimal over all equivalent encoding matrices. However, this encoding matrix can be realized with a minimal number of memory elements neither in controller canonical form nor in observer canonical form. Thus, this encoding matrix is not minimal according to Forney's definition of a minimal encoder. To resolve this difficulty, the following three minimality criteria are introduced: *minimal-basic encoding matrix* (minimal overall constraint length over equivalent basic encoding matrices), *minimal encoding matrix* (minimal number of abstract states over equivalent encoding matrices), and *minimal encoder* (realization of a minimal encoding matrix with a minimal number of memory elements over all realizations). Among other results, it is shown that all minimal-basic encoding matrices are minimal, but that there exist (basic) minimal encoding matrices that are not minimal-basic! Several equivalent conditions are given for an encoding matrix to be minimal. It is also proven that the constraint lengths of two equivalent minimal-basic encoding matrices are equal one by one up to a rearrangement. All results are proven using only elementary linear algebra. Most important among the new results are a simple minimality test, the surprising fact that there exist basic encoding matrices that are minimal but not minimal-basic, the existence of basic encoding matrices that are nonminimal, and a recent result, due to Forney, that states exactly when a basic encoding matrix is minimal.**

Index Terms— **Convolutional code, basic encoding matrix, minimal-basic encoding matrix, minimal encoding matrix, minimal encoder.**

I. Introduction

FORNEY'S landmark paper: "Convolutional codes I: Algebraic structures" [1] (see also [2]) constitutes an abundant source of important results on the algebraic structure of convolutional codes. After having introduced a most important concept, viz., a *basic* convolutional encoder (which has both a polynomial encoding matrix and a polynomial right inverse), Forney defined a rate $R = b/c$ basic encoder to be *minimal* if its overall constraint length ν is equal to the maximum degree μ of the $b \times b$ subdeterminants of its encoding matrix.

In this semitutorial paper, we show that there exist basic encoding matrices that have a minimal number of abstract states and, hence, can be realized by a minimal number of memory elements over all equivalent encoders but are, quite surprisingly, *not* minimal according to Forney's definition! To resolve this difficulty we introduce the following three minimality criteria: *minimal-basic encoding matrix* (minimal overall constraint length over equivalent basic encoding matrices), *minimal encoding matrix* (minimal number of abstract states over equivalent encoding matrices), and *minimal encoder* (realization of a minimal encoding matrix with a minimal number of memory elements over all realizations). Our definition of a minimal-basic encoding matrix is equivalent to Forney's definition of a minimal encoder.

In Section II, we introduce the controller canonical and observer canonical forms of a linear sequential circuit. The distinction between convolutional encoders and their generator and encoding matrices is discussed in Section III. In Section IV, we discuss the equivalence of encoders and we also give Forney's definition of a basic encoder. The important concept of minimal-basic encoding matrices is introduced in Section V. In this section, we give three equivalent statements for a basic encoding matrix to be minimal-basic. We also prove that the constraint lengths of two equivalent minimal-basic encoding matrices are equal one by one up to a rearrangement. From the minimal-basic encoding matrix we proceed to introduce the minimality of a general encoding matrix in Section VI. Several equivalent conditions for a general encoding matrix to be minimal are given. After having defined a minimal encoder in Section VII, we give an example of a basic minimal encoding matrix that is not minimal-basic and, hence, has no minimal realization in controller canonical form. Finally, in Section VIII, we prove that every systematic encoding matrix is minimal. We also show a minimal realization of a systematic encoding matrix which has neither a minimal realization in controller canonical form nor one in observer canonical form.

Some of our theorems are new, others can be found explicitly in Forney's papers [1]–[3], and a few are given implicitly in these papers, but we have proven all results by using only elementary linear algebra. Most important among the new results are a simple minimality test, the surprising fact that there exist basic encoding matrices that are minimal but not minimal-basic, the existence of basic encoding matrices that are nonminimal, and a recent result, due to Forney, that states exactly when a basic encoding matrix is minimal.

II. Controller and Observer Canonical Forms

Let $\mathbf{F}_2((D))$ denote the *field of binary Laurent series*. The element $x(D) = \sum_{i=r}^{\infty} x_i D^i \in \mathbf{F}_2((D))$, $r \in \mathbf{Z}$, contains

Manuscript received June 19, 1991; revised November 17, 1992. This work was supported in part by the Swedish Research Council for Engineering Sciences under Grant 91–91.

R. Johannesson is with the Department of Information Theory, University of Lund, Box 118, S-221 00, Sweden.

Z.-x. Wan is with the Department of Information Theory, University of Lund, Box 118, S-221 00, Sweden. He is also with the Institute for Systems Science, Chinese Academy of Sciences, Beijing 1000 80, China.

IEEE Log Number 9209457.

0018–9448/93$03.00 © 1993 IEEE

only finitely many negative powers of D. For example,

$$x(D) = D^{-2} + 1 + D^3 + D^7 + D^{12} + \cdots$$

is a Laurent series.

Let $\mathbf{F}_2[[D]]$ denote the *ring of formal power series*. The element $f(D) = \sum_{i=0}^{\infty} f_i D^i \in \mathbf{F}_2[[D]]$ is a Laurent series without negative powers of D.

A *polynomial* $p(D) = \sum_{i=0}^{\infty} p_i D^i$ contains no negative and only finitely many positive powers of D. If $p_0 = 1$ we have a *delayfree* polynomial, e.g.,

$$p(D) = 1 + D^2 + D^3 + D^5$$

is a binary delayfree polynomial of degree 5. The set of binary polynomials $\mathbf{F}_2[D]$ is clearly a subset of $\mathbf{F}_2((D))$.

Given any pair of polynomials $x(D), y(D) \in \mathbf{F}_2[D]$, with $y(D) \neq 0$, we can obtain the element $x(D)/y(D) \in \mathbf{F}_2((D))$ by long division. Since sequences must start at some finite time we must identify, for instance, $(1+D)/D^2(1+D+D^2)$ with the series $D^{-2} + 1 + D + D^3 + \cdots$ instead of the alternative series $D^{-3} + D^{-5} + D^{-6} + \cdots$ that can also be obtained by long division but which is not a Laurent series. Obviously, all nonzero ratios $x(D)/y(D)$ are invertible, so they form the *field of binary rational functions* $\mathbf{F}_2(D)$, which is a subfield of the field of Laurent series $\mathbf{F}_2((D))$.

We can of course consider n-tuples of elements from $\mathbf{F}_2[D], \mathbf{F}_2[[D]], \mathbf{F}_2(D)$, or $\mathbf{F}_2((D))$. For example, $\boldsymbol{x}(D) = (\sum_{i=r_1}^{\infty} x_i^{(1)} D^i, \sum_{i=r_2}^{\infty} x_i^{(2)} D^i, \cdots, \sum_{i=r_n}^{\infty} x_i^{(n)} D^i)$, where $r_1, r_2, \cdots, r_n \in \mathbf{Z}$, is an element in $\mathbf{F}_2((D))^{(n)}$, the n-dimensional vector space over the field of binary Laurent series $\mathbf{F}_2((D))$. Let $r = \min\{r_1, r_2, \cdots, r_n\}$ and put $x_i^{(j)} = 0$ for $i < r_j$, then we can express $\boldsymbol{x}(D)$ also as

$$\boldsymbol{x}(D) = \sum_{i=r}^{\infty} (x_i^{(1)}, x_i^{(2)}, \cdots, x_i^{(n)}) D^i.$$

Thus, $\mathbf{F}_2((D))^{(n)} = \mathbf{F}_2^{(n)}((D))$, where $\mathbf{F}_2^{(n)}((D))$ is the set of all Laurent series in D with coefficients in the n-dimensional row vector space $\mathbf{F}_2^{(n)}$ over $\mathbf{F}_2$. The elements in the field $\mathbf{F}_2((D))$ are usually called scalars. Similarly, $\mathbf{F}_2[D]^{(n)} = \mathbf{F}_2^{(n)}[D], \mathbf{F}_2[[D]]^{(n)} = \mathbf{F}_2^{(n)}[[D]]$, etc. If $\boldsymbol{x}(D) \in \mathbf{F}_2^{(n)}[D]$, we say that $\boldsymbol{x}(D)$ is polynomial in D. The degree of the element $\boldsymbol{x}(D) = \sum_{i=0}^{m} (x_i^{(1)} x_i^{(2)} \cdots x_i^{(n)}) D^i$ is defined to be m, provided $(x_m^{(1)} x_m^{(2)} \cdots x_m^{(n)}) \neq (0\,0 \cdots 0)$. For simplicity, we also call elements in $\mathbf{F}_2^{(n)}[[D]]$ formal power series when $n > 1$.

Consider the *controller canonical form* of a single input single output linear system as shown in Fig. 1. The delay elements form a shift register, the output is a linear function of the input and the shift register contents, and the input to the shift register is a linear function of the input and the shift register contents.

From Fig. 1, it follows that

$$v(D) = u(D)g(D)/q(D), \tag{1}$$

where

$$g(D) = g_0 + g_1 D + \cdots + g_m D^m \tag{2}$$

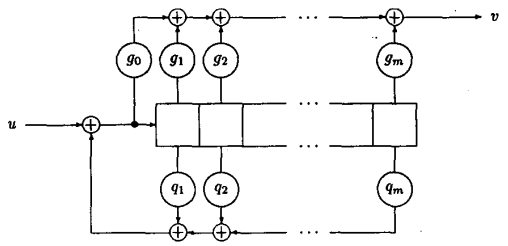

Fig. 1. The controller canonical form of a rational transfer function.

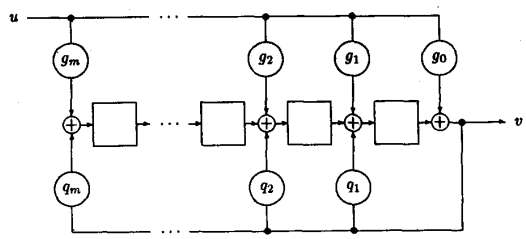

Fig. 2. The observer canonical form of a rational transfer function.

and

$$q(D) = 1 + q_1 D + \cdots + q_m D^m. \tag{3}$$

Let $t(D) = g(D)/q(D)$, then $v(D) = u(D)t(D)$ and we say that $t(D)$ is a *rational transfer function* which transfers the input $u(D)$ into the output $v(D)$. From (1), it follows that every rational function with a constant term 1 in the denominator polynomial $q(D)$ (or, equivalently, with $q(0) = 1$ or, again equivalently, with $q(D)$ delayfree) is a rational transfer function that can be realized in the canonical form shown in Fig. 1. Every rational function $g(D)/q(D)$, where $q(D)$ is delayfree, is called a *realizable function*.

In general, a matrix $G(D)$ whose entries are rational functions is called a *rational transfer function matrix*. A rational transfer function matrix $G(D)$ for a linear system with many inputs and/or many outputs whose entries are realizable functions is called *realizable*.

In practice, given a rational transfer function matrix we have to realize it by linear sequential circuits. It can be realized in many different ways. For instance, the realizable function

$$\frac{g_0 + g_1 D + \cdots + g_m D^m}{1 + q_1 D + \cdots + q_m D^m} \tag{4}$$

has the controller canonical form illustrated in Fig. 1. On the other hand, since the circuit in Fig. 2 is linear, we have

$$\begin{aligned} v(D) = {} & u(D)(g_0 + g_1 D + \cdots + g_m D^m) \\ & + v(D)(q_1 D + \cdots + q_m D^m), \end{aligned} \tag{5}$$

which is the same as (1). Thus, Fig. 2 is also a realization of (4). In this realization the delay elements do not in general form a shift register as these delay elements are separated by adders. This is the so-called *observer canonical form* of the rational function (4). The controller and observer canonical forms in Fig. 1 and 2, respectively, are two different realizations of the same rational transfer function.

III. Convolutional Codes and Their Encoders

We are now prepared to give a formal definition of a convolutional transducer.

Definition: A rate $R = b/c$ (binary) *convolutional transducer* over the field of rational functions $\mathbf{F}_2(D)$ is a linear mapping

$$\begin{aligned} \mathbf{F}_2^b((D)) &\to \mathbf{F}_2^c((D)) \\ \boldsymbol{u}(D) &\mapsto \boldsymbol{v}(D), \end{aligned} \tag{6}$$

which can be represented as

$$\boldsymbol{v}(D) = \boldsymbol{u}(D)G(D), \tag{7}$$

where $G(D)$ is a $b \times c$ transfer function matrix of rank b with entries in $\mathbf{F}_2(D)$ and $\boldsymbol{v}(D)$ is called a *code sequence* arising from the *information sequence* $\boldsymbol{u}(D)$.

Obviously we must be able to reconstruct the information sequence $\boldsymbol{u}(D)$ from the code sequence $\boldsymbol{v}(D)$ when there is no noise on the channel. Otherwise the convolutional transducer would be useless. Therefore we require that the transducer map is injective, i.e., the transfer function matrix $G(D)$ has rank b over the field $\mathbf{F}_2(D)$.

Definition: A rate $R = b/c$ *convolutional code* $\mathcal{C}$ over $\mathbf{F}_2$ with the $b \times c$ matrix $G(D)$ of rank b over $\mathbf{F}_2(D)$ as its *generator matrix* is the image set of a rate $R = b/c$ convolutional transducer with $G(D)$ as its transfer function matrix.

It follows immediately from the definition that a rate $R = b/c$ convolutional code $\mathcal{C}$ over $\mathbf{F}_2$ with the $b \times c$ matrix $G(D)$ of rank b over $\mathbf{F}_2(D)$ as a generator matrix can be regarded as the $\mathbf{F}_2((D))$ row space of $G(D)$. Hence, it can also be regarded as the rate $R = b/c$ block code over the infinite field of Laurent series which has $G(D)$ as its generator matrix.

We call a realizable transfer function matrix $G(D)$ *delayfree* if at least one of its entries $g(D)/q(D)$ has $g(0) \neq 0$. If $G(D)$ is not delayfree it can be written as

$$G(D) = D^i G_d(D), \tag{8}$$

where $i \geq 1$ and $G_d(D)$ is delayfree.

Definition: The generator matrix of a convolutional code over $\mathbf{F}_2$ is called an *encoding matrix* of the code if it is (realizable and) delayfree.

Definition: A rate $R = b/c$ *convolutional encoder* of a convolutional code with encoding matrix $G(D)$ over $\mathbf{F}_2(D)$ is a realization by a linear sequential circuit of a rate $R = b/c$ convolutional transducer whose transfer function matrix is $G(D)$.

Theorem 1: Every convolutional code $\mathcal{C}$ has an encoding matrix.

Proof: Let $G(D)$ be any generator matrix for $\mathcal{C}$. The nonzero entries of $G(D)$ can be written

$$D^{s_{ij}} g_{ij}(D)/q_{ij}(D), \tag{9}$$

where s_{ij} is an integer such that $g_{ij}(0) = q_{ij}(0) = 1$, $1 \leq i \leq b$, $1 \leq j \leq c$. The number s_{ij} is called the *start* of the sequence

$$\begin{aligned} t(D) &= D^{s_{ij}} g_{ij}(D)/q_{ij}(D) \\ &= D^{s_{ij}} + t_{s_{ij}+1} D^{s_{ij}+1} + \cdots. \end{aligned} \tag{10}$$

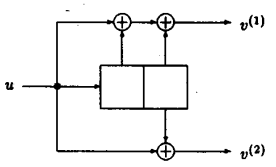

Fig. 3. A rate $R = 1/2$ convolutional encoder with encoding matrix $G_0(D)$.

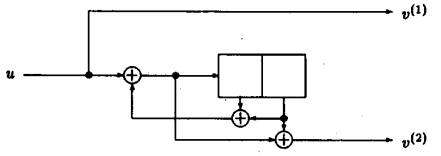

Fig. 4. A rate $R = 1/2$ systematic convolutional encoder with feedback and encoding matrix $G_1(D)$.

Let $s = \min_{i,j}\{s_{ij}\}$. Clearly

$$G'(D) = D^{-s} G(D) \tag{11}$$

is both delayfree and realizable. Since D^{-s} is a scalar in $\mathbf{F}_2((D))$, both $G(D)$ and $G'(D)$ generate the same convolutional code. Therefore $G'(D)$ is an encoding matrix for the convolutional code $\mathcal{C}$. □

Including nondelayfree generator matrices in the set of convolutional encoders would only clutter up the analysis without any benefits.

A given convolutional code can be encoded by many essentially different encoders. For example, consider the rate $R = 1/2$, binary convolutional code with the basis vector $\boldsymbol{v}_0(D) = (1 + D + D^2 \; 1 + D^2)$. The simplest encoder for this code has the encoding matrix

$$G_0(D) = (1 + D + D^2 \; 1 + D^2). \tag{12}$$

Its controller canonical form is shown in Fig. 3.

If we choose the basis to be $\boldsymbol{v}_1(D) = a_1(D)\boldsymbol{v}_0(D)$, where the scalar $a_1(D)$ is the rational function $a_1(D) = 1/(1 + D + D^2)$, we obtain the encoding matrix

$$G_1(D) = \left(1 \quad \frac{1+D^2}{1+D+D^2}\right), \tag{13}$$

which is a *systematic* encoding matrix for the same code (Fig. 4). When a rate $R = b/c$ convolutional code is encoded by a systematic encoding matrix, the b input sequences appear unchanged among the c output sequences.

For example, the output sequence $\boldsymbol{v}(D) = (v^{(1)}(D) \; v^{(2)}(D))$ of the systematic convolutional encoder with encoding matrix $G_1(D)$ shown in Fig. 4 can be written as

$$\begin{aligned} v^{(1)}(D) &= u(D) \\ v^{(2)}(D) &= u(D)\frac{1 + D^2}{1 + D + D^2}. \end{aligned} \tag{14}$$

If a convolutional code $\mathcal{C}$ is encoded by a systematic encoding matrix we can always permute its columns and obtain an

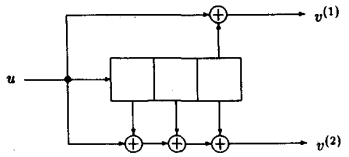

Fig. 5. A rate $R = 1/2$ catastrophic convolutional encoder with encoding matrix $G_2(D)$.

encoding matrix for an *equivalent* convolutional code C' such that the b information sequences appear unchanged among the *first* c code sequences. Thus, without loss of generality, a systematic encoding matrix can be written

$$G(D) = (I_b \; R(D)), \tag{15}$$

where I_b is a $b \times b$ identity matrix and $R(D)$ a $b \times (c-b)$ matrix whose entries are rational functions of D.

Being "systematic" is an encoding matrix property, not a code property. Every convolutional code has both systematic and *nonsystematic* encoding matrices.

If we further change the basis to $\boldsymbol{v}_2(D) = a_2(D)\boldsymbol{v}_0(D)$, where $a_2(D) \in \mathbf{F}_2(D)$ is chosen as $a_2(D) = 1 + D$, we obtain a third encoding matrix for the same code, viz.,

$$G_2(D) = (1 + D^3 \quad 1 + D + D^2 + D^3). \tag{16}$$

In Fig. 5, we show its controller canonical form.

Theorem 2: Every convolutional code C has a polynomial encoding matrix.

Proof: Let $G(D)$ be any encoding matrix for C and let $q(D)$ be the least common multiple of all the denominators in (9). Since $q(D)$ is a delayfree scalar in $\mathbf{F}_2((D))$,

$$G'(D) = q(D)G(D) \tag{17}$$

is a polynomial encoding matrix for C. □

An encoder whose encoding matrix is polynomial is called a polynomial encoder.

For convolutional codes, the choice of the encoding matrix is of great importance.

Definition: A generator matrix for a convolutional code is *catastrophic*[1] [4] if there exists an information sequence $\boldsymbol{u}(D)$ with infinitely many nonzero digits, $w_H(\boldsymbol{u}(D)) = \infty$, that results in codewords $\boldsymbol{v}(D)$ with only finitely many nonzero digits, $w_H(\boldsymbol{v}(D)) < \infty$.

Example 1: The third encoding matrix for the convolutional code given above, viz.,

$$G_2(D) = (1 + D^3 \quad 1 + D + D^2 + D^3)$$

is catastrophic since $\boldsymbol{u}(D) = 1/(1+D) = 1+D+D^2+\cdots$ has $w_H(\boldsymbol{u}(D)) = \infty$ but $\boldsymbol{v}(D) = \boldsymbol{u}(D)G_2(D) = (1+D+D^2 \; 1+D^2) = (1\;1) + (1\;0)D + (1\;1)D^2$ has $w_H(\boldsymbol{v}(D)) = 5 < \infty$.

When a catastrophic encoding matrix is used for encoding, finitely many errors (five in the previous example) in the estimate $\widehat{\boldsymbol{v}}(D)$ of the transmitted codeword $\boldsymbol{v}(D)$ can lead to infinitely many errors in the estimate $\widehat{\boldsymbol{u}}(D)$ of the information sequence $\boldsymbol{u}(D)$—a "catastrophic" situation that must be avoided!

[1]The term "catastrophic" does not actually appear in [4]; it seems to have been introduced by Massey in a seminar in 1969.

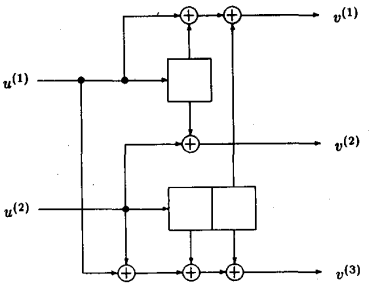

Fig. 6. A rate $R = 2/3$ convolutional encoder.

Being "catastrophic" is a generator matrix property, not a code property. Every convolutional code has both catastrophic and noncatastrophic generator matrices.

We define the *constraint length for the ith input* of a polynomial convolutional encoding matrix as [1]

$$\nu_i = \max_{1 \le j \le c} \{\deg g_{ij}(D)\}, \tag{18}$$

the *memory* m of the polynomial encoding matrix as the maximum of the constraint lengths, i.e.,

$$m = \max_{1 \le i \le b} \{\nu_i\}, \tag{19}$$

and the *overall constraint length* as the sum of the constraint lengths [1]

$$\nu = \sum_{i=1}^{b} \nu_i. \tag{20}$$

The polynomial encoding matrix can be realized by a linear sequential circuit consisting of b shift registers, the ith of length ν_i, with the outputs formed as modulo-2 sums of the appropriate shift register contents. For example, in Fig. 6 we have shown the controller canonical form of the polynomial encoding matrix

$$G(D) = \begin{pmatrix} 1+D & D & 1 \\ D^2 & 1 & 1+D+D^2 \end{pmatrix}, \tag{21}$$

whose constraint lengths of the 1st and 2nd inputs are 1 and 2, respectively, and whose overall constraint length is 3.

The number of memory elements required for the controller canonical form is equal to the overall constraint length.

We define a *physical state* $\boldsymbol{\sigma}$ of a realization of a rational encoding matrix $G(D)$ at some time instant to be the contents of its memory elements. If $G(D)$ is polynomial, then the dimension of the physical state space of its controller canonical form is equal to the overall constraint length ν.

For a rational encoding matrix $G(D)$ we define the *abstract state* $\boldsymbol{s}(D)$ associated with an input sequence $\boldsymbol{u}(D)$ to be the sequence of outputs at time 0 and later, which are due to that part of $\boldsymbol{u}(D)$ that occurs up to time -1, and to the all zero inputs thereafter. The abstract state depends only on $G(D)$ and not on its realization. Distinct abstract states must spring

from distinct physical states at time 0. Clearly, the number of physical states is greater than or equal to the number of abstract states.

Let P be the projection operator that truncates sequences to end at time -1, and $Q = 1 - P$ the projection operator that truncates sequences to start at time 0, i.e., if

$$\boldsymbol{u}(D) = \boldsymbol{u}_d D^d + \boldsymbol{u}_{d+1} D^{d+1} + \cdots,$$

then

$$\boldsymbol{u}(D)P = \begin{cases} \boldsymbol{u}_d D^d + \cdots + \boldsymbol{u}_{-1} D^{-1}, & d < 0, \\ \boldsymbol{0}, & d \geq 0, \end{cases} \tag{22}$$

and

$$\boldsymbol{u}(D)Q = \boldsymbol{u}_0 + \boldsymbol{u}_1 D + \boldsymbol{u}_2 D^2 + \cdots. \tag{23}$$

Clearly,

$$P + Q = 1. \tag{24}$$

Thus, the abstract state $\boldsymbol{s}(D)$ of $\boldsymbol{u}(D)$ can be written concisely as

$$\boldsymbol{s}(D) = \boldsymbol{u}(D)PG(D)Q. \tag{25}$$

Note that in an observer canonical form of a polynomial encoding matrix the abstract states are in $1-1$ correspondence with the physical states since the contents of the memory elements are simply shifted out in the absence of inputs.

Since the abstract state does not depend on the realization we have the same abstract states in the observer canonical form as in the controller canonical form.

IV. Equivalence of Encoders and Basic Encoders

In a communication context it is natural to say that two encoders are equivalent if they generate the same code $\mathcal{C}$. It is therefore important to look for encoders with the lowest complexity within the class of equivalent encoders.

Definition: Two convolutional encoding matrices $G(D)$ and $G'(D)$ are called *equivalent* if they encode the same code. Two convolutional encoders are called *equivalent* if their encoding matrices are equivalent.

Theorem 3: Two rate $R = b/c$ convolutional encoding matrices $G(D)$ and $G'(D)$ are equivalent, if and only if there is a $b \times b$ nonsingular matrix $T(D)$ over $\mathbf{F}_2(D)$ such that

$$G(D) = T(D)G'(D). \tag{26}$$

Proof: If (26) holds, then clearly $G(D)$ and $G'(D)$ are equivalent.

Conversely, suppose that $G(D)$ and $G'(D)$ are equivalent. Let $\boldsymbol{g}_i(D) \in \mathbf{F}_2(D)^{(c)}$ be the ith row of $G(D)$. Then there exists a $\boldsymbol{u}_i(D) \in \mathbf{F}_2((D))^{(b)}$ such that

$$\boldsymbol{g}_i(D) = \boldsymbol{u}_i(D)G'(D). \tag{27}$$

Let

$$T(D) = \begin{pmatrix} \boldsymbol{u}_1(D) \\ \boldsymbol{u}_2(D) \\ \vdots \\ \boldsymbol{u}_b(D) \end{pmatrix}. \tag{28}$$

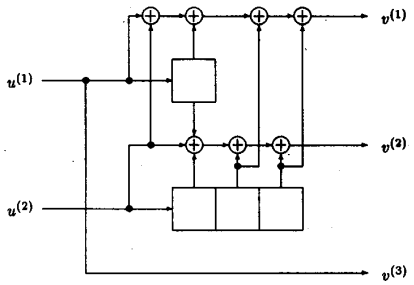

Fig. 7. Controller canonical form of the encoding matrix $G'(D)$.

Then

$$G(D) = T(D)G'(D), \tag{29}$$

where $T(D)$ is a $b \times b$ matrix over $\mathbf{F}_2((D))$. Let $S'(D)$ be a $b \times b$ nonsingular submatrix of $G'(D)$ and $S(D)$ be the corresponding $b \times b$ submatrix of $G(D)$. Then $S(D) = T(D)S'(D)$. Thus $T(D) = S(D)S'(D)^{-1}$ and, hence, $T(D)$ is over $\mathbf{F}_2(D)$. Since $G(D)$, being an encoding matrix, is of rank b it follows that $T(D)$ is also of rank b and, hence, is nonsingular. □

Example 2: By Theorem 3 the encoding matrix for the rate $R = 2/3$ convolutional encoder shown in Fig. 6

$$G(D) = \begin{pmatrix} 1+D & D & 1 \\ D^2 & 1 & 1+D+D^2 \end{pmatrix}$$

is equivalent to the encoding matrix

$$G'(D) = \begin{pmatrix} 1+D & D & 1 \\ 1+D^2+D^3 & 1+D+D^2+D^3 & 0 \end{pmatrix},$$

since there is a nonsingular matrix

$$T(D) = \begin{pmatrix} 1 & 0 \\ 1+D+D^2 & 1 \end{pmatrix}$$

such that $G'(D) = T(D)G(D)$. The controller canonical form of $G'(D)$ is shown in Fig. 7. Since $G(D)$ and $G'(D)$ are equivalent, the encoders in Figs. 6 and 7 encode the same code.

Definition: (Forney [1]) A convolutional encoding matrix is called *basic* if it is polynomial and it has a polynomial right inverse. A convolutional encoder is called *basic* if its encoding matrix is basic.

Next, we consider the invariant-factor decomposition of a rational matrix [1], [5].

Invariant-Factor Theorem: Let $G(D)$ be a $b \times c$, $b \leq c$, binary rational matrix of rank r. Then $G(D)$ can be written in the following manner:

$$G(D) = A(D)\Gamma(D)B(D), \tag{30}$$

where $A(D)$ and $B(D)$ are $b \times b$ and $c \times c$, respectively, binary unimodular matrices and where $\Gamma(D)$ is the $b \times c$ matrix

$$\Gamma(D) = \begin{pmatrix} \frac{\alpha_1(D)}{\beta_1(D)} & & & & & & \\ & \frac{\alpha_2(D)}{\beta_2(D)} & & & & & \\ & & \ddots & & & & \\ & & & \frac{\alpha_r(D)}{\beta_r(D)} & & & 0^{(b,c-b)} \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{pmatrix},$$

in which $0^{(b,c-b)}$ is a $b \times (c-b)$ zero matrix, $\alpha_i(D)$ and $\beta_i(D)$ are nonzero binary polynomials satisfying $\alpha_i(D) \mid \alpha_{i+1}(D)$ and $\beta_{i+1}(D) \mid \beta_i(D)$, $i = 1, 2, \cdots, r-1$.

A square binary polynomial matrix is called *unimodular* if its determinant is 1. The unimodular matrices are uniquely characterized as the square polynomial matrices that have polynomial inverses. In an invariant-factor decomposition the unimodular matrices $A(D)$ and $B(D)$ are not unique, although the $\frac{\alpha_i(D)}{\beta_i(D)}$'s are uniquely determined by $G(D)$ and are called the invariant factors of $G(D)$.

Consider a rational encoding matrix $G(D)$ with invariant-factor decomposition $G(D) = A(D)\Gamma(D)B(D)$ and let $G'(D)$ be an encoding matrix consisting of the first b rows of $B(D)$. Then

$$G(D) = A(D) \begin{pmatrix} \frac{\alpha_1(D)}{\beta_1(D)} & & & \\ & \frac{\alpha_2(D)}{\beta_2(D)} & & \\ & & \ddots & \\ & & & \frac{\alpha_b(D)}{\beta_b(D)} \end{pmatrix} G'(D), \tag{31}$$

where $\alpha_i(D)$ and $\beta_i(D)$ are polynomials satisfying $\alpha_i(D) \mid \alpha_{i+1}(D)$ and $\beta_{i+1}(D) \mid \beta_i(D)$, $i = 1, 2, \cdots, b-1$. Since both $A(D)$ and

$$\begin{pmatrix} \frac{\alpha_1(D)}{\beta_1(D)} & & & \\ & \frac{\alpha_2(D)}{\beta_2(D)} & & \\ & & \ddots & \\ & & & \frac{\alpha_b(D)}{\beta_b(D)} \end{pmatrix}$$

are nonsingular matrices over $\mathbf{F}_2(D)$ it follows from Theorem 3 that $G(D)$ and $G'(D)$ are equivalent. But $G'(D)$ is polynomial and since $B(D)$ has a polynomial inverse, it follows that $G'(D)$ has a polynomial right inverse (consisting of the first b columns of $B^{-1}(D)$). Therefore, $G'(D)$ is basic and we have the following algorithm.

An Algorithm to Construct a Basic Encoding Matrix Equivalent to a Given Encoding Matrix:

1) Compute the invariant-factor decomposition $G(D) = A(D)\Gamma(D)B(D)$.
2) Let $G'(D)$ be the first b rows of $B(D)$. $G'(D)$ is a basic encoding matrix equivalent to $G(D)$.

We summarize these results in Theorem 4.

Theorem 4: (Forney [1]) Every rational encoding matrix is equivalent to a basic convolutional encoding matrix.

Now, we have the following.

Theorem 5: (Forney [1]) Two basic convolutional encoding matrices $G(D)$ and $G'(D)$ are equivalent if and only if $G'(D) = T(D)G(D)$, where $T(D)$ is a $b \times b$ polynomial matrix with determinant 1.

Proof: Let $G'(D) = T(D)G(D)$, where $T(D)$ is a polynomial matrix with determinant 1. By Theorem 3, $G(D)$ and $G'(D)$ are equivalent.

Conversely, suppose that $G'(D)$ and $G(D)$ are equivalent. By Theorem 3, there is a nonsingular $b \times b$ matrix $T(D)$ over $\mathbf{F}_2(D)$ such that $G'(D) = T(D)G(D)$. Since $G(D)$ is basic it has a polynomial right inverse $G^{-1}(D)$. Then, $T(D) = G'(D)G^{-1}(D)$ is polynomial. We can repeat the argument with $G(D)$ and $G'(D)$ reversed to obtain $G(D) = S(D)G'(D)$ for some polynomial matrix $S(D)$. Thus, $G(D) = S(D)T(D)G(D)$. Since $G(D)$ is of full rank, we conclude that $S(D)T(D) = I_b$. Finally, since both $T(D)$ and $S(D)$ are polynomial, $T(D)$ must have determinant 1 and the proof is completed. □

Let $G(D) = A(D)\Gamma(D)B(D)$ be an invariant-factor decomposition of a basic encoding matrix $G(D)$. Then $G(D) = A(D)G'(D)$ where $G'(D)$ is the $b \times c$ polynomial matrix that consists of the first b rows of the matrix $B(D)$. Since the matrix $A(D)$ is a $b \times b$ unimodular matrix it follows from Theorem 5 that $G(D)$ and $G'(D)$ are equivalent basic encoding matrices.

V. Minimal-Basic Encoding Matrices

We shall now show that among all equivalent encoding matrices there exists a basic encoding matrix whose controller canonical form requires a minimal number of memory elements.

First, we shall consider only basic encoding matrices. The following definition is equivalent to Forney's definition of a minimal encoder [1].

Definition: A *minimal-basic* encoding matrix is a basic encoding matrix whose overall constraint length ν is minimal over all equivalent basic encoding matrices.

Let $G(D)$ be a basic encoding matrix. The positions for the row-wise highest order coefficients in $G(D)$ will play a significant role in the sequel. Hence, we let $[G(D)]_h$ be a $(0,1)$-matrix with 1 in the position (i,j) where $\deg g_{ij}(D) = \nu_i$ and 0, otherwise.

Let us write

$$G(D) = G_0(D) + G_1(D), \tag{32}$$

where

$$G_1(D) = \begin{pmatrix} D^{\nu_1} & & & \\ & D^{\nu_2} & & \\ & & \ddots & \\ & & & D^{\nu_b} \end{pmatrix} [G(D)]_h. \tag{33}$$

Then, all entries in the ith row of $G_0(D)$ are of degree $< \nu_i$. Clearly the maximum degree μ among the $b \times b$ subdeterminants of $G(D)$ is $\leq \nu$.

Theorem 6: Let $G(D)$ be a $b \times c$ basic encoding matrix with overall constraint length ν. Then, the following statements are equivalent.

a) $G(D)$ is a minimal-basic encoding matrix.
b) The maximum degree μ among the $b \times b$ subdeterminants of $G(D)$ is equal to the overall constraint length ν.
c) $[G(D)]_h$ is of full rank.

Proof: It follows immediately from (33) that b) and c) are equivalent. Thus we need only prove that a) and b) are equivalent.

a)⇒b): Assume that $G(D)$ is minimal-basic.

Suppose that $\mu < \nu$, i.e., rank $[G(D)]_h < b$. Denote the rows of $G(D)$ by $\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_b$ and the rows of $[G(D)]_h$ by $[\boldsymbol{r}_1], [\boldsymbol{r}_2], \cdots, [\boldsymbol{r}_b]$. Then there is a linear relation

$$[\boldsymbol{r}_{i_1}] + [\boldsymbol{r}_{i_2}] + \cdots + [\boldsymbol{r}_{i_d}] = \boldsymbol{0}. \tag{34}$$

The ith row of $G_1(D)$ is $D^{\nu_i}[\boldsymbol{r}_i]$. Without loss of generality we can assume that $\nu_{i_d} \geq \nu_{i_j}$, $j = 1, 2, \cdots, d-1$. Adding

$$\begin{aligned} &D^{\nu_{i_d}-\nu_{i_1}} D^{\nu_{i_1}}[\boldsymbol{r}_{i_1}] + D^{\nu_{i_d}-\nu_{i_2}} D^{\nu_{i_2}}[\boldsymbol{r}_{i_2}] + \cdots \\ &\quad + D^{\nu_{i_d}-\nu_{i_{d-1}}} D^{\nu_{i_{d-1}}}[\boldsymbol{r}_{i_{d-1}}] \\ &\qquad = D^{\nu_{i_d}}([\boldsymbol{r}_{i_1}] + [\boldsymbol{r}_{i_2}] + \cdots + [\boldsymbol{r}_{i_{d-1}}]) \end{aligned} \tag{35}$$

to the i_dth row of $G_1(D)$ reduces it to an all zero row. Similarly, adding

$$D^{\nu_{i_d}-\nu_{i_1}} \boldsymbol{r}_1 + D^{\nu_{i_d}-\nu_{i_2}} \boldsymbol{r}_2 + \cdots + D^{\nu_{i_d}-\nu_{i_{d-1}}} \boldsymbol{r}_{i_{d-1}} \tag{36}$$

to the i_dth row of $G(D)$ will reduce the highest degree of the i_dth row of $G(D)$ but leave the other rows of $G(D)$ unchanged. Thus we obtain a basic encoding matrix equivalent to $G(D)$ with an overall constraint length which is less than that of $G(D)$. This is a contradiction to the assumption that $G(D)$ is minimal-basic and we conclude that $\mu = \nu$.

b)⇒a): Assume that $\mu = \nu$.

Let $G'(D)$ be a basic encoding matrix equivalent to $G(D)$. From Theorem 5 follows that $G'(D) = T(D)G(D)$, where $T(D)$ is a $b \times b$ polynomial matrix with determinant 1. Since $\det T(D) = 1$, the maximum degree among the $b \times b$ subdeterminants of $G'(D)$ is equal to that of $G(D)$. Hence, μ is invariant over all equivalent basic encoding matrices.

Clearly μ is less than or equal to the overall constraint length for all equivalent basic encoding matrices and it follows that $G(D)$ is a minimal-basic encoding matrix. □

Corollary 7: Let $G(D)$ be a $b \times c$ basic encoding matrix with maximum degree μ among its $b \times b$ subdeterminants. Then $G(D)$ has an equivalent minimal-basic encoding matrix whose overall constraint length $\nu = \mu$.

Proof: Follows from the proof of Theorem 6 and the fact that μ is invariant over all equivalent basic encoding matrices. □

Example 3: Consider the encoding matrix for the encoder in Fig. 7, viz.,

$$G'(D) = \begin{pmatrix} 1+D & D & 1 \\ 1+D^2+D^3 & 1+D+D^2+D^3 & 0 \end{pmatrix}.$$

The rank of

$$[G'(D)]_h = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

is one. Hence, $G'(D)$ cannot be a minimal-basic encoding matrix.

On the other hand, $G'(D)$ has the following three $b \times b$ subdeterminants:

$$1+D+D^3, 1+D^2+D^3, 1+D+D^2+D^3$$

and, thus, $\mu = 3$. Hence, any minimal-basic encoding matrix equivalent to $G'(D)$ has overall constraint length $\nu = 3$.

The equivalent basic encoding matrix for the encoder in Fig. 6

$$G(D) = \begin{pmatrix} 1+D & D & 1 \\ D^2 & 1 & 1+D+D^2 \end{pmatrix}$$

has

$$[G(D)]_h = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

with full rank and, hence, is a minimal-basic encoding matrix.

Clearly we can use the technique in the proof of Theorem 6 to obtain a minimal-basic encoding matrix equivalent to the basic encoding matrix $G'(D)$ for the encoder in Fig. 7. We simply multiply the first row of $G'(D)$ by $D^{\nu_2-\nu_1} = D^2$ and add it to the second row:

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

Thus, the minimal-basic encoding matrix equivalent to a given basic encoding matrix is not necessarily unique.

In general, we have [1] the next algorithm.

A Simple Algorithm to Construct a Minimal-Basic Encoding Matrix Equivalent to a Given Basic Encoding Matrix:

1) IF $[G(D)]_h$ has full rank, THEN $G(D)$ is a minimal-basic encoding matrix and we STOP; ELSE GO TO next step.
2) Let $[\boldsymbol{r}_{i_1}], [\boldsymbol{r}_{i_2}], \cdots, [\boldsymbol{r}_{i_d}]$ denote a set of rows of $[G(D)]_h$ such that $\nu_{i_d} \geq \nu_{i_j}$, $1 \leq j < d$, and

$$[\boldsymbol{r}_{i_1}] + [\boldsymbol{r}_{i_2}] + \cdots + [\boldsymbol{r}_{i_d}] = \boldsymbol{0}.$$

Let $\boldsymbol{r}_{i_1}, \boldsymbol{r}_{i_2}, \cdots, \boldsymbol{r}_{i_d}$ denote the corresponding set of rows of $G(D)$. Add

$$D^{\nu_{i_d}-\nu_{i_1}} \boldsymbol{r}_1 + D^{\nu_{i_d}-\nu_{i_2}} \boldsymbol{r}_2 + \cdots + D^{\nu_{i_d}-\nu_{i_{d-1}}} \boldsymbol{r}_{i_{d-1}}$$

to the i_dth row of $G(D)$.

Call the new matrix $G(D)$ and GO TO step 1.

Corollary 8 (Forney [1]): Every encoding matrix is equivalent to a minimal-basic encoding matrix.

The physical state space of a controller canonical form of an encoding matrix of overall constraint length ν contains 2^ν states. This type of realization plays an important role in connection with minimal-basic encoding matrices as we shall see in the sequel, but first we prove a technical lemma.

Lemma 9: Let $G(D)$ be a minimal-basic encoding matrix and let

$$\boldsymbol{u}(D) = \sum_{i=-m}^{n} (u_i^{(1)} u_i^{(2)} \cdots u_i^{(b)}) D^i, \tag{37}$$

where m is the memory of $G(D)$ and $n \geq -m$. If $\boldsymbol{u}(D)G(D)Q = \mathbf{0}$, then $\boldsymbol{u}(D) = \mathbf{0}$.

Proof: Let

$$\boldsymbol{v}(D) = \boldsymbol{u}(D)G(D). \tag{38}$$

Then, by assumption,

$$\boldsymbol{v}(D)Q = \boldsymbol{u}(D)G(D)Q = \mathbf{0}. \tag{39}$$

Thus each coefficient of D^i, $i \geq 0$, in $\boldsymbol{v}(D)$ must be $\mathbf{0}$. Write $G(D)$ as in (32), then we have

$$\boldsymbol{v}(D) = \boldsymbol{u}(D)G_0(D) + \boldsymbol{u}(D)\begin{pmatrix} D^{\nu_1} & & & \\ & D^{\nu_2} & & \\ & & \ddots & \\ & & & D^{\nu_b} \end{pmatrix}[G(D)]_h. \tag{40}$$

Without loss of generality, we can assume that

$$m = \nu_1 = \nu_2 = \cdots = \nu_l > \nu_{l+1} \geq \cdots \geq \nu_b. \tag{41}$$

Then the coefficient of D^{m+n}, where $m + n \geq 0$, in $\boldsymbol{v}(D)$ is

$$(u_n^{(1)}\ u_n^{(2)}\ \cdots\ u_n^{(l)}\ 0\ \cdots\ 0)[G(D)]_h,$$

which must be $\mathbf{0}$. Since $G(D)$ is a minimal-basic encoding matrix, $[G(D)]_h$ is of rank b. Hence, $u_n^{(1)} = u_n^{(2)} = \cdots = u_n^{(l)} = 0$. Proceeding in this way, we can prove that $\boldsymbol{u}(D) = \mathbf{0}$. □

(Lemma 9 also follows from the *predictable degree property* introduced in [1].)

Theorem 10: Let $G(D)$ be a minimal-basic encoding matrix whose overall constraint length is ν. Then[2]

$$\#\{\text{abstract states}\} = 2^\nu. \tag{42}$$

Proof: Consider the controller canonical form of the minimal-basic encoding matrix $G(D)$. Clearly input sequences of the form

$$\boldsymbol{u}(D) = \left(\sum_{i=1}^{\nu_1} u_{-i}^{(1)} D^{-i} \quad \sum_{i=1}^{\nu_2} u_{-i}^{(2)} D^{-i} \quad \cdots \quad \sum_{i=1}^{\nu_b} u_{-i}^{(b)} D^{-i}\right) \tag{43}$$

will carry us to all physical states at time 0. Then we have the abstract states

$$\boldsymbol{s}(D) = \boldsymbol{u}(D)G(D)Q, \tag{44}$$

where $\boldsymbol{u}(D)$ is of the form given in (43). Every abstract state can be obtained in this way and we have

$$\#\{\text{abstract states}\} \leq 2^\nu. \tag{45}$$

To prove that the equality sign holds, it is enough to show that $\boldsymbol{u}(D) = \mathbf{0}$ is the only physical state that produces the abstract state $\boldsymbol{s}(D) = \mathbf{0}$. This follows from Lemma 9. □

We shall conclude this section by proving that the constraint lengths are invariants of equivalent minimal-basic encoding matrices. First we need the following

Lemma 11: Let V be a k-dimensional vector space over a field F and let $\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots, \boldsymbol{\alpha}_k\}$ be a basis of V. Let $\{\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots, \boldsymbol{\beta}_l\}$ be a set of l, $l < k$, linearly independent vectors of V. Then there exist $k - l$ vectors $\boldsymbol{\alpha}_{i_{l+1}}, \boldsymbol{\alpha}_{i_{l+2}}, \cdots, \boldsymbol{\alpha}_{i_k}$, $1 \leq i_{l+1} < i_{l+2} < \cdots < i_k \leq k$, such that $\{\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots, \boldsymbol{\beta}_l, \boldsymbol{\alpha}_{i_{l+1}}, \boldsymbol{\alpha}_{i_{l+2}}, \cdots, \boldsymbol{\alpha}_{i_k}\}$ is also a basis of V.

Proof: Consider the vectors in the sequence $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots, \boldsymbol{\beta}_l, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots, \boldsymbol{\alpha}_k$ one by one successively from left to right. If the vector under consideration is a linear combination of vectors to the left of it, then delete it; otherwise keep it. Finally, we obtain a basis $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots, \boldsymbol{\beta}_l, \boldsymbol{\alpha}_{i_{l+1}}, \boldsymbol{\alpha}_{i_{l+2}}, \cdots, \boldsymbol{\alpha}_{i_k}$, $1 \leq i_{l+1} < \cdots < i_k \leq k$, of V. □

Theorem 12: The constraint lengths of two equivalent minimal-basic encoding matrices are equal one by one up to a rearrangement.

Proof: Let $G(D)$ and $G'(D)$ be two equivalent minimal-basic encoding matrices with constraint lengths $\nu_1, \nu_2, \cdots, \nu_b$ and $\nu_1', \nu_2', \cdots, \nu_b'$, respectively. Without loss of generality, we assume that $\nu_1 \leq \nu_2 \leq \cdots \leq \nu_b$ and $\nu_1' \leq \nu_2' \leq \cdots \leq \nu_b'$.

Now suppose that ν_i and ν_i' are not equal for all i, $1 \leq i \leq b$. Let j be the smallest index such that $\nu_j \neq \nu_j'$. Then, without loss of generality, we assume that $\nu_j < \nu_j'$. From the sequence $\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_j, \boldsymbol{g}_1', \boldsymbol{g}_2', \cdots, \boldsymbol{g}_b'$ we can according to Lemma 11, obtain a basis $\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_j, \boldsymbol{g}_{i_{j+1}}', \boldsymbol{g}_{i_{j+2}}', \cdots, \boldsymbol{g}_{i_b}'$ of $\mathcal{C}$. These b row vectors form an encoding matrix $G''(D)$ which is equivalent to $G'(D)$. Let

$$\{g_1', g_2', \cdots, g_b'\} \setminus \{g_{i_{j+1}}', g_{i_{j+2}}', \cdots, g_{i_b}'\} = \{g_{i_1}', g_{i_2}', \cdots, g_{i_j}'\}. \tag{46}$$

From our assumptions, it follows that

$$\sum_{l=1}^{j} \nu_l < \sum_{l=1}^{j} \nu_l' \leq \sum_{l=1}^{j} \nu_{i_l}'. \tag{47}$$

Then we have

$$\nu'' = \sum_{l=1}^{j} \nu_l + \sum_{l=j+1}^{b} \nu_{i_l}' < \sum_{l=1}^{j} \nu_{i_l}' + \sum_{l=j+1}^{b} \nu_{i_l}' = \nu', \tag{48}$$

where ν' and ν'' are the overall constraint lengths of the encoding matrices $G'(D)$ and $G''(D)$, respectively.

[2] $\#\{\cdot\}$ denotes the cardinality of the set $\{\cdot\}$.

From Theorem 3, it follows that there exists a $b \times b$ nonsingular matrix $T(D)$ over $\mathbf{F}_2(D)$ such that

$$G''(D) = T(D)G'(D). \tag{49}$$

Since $G'(D)$ is basic it has a polynomial right inverse $G'^{-1}(D)$ and it follows that

$$T(D) = G''(D)G'^{-1}(D) \tag{50}$$

is polynomial. Denote by μ' and μ'' the maximum degrees among the $b \times b$ minors of $G'(D)$ and $G''(D)$, respectively. It follows from (49) that

$$\mu'' = \deg \mid T(D) \mid + \mu'. \tag{51}$$

Clearly $\nu'' \geq \mu''$ and, since $G'(D)$ is minimal-basic, $\nu' = \mu'$ by Theorem 6. Thus,

$$\nu'' \geq \deg \mid T(D) \mid + \nu' \geq \nu', \tag{52}$$

which contradicts (48) and the proof is completed. □

Corollary 13: Two equivalent minimal-basic encoding matrices have the same memory.

The statement in Theorem 12 is equivalent to a classical result of Kronecker [6]; see also Forney [7].

VI. Minimal Encoding Matrices

We shall now proceed to show that a minimal-basic encoding matrix is also minimal in a more general sense.

Definition: A convolutional encoding matrix is *minimal* if its number of abstract states is minimal over all equivalent encoding matrices.

Before we can show that every minimal-basic encoding matrix is also a (basic) minimal encoding matrix we have to prove the following lemmas.

Lemma 14 (Forney [1], [3]): Only the zero abstract state of a minimal-basic encoding matrix $G(D)$ can be a codeword.

Proof: We can assume that the abstract state $\boldsymbol{s}(D)$ arises from an input $\boldsymbol{u}(D)$ which is polynomial in D^{-1} and of degree $\leq m$ and without a constant term, i.e., $\boldsymbol{u}(0) = \mathbf{0}$. Thus,

$$\boldsymbol{s}(D) = \boldsymbol{u}(D)G(D)Q. \tag{53}$$

Then, it follows that

$$\boldsymbol{u}(D)G(D) = \boldsymbol{w}(D) + \boldsymbol{s}(D), \tag{54}$$

where $\boldsymbol{w}(D)$ is polynomial in D^{-1} without a constant term.

Assume that $\boldsymbol{s}(D)$ is a codeword, i.e., there is an input $\boldsymbol{u}'(D) \in \mathbf{F}_2((D))$ such that

$$\boldsymbol{s}(D) = \boldsymbol{u}'(D)G(D). \tag{55}$$

Since $\boldsymbol{s}(D)$ is polynomial and $G(D)$ has a polynomial inverse it follows that $\boldsymbol{u}'(D) \in \mathbf{F}_2[D]$.

Combining (54) and (55) we have

$$(\boldsymbol{u}(D) + \boldsymbol{u}'(D))G(D) = \boldsymbol{w}(D). \tag{56}$$

Consequently

$$(\boldsymbol{u}(D) + \boldsymbol{u}'(D))G(D)Q = \mathbf{0}. \tag{57}$$

By Lemma 9,

$$\boldsymbol{u}(D) + \boldsymbol{u}'(D) = \mathbf{0} \tag{58}$$

and, hence, $\boldsymbol{u}'(D) = \mathbf{0}$. It follows from (55) that $\boldsymbol{s}(D) = \mathbf{0}$. □

Lemma 15: Let $G(D)$ and $G'(D)$ be equivalent encoding matrices. Then, every abstract state of $G(D)$ can be expressed as a sum of an abstract state of $G'(D)$ and a codeword. Furthermore, if $G'(D)$ is minimal-basic, then the expression is unique.

Proof: Assume that $G(D) = T(D)G'(D)$, where $T(D)$ is a $b \times b$ nonsingular matrix over $\mathbf{F}_2(D)$. Any abstract state of $G(D)$, $\boldsymbol{s}_G(D)$, can be written in the form $\boldsymbol{u}(D)G(D)Q$, where $\boldsymbol{u}(D)$ is polynomial in D^{-1} without a constant term. Thus, we have

$$\begin{aligned}\boldsymbol{s}_G(D) &= \boldsymbol{u}(D)G(D)Q = \boldsymbol{u}(D)T(D)G'(D)Q \\ &= \boldsymbol{u}(D)T(D)(P+Q)G'(D)Q \\ &= \boldsymbol{u}(D)T(D)PG'(D)Q + \boldsymbol{u}(D)T(D)QG'(D)Q.\end{aligned} \tag{59}$$

Since $\boldsymbol{u}(D)T(D)P$ is polynomial in D^{-1} without a constant term it follows from (25) that

$$\boldsymbol{s}_{G'}(D) = \boldsymbol{u}(D)T(D)PG'(D)Q \tag{60}$$

is an abstract state of $G'(D)$. Furthermore, $\boldsymbol{u}(D)T(D)Q$ is a formal power series, and so is $\boldsymbol{u}(D)T(D)QG'(D)$. Hence,

$$\begin{aligned}\boldsymbol{v}(D) &\stackrel{\text{def}}{=} \boldsymbol{u}(D)T(D)QG'(D)Q \\ &= \boldsymbol{u}(D)T(D)QG'(D)\end{aligned} \tag{61}$$

is a codeword encoded by $G'(D)$. Combining (59), (60), and (61) we obtain

$$\boldsymbol{s}_G(D) = \boldsymbol{s}_{G'}(D) + \boldsymbol{v}(D) \tag{62}$$

and we have proved that every abstract state of $G(D)$ can be written as a sum of an abstract state of $G'(D)$ and a codeword.

Assume now that $G'(D) = G_{mb}(D)$ is minimal-basic. To prove uniqueness, we assume that

$$\boldsymbol{s}_G(D) = \boldsymbol{s}_{mb}(D) + \boldsymbol{v}(D) = \boldsymbol{s}'_{mb}(D) + \boldsymbol{v}'(D), \tag{63}$$

where $\boldsymbol{s}_{mb}(D), \boldsymbol{s}'_{mb}(D)$ are abstract states of $G_{mb}(D)$, and $\boldsymbol{v}(D), \boldsymbol{v}'(D)$ are codewords. Since the sum of two abstract states is an abstract state and the sum of two codewords is a codeword it follows from (63) that

$$\boldsymbol{s}''_{mb}(D) = \boldsymbol{s}_{mb}(D) + \boldsymbol{s}'_{mb}(D) = \boldsymbol{v}(D) + \boldsymbol{v}'(D) = \boldsymbol{v}''(D) \tag{64}$$

is both an abstract state of $G_{mb}(D)$ and a codeword. From Lemma 11 we deduce that

$$\boldsymbol{s}''_{mb}(D) = \mathbf{0} \tag{65}$$

and, hence, that

$$\boldsymbol{s}_{mb}(D) = \boldsymbol{s}'_{mb}(D), \tag{66}$$

and

$$\boldsymbol{v}(D) = \boldsymbol{v}'(D), \tag{67}$$

which completes the proof. □

Theorem 16 (Forney [1]): Let $G(D)$ be any encoding matrix equivalent to a minimal-basic encoding matrix $G_{mb}(D)$. Then

$$\#\{\text{abstract states of } G(D)\} \geq \#\{\text{abstract states of } G_{mb}(D)\}. \tag{68}$$

Proof: Consider the following map:

$$\phi : \{\text{abstract states of } G(D)\} \rightarrow \{\text{abstract states of } G_{mb}(D)\}$$
$$\boldsymbol{s}_G(D) \mapsto \boldsymbol{s}_{mb}(D),$$

where

$$\boldsymbol{s}_G(D) = \boldsymbol{s}_{mb}(D) + \boldsymbol{v}(D), \tag{69}$$

in which $\boldsymbol{v}(D)$ is a codeword. From Lemma 15, it follows that ϕ is well-defined.

By the first statement of Lemma 15 we can prove that every abstract state $\boldsymbol{s}_{mb}(D)$ can be written as a sum of an abstract state of $G(D)$ and a codeword. Hence, we conclude that ϕ is surjective which proves the theorem. □

Remark: The map ϕ in Theorem 16 is clearly linear. Moreover, if $G(D)$ is a minimal encoding matrix, then ϕ is necessarily an isomorphism of abstract state space of $G(D)$ and that of $G_{mb}(D)$.

From Theorem 16, Corollary 17 follows immediately.

Corollary 17: Every minimal-basic encoding matrix is a (basic) minimal encoding matrix.

Next, we shall prove the following little lemma.

Lemma 18: Let $G(D)$ be a $b \times c$ matrix of rank b whose entries are rational functions of D. Then a necessary and sufficient condition for $G(D)$ to have a polynomial inverse is: for each $\boldsymbol{u}(D) \in \mathbb{F}_2^b(D)$ satisfying $\boldsymbol{u}(D)G(D) \in \mathbb{F}_2^c[D]$ we must have $\boldsymbol{u}(D) \in \mathbb{F}_2^b[D]$.

Proof: Since the necessity of the condition is obvious we shall prove only the sufficiency. Let us assume that $G(D)$ does not have a polynomial inverse. Then, from the invariant-factor decomposition

$$G(D) = A(D)\begin{pmatrix} \frac{\alpha_1(D)}{\beta_1(D)} & & & & \\ & \frac{\alpha_2(D)}{\beta_2(D)} & & & \\ & & \ddots & & \\ & & & \frac{\alpha_b(D)}{\beta_b(D)} & 0 \cdots 0 \end{pmatrix} B(D) \tag{70}$$

follows that $\alpha_b(D) \neq 1$. Clearly,

$$\boldsymbol{u}(D) = \left(0 \cdots 0 \, \frac{\beta_b(D)}{\alpha_b(D)}\right) A^{-1}(D) \notin \mathbb{F}_2^b[D] \tag{71}$$

but

$$\boldsymbol{u}(D)G(D) = \left(0 \cdots 0 \, \frac{\beta_b(D)}{\alpha_b(D)}\right) A^{-1}(D)G(D) = (0 \cdots 0\, 1\, 0 \cdots 0)B(D) \in \mathbb{F}_2^c[D]. \tag{72}$$

Hence, we have proved our lemma. □

We are now well prepared to prove the following theorem on minimal encoding matrices.

Theorem 19 (cf. [1]): Let $G(D)$ be an encoding matrix and $G_{mb}(D)$ be an equivalent minimal-basic encoding matrix. Then, the following statements are equivalent.

a) $G(D)$ is a minimal encoding matrix.
b) # {abstract states of $G(D)$} = # {abstract states of $G_{mb}(D)$}.
c) Only the zero abstract state of $G(D)$ can be a codeword.
d) $G(D)$ has a polynomial right inverse in D and a polynomial right inverse in D^{-1}.

Proof: It follows immediately from Theorem 16 that a) and b) are equivalent.

Next, we prove that b) and c) are equivalent. In the proof of Theorem 16, we have defined a surjective map

$$\phi : \{\text{abstract states of } G(D)\} \rightarrow \{\text{abstract states of } G_{mb}(D)\}$$
$$\boldsymbol{s}_G(D) \mapsto \boldsymbol{s}_{mb}(D),$$

where

$$\boldsymbol{s}_G(D) = \boldsymbol{s}_{mb}(D) + \boldsymbol{v}(D), \tag{73}$$

in which $\boldsymbol{v}(D)$ is a codeword. Clearly, ϕ is injective, if and only if b) holds, and if and only if c) holds. Hence, b) and c) are equivalent.

It remains to prove that c) and d) are equivalent.

c)⇒d): Suppose that c) holds. First we shall prove that $G(D)$ has a polynomial right inverse in D^{-1}. Let $\boldsymbol{u}(D) \in \mathbb{F}_2^b(D)$ and assume that $\boldsymbol{v}(D) = \boldsymbol{u}(D)G(D)$ is polynomial in D^{-1}. Then $D^{-1}\boldsymbol{v}(D)$ is polynomial in D^{-1} without a constant term, i.e.,

$$D^{-1}\boldsymbol{v}(D)Q = \boldsymbol{0}. \tag{74}$$

But

$$\begin{aligned} D^{-1}\boldsymbol{v}(D)Q &= D^{-1}\boldsymbol{u}(D)(P+Q)G(D)Q \\ &= D^{-1}\boldsymbol{u}(D)PG(D)Q + D^{-1}\boldsymbol{u}(D)QG(D)Q \\ &= \boldsymbol{0}, \end{aligned} \tag{75}$$

where

$$D^{-1}\boldsymbol{u}(D)QG(D)Q = D^{-1}\boldsymbol{u}(D)QG(D) \tag{76}$$

is a codeword. Hence, from (75) and (76) it follows that the abstract state $D^{-1}\boldsymbol{u}(D)PG(D)Q$ is a codeword and, then, since c) holds, it is the zero codeword. Thus,

$$D^{-1}\boldsymbol{u}(D)QG(D) = \boldsymbol{0} \tag{77}$$

and, since $G(D)$ has full rank,

$$D^{-1}\boldsymbol{u}(D)Q = \boldsymbol{0} \tag{78}$$

or, in other words, $\boldsymbol{u}(D)$ is polynomial in D^{-1}. Since every rational function in D can be written as a rational function in D^{-1}, $G(D)$ can be written as a matrix whose entries are rational functions in D^{-1}. We can apply Lemma 18 and conclude that $G(D)$ has a polynomial right inverse in D^{-1}.

Next, we shall prove that $G(D)$ has a polynomial right pseudo-inverse in D. (By a polynomial right pseudo-inverse of $G(D)$ we mean a $c \times b$ polynomial matrix $\widetilde{G^{-1}}(D)$ such that $G(D)\widetilde{G^{-1}}(D) = D^s I_b$ for some $s \geq 0$.) Let $G_{-1}^{-1}(D)$ be a polynomial right inverse in D^{-1} of $G(D)$. Then there exists an integer $s \geq 0$ such that $D^s G_{-1}^{-1}(D)$ is a polynomial matrix in D and

$$G(D)D^s G_{-1}^{-1}(D) = D^s I_b, \tag{79}$$

i.e., $D^s G_{-1}^{-1}(D)$ is a polynomial right pseudo-inverse in D of $G(D)$.

Finally, we shall prove that $G(D)$ has also a polynomial right inverse in D. Let $\boldsymbol{u}(D) \in \mathbb{F}_2^b(D)$ and assume that $\boldsymbol{v}(D) = \boldsymbol{u}(D)G(D)$ is polynomial in D. Then,

$$\boldsymbol{v}(D) = \boldsymbol{u}(D)PG(D) + \boldsymbol{u}(D)QG(D), \tag{80}$$

where $\boldsymbol{u}(D)Q$ is a formal power series. Thus, $\boldsymbol{u}(D)QG(D)$ is also a formal power series and, since $\boldsymbol{v}(D)$ is polynomial in D, it follows that $\boldsymbol{u}(D)PG(D)$ is a formal power series. Then

$$\boldsymbol{u}(D)PG(D) = \boldsymbol{u}(D)PG(D)Q \tag{81}$$

is an abstract state. From (80), it follows that it is also a codeword and, since c) holds, we conclude that

$$\boldsymbol{u}(D)PG(D) = \boldsymbol{0}. \tag{82}$$

Since $G(D)$ has full rank,

$$\boldsymbol{u}(D)P = \boldsymbol{0} \tag{83}$$

or, in other words, $\boldsymbol{u}(D)$ is a formal power series.

Since $\boldsymbol{v}(D)$ is polynomial and $D^s G_{-1}^{-1}(D)$ is a polynomial matrix in D it follows that

$$\begin{aligned}\boldsymbol{v}(D)D^s G_{-1}^{-1}(D) &= \boldsymbol{u}(D)G(D)D^s G_{-1}^{-1}(D) \\ &= \boldsymbol{u}(D)D^s\end{aligned} \tag{84}$$

is polynomial, i.e., $\boldsymbol{u}(D)$ has finitely many terms. But $\boldsymbol{u}(D)$ is a formal power series, hence, we conclude that it is polynomial in D. By Lemma 18, $G(D)$ has a polynomial right inverse in D.

d)⇒c): Assume that the abstract state $\boldsymbol{s}_G(D)$ of $G(D)$ is a codeword. That is,

$$\begin{aligned}\boldsymbol{s}_G(D) &= \boldsymbol{u}(D)G(D)Q \\ &= \boldsymbol{u}'(D)G(D),\end{aligned} \tag{85}$$

where $\boldsymbol{u}(D)$ is polynomial in D^{-1} but without a constant term and $\boldsymbol{u}'(D) \in \mathbb{F}_2^b((D))$. Since $\boldsymbol{s}_G(D)$ is a formal power series and $G(D)$ has a polynomial right inverse, it follows that $\boldsymbol{u}'(D)$ is also a formal power series.

Let us use the fact that

$$Q = 1 + P \tag{86}$$

and rewrite (85) as follows:

$$\begin{aligned}\boldsymbol{s}_G(D) &= \boldsymbol{u}(D)G(D) + \boldsymbol{u}(D)G(D)P \\ &= \boldsymbol{u}'(D)G(D).\end{aligned} \tag{87}$$

Let $G_{-1}^{-1}(D)$ be a right inverse of $G(D)$ whose entries are polynomials in D^{-1}. Then

$$\begin{aligned}&\boldsymbol{u}(D)G(D)G_{-1}^{-1}(D) + \boldsymbol{u}(D)G(D)PG_{-1}^{-1}(D) \\ &\quad = \boldsymbol{u}'(D)G(D)G_{-1}^{-1}(D),\end{aligned} \tag{88}$$

which can be simplified to

$$\boldsymbol{u}(D) + \boldsymbol{u}(D)G(D)PG_{-1}^{-1}(D) = \boldsymbol{u}'(D). \tag{89}$$

Since $\boldsymbol{u}(D)G(D)P$ is polynomial in D^{-1} without a constant term, it follows that $\boldsymbol{u}(D)G(D)PG_{-1}^{-1}(D)$ is polynomial in D^{-1} without a constant term. Furthermore, $\boldsymbol{u}(D)$ is polynomial in D^{-1} without a constant term and $\boldsymbol{u}'(D)$ is a formal power series. Thus, we conclude that $\boldsymbol{u}'(D) = \boldsymbol{0}$ and, hence, that $\boldsymbol{s}_G(D) = \boldsymbol{0}$. □

The simple minimality test d) is a new result related to Forney's *global invertability test* [7] and to the minimality test of Loeliger and Mittelholzer [8].

Corollary 20: Every minimal encoding matrix is noncatastrophic.

Proof: It was pointed out by Forney [1] that a convolutional encoding matrix is non-catastrophic if and only if it has a polynomial right pseudo-inverse. Hence, our corollary follows immediately from Theorem 19. □

The following simple example shows that not all basic encoding matrices are minimal.

Example 4: Consider the basic encoding matrix

$$G(D) = \begin{pmatrix} 1+D & D \\ D & 1+D \end{pmatrix}, \tag{90}$$

which has $\mu = 0$ but $\nu = 2$. Clearly, it is not minimal-basic.

The *equivalent* minimal-basic encoding matrix,

$$G_{mb}(D) = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \tag{91}$$

has only one abstract state, viz., $\boldsymbol{s}_{mb} = (0,0)$, and can, of course, be realized without any memory element.

Since $G(D)$ has two abstract states, viz., $\boldsymbol{s}_0 = (0,0)$ and $\boldsymbol{s}_1 = (1,1)$, it is not minimal!

Moreover, $G(D)$ is invertible and its unique inverse is $G(D)$ itself. But

$$\begin{aligned}G^{-1}(D) &= G(D) \\ &= \begin{pmatrix} 1+D & D \\ D & 1+D \end{pmatrix} \\ &= \begin{pmatrix} \frac{1+D^{-1}}{D^{-1}} & \frac{1}{D^{-1}} \\ \frac{1}{D^{-1}} & \frac{1+D^{-1}}{D^{-1}} \end{pmatrix},\end{aligned} \tag{92}$$

which is not a polynomial matrix in D^{-1}. By Theorem 16 we deduce again that $G(D)$ is not minimal.

Before we state a theorem on when a basic encoding matrix is minimal, we shall prove two lemmas.

Lemma 21: Let $f_1(D), f_2(D), \cdots, f_l(D) \in \mathbf{F}_2[D]$ with

$$(f_1(D), f_2(D), \cdots, f_l(D)) = 1, \tag{93}$$

where $(f_1(D), f_2(D), \cdots, f_l(D))$ denotes the greatest common divisor of $f_1(D), f_2(D), \cdots, f_l(D)$, and let

$$n = \max(\deg f_1(D), \deg f_2(D), \cdots, \deg f_l(D)). \tag{94}$$

Then for $m \geq n$ $D^{-m}f_1(D), D^{-m}f_2(D), \cdots, D^{-m}f_l(D) \in \mathbf{F}_2[D^{-1}]$ and

$$(D^{-m}f_1(D), D^{-m}f_2(D), \cdots, D^{-m}f_l(D)) = D^{-(m-n)}. \tag{95}$$

Proof: Let

$$f_i(D) = D^{s_i}g_i(D), \qquad i = 1, 2, \cdots, l, \tag{96}$$

where s_i is the start of $f_i(D)$ and $g_i(D) \in \mathbf{F}_2[D]$ is delayfree. From (93) follows

$$\min(s_1, s_2, \cdots, s_l) = 0 \tag{97}$$

and

$$(g_1(D), g_2(D), \cdots, g_l(D)) = 1. \tag{98}$$

For $m \geq n$

$$\begin{aligned} D^{-m}f_i(D) &= D^{-m}D^{s_i}g_i(D) \\ &= D^{-(m-s_i-\deg g_i(D))}\left(D^{-\deg g_i(D)}g_i(D)\right) \\ &= D^{-(m-\deg f_i(D))}\left(D^{-\deg g_i(D)}g_i(D)\right), \end{aligned}$$

$$i = 1, 2, \cdots, l, \tag{99}$$

where the last equality follows from the fact that

$$\deg f_i(D) = s_i + \deg g_i(D), \qquad i = 1, 2, \cdots, l. \tag{100}$$

Since $D^{-\deg g_i(D)}g_i(D)$, $i = 1, 2, \cdots, l$, are delayfree it follows from (99) that

$$\begin{aligned} &(D^{-m}f_1(D), D^{-m}f_2(D), \cdots, D^{-m}f_l(D)) \\ &\quad = D^{-(m-n)}(D^{-\deg g_1(D)}g_1(D), D^{-\deg g_2(D)}g_2(D), \cdots, \\ &\qquad\qquad D^{-\deg g_l(D)}g_l(D)). \end{aligned} \tag{101}$$

Clearly,

$$\begin{aligned} &\left(D^{-\deg g_1(D)}g_1(D), D^{-\deg g_2(D)}g_2(D),\right. \\ &\qquad \left. \cdots, D^{-\deg g_l(D)}g_l(D)\right) = 1 \end{aligned} \tag{102}$$

and the proof is completed. □

Lemma 22: Let $G(D)$ be a basic encoding matrix and let r and s be the maximum degree of its $b \times b$ minors and $(b-1) \times (b-1)$ minors, respectively. Then the b-th invariant factor of $G(D)$ regarded as a matrix over $\mathbf{F}_2(D^{-1})$ is $1/D^{-(r-s)}$.

Proof: Let $G(D) = (g_{ij}(D))$, $1 \leq i \leq b$, $1 \leq j \leq c$, and let $n = \max_{i,j}(\deg g_{ij}(D))$. Write $G(D)$ as a matrix over $\mathbf{F}_2(D^{-1})$ as follows:

$$G(D) = \left(\frac{D^{-n}g_{ij}(D)}{D^{-n}}\right)_{i,j} = \frac{1}{D^{-n}}G_{-1}(D), \tag{103}$$

where

$$G_{-1}(D) = \left(D^{-n}g_{ij}(D)\right)_{i,j} \tag{104}$$

is a matrix of polynomials in D^{-1}.

Since $G(D)$ is basic it follows, by definition, that it has a polynomial right inverse. Hence, it follows from the invariant-factor decomposition (70) that

$$\alpha_1(D) = \alpha_2(D) = \cdots = \alpha_b(D) = 1 \tag{105}$$

(all $\beta_i(D)$ are trivially 1 for a polynomial matrix). Let $\Delta_i(G(D))$ be the greatest common divisor of the $i \times i$ minors of $G(D)$. Since [5]

$$\Delta_i(G(D)) = \alpha_1(D)\alpha_2(D)\cdots\alpha_i(D), \tag{106}$$

we have in particular

$$\Delta_b(G(D)) = \Delta_{b-1}(G(D)) = 1. \tag{107}$$

An $i \times i$ minor of $G_{-1}(D)$ is equal to the corresponding minor of $G(D)$ multiplied by D^{-ni}. Hence, by Lemma 21, we have

$$\Delta_b(G_{-1}(D)) = D^{-(nb-r)} \tag{108}$$

and

$$\Delta_{b-1}(G_{-1}(D)) = D^{-(n(b-1)-s)}. \tag{109}$$

Thus the bth invariant-factor of $G_{-1}(D)$ is [5]

$$\frac{\Delta_b(G_{-1}(D))}{\Delta_{b-1}(G_{-1}(D))} = \frac{D^{-n}}{D^{-(r-s)}}. \tag{110}$$

From (103) and (110), it follows that the bth invariant factor of $G(D)$, regarded as a matrix over $\mathbf{F}_2[D^{-1}]$, is

$$\frac{1}{D^{-n}} \cdot \frac{D^{-n}}{D^{-(r-s)}} = \frac{1}{D^{-(r-s)}}.$$

□

We are now ready to prove the following new theorem which was recently formulated by Forney [9].

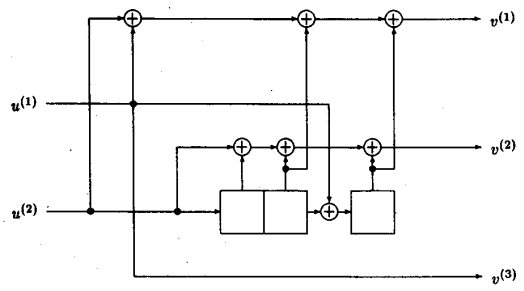

Fig. 8. Minimal encoder for the encoding matrix $G'(D)$ given in (111).

Theorem 23: A basic encoding matrix $G(D)$ is minimal if and only if the maximum degree of its $b \times b$ minors is not less than the maximum degree of its $(b-1) \times (b-1)$ minors.

Proof: From Theorem 19, it follows that a basic encoding matrix $G(D)$ is minimal if and only if it has a polynomial right inverse in D^{-1}. By the invariant factor decomposition $G(D)$ has a polynomial right inverse in D^{-1} if and only if the inverse of its bth invariant factor, regarded as a matrix over $\mathbf{F}_2[D^{-1}]$, is a polynomial in D^{-1}. By applying Lemma 22, the theorem follows. □

This theorem follows also from Forney's global invertibility test [7].

VII. Minimal Encoders

We shall now return to our favorite encoding matrix given in Example 2, viz.,

$$G'(D) = \begin{pmatrix} 1+D & D & 1 \\ 1+D^2+D^3 & 1+D+D^2+D^3 & 0 \end{pmatrix}. \quad (111)$$

In Example 3, we showed that $G'(D)$ is not minimal-basic, i.e., $\mu < \nu$. Its controller canonical form (Fig. 7) requires four memory elements but the controller canonical form of an equivalent encoding matrix (Fig. 6) requires only three memory elements. However, $G'(D)$ is a basic encoding matrix and, hence, it has a polynomial right inverse. Furthermore, it has a polynomial right inverse in D^{-1}, viz.,

$$G'^{-1}_{-1}(D) = \begin{pmatrix} 1+D^{-1}+D^{-2}+D^{-3} & D^{-1} \\ 1+D^{-1}+D^{-3} & D^{-1} \\ D^{-2}+D^{-3} & D^{-1} \end{pmatrix}. \quad (112)$$

Thus, from Theorem 19, we conclude that $G'(D)$ is indeed a minimal encoding matrix!

Definition: A *minimal encoder* is a realization of a minimal encoding matrix $G(D)$ with a minimal number of memory elements over all realizations of $G(D)$.

Theorem 24 (Forney [1]): The controller canonical form of a minimal-basic encoding matrix is a minimal encoder.

Proof: The proof follows immediately from Corollary 7 and Corollary 17. □

Example 5: The realization shown in Fig. 8 of the minimal encoding matrix $G'(D)$ given in (111) is a minimal encoder. (This realization was obtained by minimizing $G'(D)$ using a standard sequential circuits minimization method.)

Notice that the minimal realization shown in Fig. 8 is neither in controller canonical nor observer canonical form! This particular minimal encoding matrix does not have a *minimal* controller canonical form, but it has, of course, an *equivalent minimal-basic* encoding matrix whose controller canonical form (Fig. 6) is a minimal encoder for the same convolutional code.

VIII. Systematic Encoders

In a rate $R = b/c$ convolutional code encoded by a systematic encoding matrix, the b information sequences appear among the c output sequences. Without loss of generality, we assume that the first b output sequences are the exact replicas of the b input sequences. Systematic convolutional encoding matrices are simpler to implement, have trivial right inverses, but unless we use rational encoding matrices, i.e., allow feedback in the encoder, they are, as we know (see e.g., [10]), in general, less powerful when used together with maximum likelihood decoding.

A basic encoding matrix has the greatest common divisor of all $b \times b$ minors equal to 1 [1]. Thus, it follows that every basic encoding matrix must have some $b \times b$ submatrix whose determinant is a delayfree polynomial, since otherwise all subdeterminants would be divisible by D. Premultiplication by the inverse of such a submatrix yields an equivalent systematic encoding matrix, possibly rational. Thus, we have the following.

Theorem 25 (Costello [11]): Every convolutional encoding matrix is equivalent to a systematic rational encoding matrix.

Example 6: Consider the rate $R = 2/3$ nonsystematic convolutional encoder illustrated in Fig. 6. It has the minimal-basic encoding matrix

$$G(D) = \begin{pmatrix} 1+D & D & 1 \\ D^2 & 1 & 1+D+D^2 \end{pmatrix}$$

with $\mu = \nu = 3$. Let $T(D)$ be the matrix consisting of the first two columns of $G(D)$:

$$T(D) = \begin{pmatrix} 1+D & D \\ D^2 & 1 \end{pmatrix}.$$

We have $\det(T(D)) = 1+D+D^3$, and

$$T^{-1}(D) = \frac{1}{1+D+D^3}\begin{pmatrix} 1 & D \\ D^2 & 1+D \end{pmatrix}.$$

Multiplying $G(D)$ by $T^{-1}(D)$ yields a systematic encoding matrix $G_{\text{sys}}(D)$ equivalent to $G(D)$:

$$\begin{aligned} G_{\text{sys}}(D) &= T^{-1}(D)G(D) \\ &= \frac{1}{1+D+D^3} \\ &\quad \begin{pmatrix} 1 & D \\ D^2 & 1+D \end{pmatrix}\begin{pmatrix} 1+D & D & 1 \\ D^2 & 1 & 1+D+D^2 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & \frac{1+D+D^2+D^3}{1+D+D^3} \\ 0 & 1 & \frac{1+D^2+D^3}{1+D+D^3} \end{pmatrix}. \end{aligned}$$

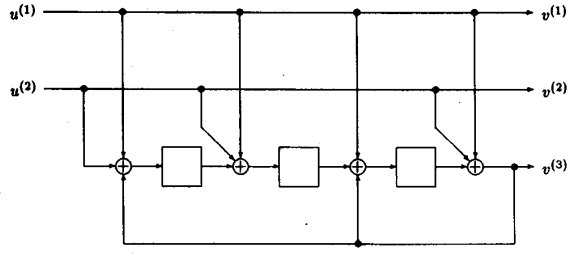

Fig. 9. Observer canonical form of the systematic encoding matrix in Example 6.

Its observer canonical form requires a linear sequential circuit with feedback and $\mu = 3$ memory elements as shown in Fig. 9.

The systematic encoding matrix in the previous example was realized with the same number of memory elements as the equivalent minimal-basic encoding matrix (Example 3). Hence, it is a minimal encoding matrix. Every systematic encoding matrix can be written (15)

$$G(D) = (I_b \; R(D)), \tag{113}$$

where I_b is a $b \times b$ identity matrix and $R(D)$ a $b \times (c-b)$ matrix whose entries are rational functions of D. Such a systematic encoding matrix $G(D)$ has a trivial right inverse, viz., the $c \times b$ matrix

$$G^{-1}(D) = \begin{pmatrix} I_b \\ \mathbf{0} \end{pmatrix}, \tag{114}$$

which is polynomial in both D and D^{-1}. Hence, it follows from Theorem 19 that this minimality holds in general.

Theorem 26 (Forney [1]): Every systematic encoding matrix is minimal.

An Algorithm to Construct a (Minimal) Systematic Encoding Matrix Equivalent to a Given Minimal-Basic Encoding Matrix:

1) Find a $b \times b$ nonzero minor of the minimal-basic encoding matrix $G(D)$ and let it be the determinant of the $b \times b$ submatrix $T(D)$ of $G(D)$.
2) Compute
$$T^{-1}(D)G(D),$$
which is a (minimal) systematic encoding matrix equivalent to $G(D)$.
3) IF $T^{-1}(D)G(D) \neq (I_b \; R(D))$, THEN permute the columns of $T^{-1}(D)G(D)$ and form $G_{\text{sys}}(D) = (I_b \; R(D))$, where $G_{\text{sys}}(D)$ is a (minimal) systematic encoding matrix that encodes a convolutional code which is *equivalent* to the code that $G(D)$ encodes.

Example 7: Consider the rate $R = 2/4$ minimal-basic encoding matrix

$$G(D) = \begin{pmatrix} 1+D & D & 1 & D \\ D & 1 & D & 1+D \end{pmatrix}$$

with $\mu = \nu = 2$. Let

$$T(D) = \begin{pmatrix} 1+D & D \\ D & 1 \end{pmatrix}.$$

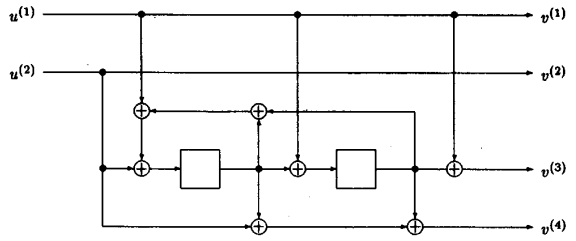

Fig. 10. Minimal realization of the systematic encoding matrix in Example 7.

Then, we have

$$T^{-1}(D) = \frac{1}{1+D+D^2}\begin{pmatrix} 1 & D \\ D & 1+D \end{pmatrix}$$

and

$$\begin{aligned} G_{\text{sys}}(D) &= T^{-1}(D)G(D) \\ &= \frac{1}{1+D+D^2} \\ &\quad \begin{pmatrix} 1 & D \\ D & 1+D \end{pmatrix}\begin{pmatrix} 1+D & D & 1 & D \\ D & 1 & D & 1+D \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & \frac{1+D^2}{1+D+D^2} & \frac{D^2}{1+D+D^2} \\ 0 & 1 & \frac{D^2}{1+D+D^2} & \frac{1}{1+D+D^2} \end{pmatrix}. \end{aligned}$$

Clearly, $G_{\text{sys}}(D)$ has neither a minimal controller canonical form nor a minimal observer canonical form but by standard minimization techniques for sequential circuits we obtain the minimal realization shown in Fig. 10.

IX. Conclusion

In this semitutorial paper, we have given a summary of some of Forney's previous work along with a few new contributions. Most important among the new results are Theorem 19d), the surprising fact that there exist basic encoding matrices that are minimal but not minimal-basic, the existence of basic encoding matrices that are nonminimal, and a recent result, due to Forney, which states exactly when a basic encoding matrix is minimal (Theorem 23).

X. Comments

Current work on minimal encoders over groups (e.g., Forney and Trott [12], Loeliger and Mittelholzer [8]) constructs canonical minimal encoders from a set of shortest linearly independent code sequences ("trellis-oriented generators"), a type of construction that may have been first published by Roos [13]; see also Piret [14]. In Forney [7], the approach of [1] is extended to generalized minimal encoders that are in controller canonical form but not necessarily polynomial.

Acknowledgment

The authors' debt to D. Forney is both obvious and gratefully acknowledged. The authors are also grateful to J. Massey for his interest and encouragement over the years. Without their stimulation, the authors would not have reached the view of convolutional codes that is presented in this paper. D. Forney's detailed comments on the manuscript have been of great value.

REFERENCES

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

[2] ——, "Correction to 'Convolutional codes I: Algebraic structure,'" *IEEE Trans. Inform. Theory*, vol. IT-17, pp. 360, 1971.

[3] ——, "Structural analyses of convolutional codes via dual codes," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 512–518, 1973

[4] J. L. Massey and M. K. Sain, "Inverses of linear sequential circuits," *IEEE Trans. Comput.*, vol. C-17, pp. 330–337, 1968.

[5] N. Jacobson, *Basic Algebra I*. San Fransisco, CA: Freeman, 1974.

[6] T. Kailath, *Linear systems*. Englewood Cliffs, NJ: Prentice Hall, 1980.

[7] G. D. Forney, Jr., "Algebraic structure of convolutional codes, and algebraic system theory," in *Mathematical System Theory*, A. C. Antoulas, Ed. Berlin: Springer-Verlag, 1991, pp. 527–558.

[8] H.-A. Loeliger and T. Mittelholzer, "Convolutional codes over groups," submitted to *IEEE Trans. Inform. Theory*, 1992.

[9] G. D. Forney, Jr., Private communication, Aug. 1991.

[10] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. New York: McGraw-Hill, 1979

[11] D. J. Costello, Jr., "Construction of convolutional codes for sequential decoding," Rep. EE-692, Univ. of Notre Dame, Notre Dame, IN, Aug. 1969.

[12] G. D. Forney, Jr. and M. D. Trott, "The dynamics of linear codes over groups: state spaces, trellis diagrams and canonical encoders," submitted to *IEEE Trans. Inform. Theory*, 1992.

[13] C. Roos, "On the structure of convolutional and cyclic convolutional codes," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 676–683, 1979.

[14] P. Piret, *Convolutional Codes: An Algebraic Approach*. Cambridge, MA: MIT Press, 1988.