CODES, AUTOMATA, AND CONTINUOUS SYSTEMS: EXPLICIT INTERCONNECTIONS

BY

J. L. MASSEY AND M. K. SAIN

Reprinted from IEEE TRANSACTIONS
ON *AUTOMATIC CONTROL*
Volume AC-12, Number 6, December, 1967
pp. 644-650

Codes, Automata, and Continuous Systems: Explicit Interconnections

JAMES L. MASSEY, MEMBER, IEEE, AND MICHAEL K. SAIN, MEMBER, IEEE

Abstract—**Close relationships are established between convolutional codes and zero-state automata and between cyclic codes and zero-input automata. Furthermore, techniques of automata theory and continuous system theory are used to elaborate on the coding problem; and approaches from coding and automata are used to establish and interpret typical structural conditions in continuous systems. The investigation incorporates basic coding concepts into the currently emerging common basis for automata and continuous systems, and it gives explicit examples of the resulting benefits accruing to each of these areas from the others.**

I. Introduction

THE THEORIES of error correcting codes, finite-state machines, and continuous systems have been constructed over the past twenty years in a manner which has left each one largely uninfluenced by the others. Recently, a number of fundamental investigations have structured a unified basis for machines and continuous systems.[1],[2] With a view toward incorporating basic coding concepts into this currently emerging common basis, this paper points out some strong parallelisms between results in coding theory and results in the theories of machines and continuous systems. Furthermore, the strength and clarity of a number of basic inductive arguments, which are typical in the analysis of systems over finite fields, are used to illuminate the meaning of representative algebraic criteria for structural properties of continuous systems.

II. Linear Sequential and Differential Systems

For purposes of brevity, the following discussion is limited to two types of linear, time-invariant systems: the finite-state machine (FSM) and the continuous differential system (CDS).

The FSM is defined by the equations

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B\boldsymbol{u}(k) \tag{1a}$$

$$\boldsymbol{y}(k) = C\mathbf{x}(k) + D\boldsymbol{u}(k) \tag{1b}$$

where $k=0, 1, 2, \cdots$ is the time variable, $\mathbf{x}(k)$ an n-dimensional state vector, $\boldsymbol{u}(k)$ an m-dimensional input vector, $\boldsymbol{y}(k)$ an r-dimensional output vector, and A, B, C, and D are constant matrices of appropriate dimension. Entries in vectors and matrices are elements in the finite field of q elements, denoted by $GF(q)$, and all operations are carried out in this field. Most of the study of such machines has been for the binary case $GF(2)$ in which the operations are addition and multiplication modulo two, but many of the results can be generalized to an arbitrary finite field. Many of the steps which will be carried out for the FSM (1) are of course also valid for the case in which the entries in vectors and matrices are real numbers, but for economy of presentation the explicit treatment of that case has been omitted.

The CDS is defined by the relations

$$\dot{\mathbf{x}} = A\mathbf{x}(t) + B\boldsymbol{u}(t) \tag{2a}$$

$$\boldsymbol{y}(t) = C\mathbf{x}(t) + D\boldsymbol{u}(t) \tag{2b}$$

where t is the positive time variable, and all entries in vectors and matrices are elements in the real field. Dimensions and designations of vectors are the same as those for (1).

Under the assumptions described above, it is clear that the fourtuple (A, B, C, D) structurally characterizes the FSM or the CDS, as the case may be. The context clarifies whether (1) or (2) is under discussion, and so there is no need for a refinement in notation to distinguish between them.

The most important classes of codes which have been found to date are the convolutional codes and the cyclic codes. In the sequel it is shown that these codes are closely related to the zero-state response and the zero-input response, respectively, of the FSM.

III. Convolutional Codes and the Zero-State FSM

Convolutional codes are conveniently described in terms of the delay-operator transform ($\mathfrak{D}$ transform) introduced by Huffman.[3]

Definition 1

The $\mathfrak{D}$ transform of a sequence $i_0, i_1, i_2, \cdots$ of digits from $GF(q)$ is the formal power series

$$I(\mathfrak{D}) = i_0 + i_1\mathfrak{D} + i_2\mathfrak{D}^2 + \cdots. \tag{3}$$

The $\mathfrak{D}$ transform of a sequence of matrices (or vectors) is the matrix (or vector) whose ijth component is the $\mathfrak{D}$ transform of the sequence of ij components.

Manuscript received April 5, 1967; revised August 7, 1967. This work was supported in part by NASA under Grant NsG-334 and the Joint Services Electronics Program under Contract DA208-043-AMC-02536E, both at the M.I.T. Research Laboratory of Electronics, and in part by NSF under Grant GK-804 at the University of Notre Dame. The contribution of the first author was made while he was visiting in the Dept. of Elec. Engrg., M.I.T. The results reported herein were presented at the Sixth Adaptive Processes Symposium, Chicago, Ill., October 1967.

The authors are with the Dept. of Electrical Engineering, University of Notre Dame, Notre Dame, Ind.

It should be observed that multiplication of a transform by $\mathfrak{D}$ corresponds to delaying the elements of the sequence by one time unit.

Let $\boldsymbol{i}_0, \boldsymbol{i}_1, \boldsymbol{i}_2, \cdots$ be a sequence of K-dimensional column vectors over $GF(q)$, each of which constitutes a "block" of information digits to be transformed by some encoder into a sequence $\boldsymbol{v}_0, \boldsymbol{v}_1, \boldsymbol{v}_2, \cdots$ of N-dimensional vectors, each of which constitutes a "block" of encoded digits by means of which the information is transmitted. A *block linear code* is defined by an $N \times K$ matrix G, called the *generating matrix*, such that

$$\boldsymbol{v}_k = G\boldsymbol{i}_k, \qquad k = 0, 1, 2, \cdots. \tag{4}$$

The *elementary condition* for a code to be useful is that the information digits be unambiguously recoverable from the encoded digits, i.e., that (4) be uniquely solvable for $\boldsymbol{i}_k$, which in turn is equivalent to the condition that rank $[G] = K$. This condition requires that $R = K/N \leqq 1$, where R, the number of information digits per encoded digit, is called the *code rate*.

In 1955, Elias[4] generalized the concept of a block linear code to that of a so-called convolutional code.

Definition 2

A convolutional code of memory order M, $M < \infty$, is a code in which the encoded blocks are determined by

$$\boldsymbol{v}_k = G_0\boldsymbol{i}_k + G_1\boldsymbol{i}_{k-1} + \cdots + G_M\boldsymbol{i}_{k-M} \tag{5}$$

where each G_i is an $N \times K$ matrix over $GF(q)$, and where by definition $\boldsymbol{i}_j = \mathbf{0}$ for $j < 0$.

Equation (5) can be written in terms of the $\mathfrak{D}$ transform as

$$\boldsymbol{V}(\mathfrak{D}) = G(\mathfrak{D})\boldsymbol{I}(\mathfrak{D}) \tag{6}$$

where

$$G(\mathfrak{D}) = G_0 + G_1\mathfrak{D} + \cdots + G_M\mathfrak{D}^M. \tag{7}$$

The elementary condition for a code to be useful is then that rank $[G(\mathfrak{D})] = K$, where the entries in $G(\mathfrak{D})$, each being a polynomial of degree M or less, are considered as elements in the field of rational functions over $GF(q)$.

The preceding ideas are closely related to the zero-state response of the FSM (1). Application of Definition 1 to (1) under the assumption that $\mathbf{x}(0) = \mathbf{0}$ gives the transform relationship

$$\boldsymbol{Y}(\mathfrak{D}) = T(\mathfrak{D})\boldsymbol{U}(\mathfrak{D}) \tag{8}$$

where $T(\mathfrak{D})$ is the transfer matrix defined by

$$T(\mathfrak{D}) = C(\mathfrak{D}^{-1}I_n - A)^{-1}B + D \tag{9}$$

whose entries are rational functions in $\mathfrak{D}$. I_n is of course the $n \times n$ identity matrix. Identification of K with m, $\boldsymbol{i}_k$ with $\boldsymbol{u}(k)$, N with r, and $\boldsymbol{v}_k$ with $\boldsymbol{y}(k)$, together with a comparison of (6) and (8), shows that the FSM defines a convolutional code, i.e., may serve as an encoder, if and only if the entries in $T(\mathfrak{D})$ are all polynomials (i.e., the FSM has finite input memory and can be realized by a feedback-free configuration of delay units, scalars, and $GF(q)$ adders) and rank $[T(\mathfrak{D})] = m$. The latter condition is equivalent to the requirement that

$$T(\mathfrak{D})\boldsymbol{U}(\mathfrak{D}) = \mathbf{0} \tag{10}$$

should imply the stronger condition

$$\boldsymbol{U}(\mathfrak{D}) = \mathbf{0}. \tag{11}$$

The identity of (10) and (11) will be called *input functional reproducibility*. The term functional reproducibility was first introduced by Brockett and Mesarović in connection with the identity of the expressions $\boldsymbol{Y}'(s)T(s) = \mathbf{0}'$ and $\boldsymbol{Y}'(s) = \mathbf{0}'$ for the CDS (2), where the prime denotes transposition and s is the Laplace variable. The identity of (10) and (11) may be viewed as a rigid observability requirement on the input of the FSM, whereas the Brockett–Mesarović condition is essentially a rigid controllability requirement on the output of the CDS, i.e., *output functional reproducibility*. Algebraic criteria for output functional reproducibility may be found in Brockett and Mesarović.[5] By duality, they may be applied to input functional reproducibility. It is interesting to note that input functional reproducibility is the condition needed by Perkins and Cruz to construct their classical sensitivity matrix for the CDS.[6]

These ideas may be summarized in the following theorem.

Theorem 1

The FSM (1) with $\mathbf{x}(0) = \mathbf{0}$ is an encoder for some convolutional code if and only if it has finite input memory and exhibits input functional reproducibility. Conversely, the encoder for a convolutional code can always be taken as some FSM (1) with $\mathbf{x}(0) = \mathbf{0}$.

In light of the remarks above, it might seem advantageous in coding theory to remove the restriction that M be finite when defining a convolutional code, i.e., to consider generalized codes such that any FSM (1) with input functional reproducibility would be a valid encoder. That such a generalization has no advantage follows from the next theorem.

Theorem 2

Corresponding to any FSM (1) with $\mathbf{x}(0) = \mathbf{0}$ there is a second FSM (1) with $\mathbf{x}(0) = \mathbf{0}$, having finite input memory and a one-to-one correspondence between input sequences, such that the responses to corresponding input sequences are identical.

The proof is easily obtained by noting that the entries in $T(\mathfrak{D})$ are rational functions, the denominators of which must have nonzero constant terms since the FSM (1) is always causal. Thus the least common multiple, $b(\mathfrak{D})$, of these denominators must also have nonzero constant term. Define the second FSM (1) by the transfer matrix

$$T^*(\mathfrak{D}) = b(\mathfrak{D})T(\mathfrak{D}) \tag{12}$$

which now specifies an FSM with finite input memory.

If $U(\mathfrak{D})$ is an input sequence for the first machine, define the corresponding input sequence for the second machine by

$$U^*(\mathfrak{D}) = U(\mathfrak{D})/b(\mathfrak{D}) \tag{13}$$

which is well defined since $b(0) \neq 0$. Clearly,

$$Y(\mathfrak{D}) = T(\mathfrak{D})U(\mathfrak{D}) = T^*(\mathfrak{D})U^*(\mathfrak{D}) = Y^*(\mathfrak{D}) \tag{14}$$

and the theorem follows.

This simple theorem has a number of interesting practical consequences in coding theory. In the first place, the whole purpose of coding is to correct transmission errors, and a decoder can be considered as a device that maps a received sequence into a (probable) valid encoded sequence $V(\mathfrak{D})$, or $Y(\mathfrak{D})$ in FSM terms. Thus the only aspect important for decoding is the set of encoded sequences, and Theorem 2 shows that the possible sets would not be enlarged by allowing infinite memory encoders. The theorem actually says somewhat more. In fact, an infinite memory encoder could be used (and might be advantageous in gaining an economical realization of the encoder), and the received sequence could be decoded by the same decoder as that used for the conventional finite memory encoder $T^*(\mathfrak{D})$. This decoder would yield the information sequence $U^*(\mathfrak{D})$ from which $U(\mathfrak{D})$ could be recovered by (13), i.e., by another finite memory device. This latter point is important since it guarantees that a decoding error in some digit of $U^*(\mathfrak{D})$ would not propagate indefinitely as errors in the digits of $U(\mathfrak{D})$.

IV. Cyclic Codes and the Zero-Input FSM

The present section takes up the question of cyclic codes.

Definition 3

A cyclic code is a block linear code in which the cyclic shift $(a_{N-1}, a_0, a_1, \cdots, a_{N-2})$ of a codeword $(a_0, a_1, \cdots, a_{N-2}, a_{N-1})$ is always again a codeword.[7]

It is customary in coding theory to identify the polynomial

$$a(x) = a_0 + a_1x + \cdots + a_{N-1}x^{N-1} \tag{15}$$

with the codeword $(a_0, a_1, \cdots, a_{N-1})$. A codeword is, of course, the "block" of encoded digits. It is known that for every cyclic code there is a unique monic (highest coefficient unity) polynomial $g(x)$ such that 1) $a(x)$ is a codeword if and only if $g(x)$ divides $a(x)$, and 2) $g(x)$ divides x^N-1. Conversely, every $g(x)$ which divides x^N-1 generates a cyclic code in this manner. See, for example, Peterson.[7]

The most powerful class of constructive codes presently known, the Bose-Chaudhuri-Hocquenghem codes, are cyclic codes. A systems approach has recently yielded considerable insight into the decoding procedure for these codes.[8]

Cyclic codes are closely related to the zero-input response of the FSM (1), i.e., the response when $\mathbf{u}(k) = \mathbf{0}$, $k=0, 1, 2, \cdots$. Under this assumption, (1) becomes

$$\mathbf{x}(k) = A^k\mathbf{x}(0) \tag{16a}$$

$$\mathbf{y}(k) = CA^k\mathbf{x}(0). \tag{16b}$$

The FSM (1) will be called *periodic* if there is a positive integer T such that $\mathbf{y}(T+k) = \mathbf{y}(k)$ for all k and all $\mathbf{x}(0)$. The smallest such T is the *period*.

Lemma 1

The FSM (1) is zero-input periodic if and only if there is a positive integer T, $T \leqq q^{n^2}$, such that $C = CA^T$.

The lemma follows directly from (16b) and the fact that there are only q^{n^2} distinct $n \times n$ matrices over $GF(q)$.

Theorem 3

Let the FSM (1) have a single output ($r=1$, $C=\mathbf{c}'$), and let i be any positive integer. Furthermore, denote the finite sequence of outputs $\{y(0), y(1), \cdots, y(iT-1)\}$ as the *response segment* Y_{iT} of FSM (1) corresponding to an initial vector $\mathbf{x}(0)$. If the FSM has period T, then the set of Y_{iT} corresponding to the set of $\mathbf{x}(0)$ vectors forms a cyclic code.

In establishing Theorem 3, note first that the set of Y_{iT} forms a linear vector space, for if A_{iT} is the response segment corresponding to $\mathbf{x}(0) = \mathbf{x}_a$ and if B_{iT} is the response segment corresponding to $\mathbf{x}(0) = \mathbf{x}_b$, then $\alpha A_{iT} + \beta B_{iT}$ is the response segment corresponding to $\mathbf{x}(0) = \alpha\mathbf{x}_a + \beta\mathbf{x}_b$. The notation αA_{iT} denotes the finite sequence, each element of which is obtained by multiplying the corresponding element of A_{iT} by α. Finally, since the response segment $\{a_{iT-1}, a_0, a_1, \cdots, a_{iT-2}\}$ is just the response to the input vector $\mathbf{x}(0) = A^{T-1}\mathbf{x}_a$, the above block linear code is a cyclic code.

It is next appropriate to inquire concerning the conditions under which a periodic FSM (1) can serve as an encoder for the cyclic code of Theorem 3 when the initial state $\mathbf{x}(0)$ is taken as the information vector. For simplicity, let $i=1$ in Theorem 3. Again, the elementary condition for the FSM (1) to serve as an encoder is that each initial state yield a distinct codeword Y_T, or equivalently that the initial state $\mathbf{x}(0)$ be recoverable from Y_T. But it is known in coding theory that any K consecutive code digits uniquely specify the codeword in a cyclic code,[7] where K is the *code dimension*. In the present instance, the code dimension is n, so that the initial state must be recoverable from Y_n. These concepts are related to the state observability of the FSM (1).

Definition 4

The zero-input FSM (1) is i-step observable if $\mathbf{x}(0)$ is uniquely determined by Y_i.

Theorem 4

A single-output, periodic, zero-input FSM (1) is a cyclic encoder for the cyclic code specified in Theorem 3, when $\mathbf{x}(0)$ is regarded as the information vector, if and only if the FSM is n-step observable or, in the terminology of the CDS (2), completely state observable. Conversely, every cyclic code may be encoded in this manner by some single-output, periodic, zero-input FSM (1).

The converse follows from the well-known fact[7] that any cyclic code can be encoded by a K-stage linear feedback shift register whose initial state is the information vector.

The FSM (1) is said to be *nonsingular* if A is nonsingular. It is interesting to note that any FSM (1) which is a cyclic encoder must be nonsingular. For if $\mathbf{c}'=\mathbf{c}'A^T$ but $A^T \neq I_n$, as must be the case if A is singular, for otherwise $A^{-1}=A^{T-1}$, then there is some state $\mathbf{x}_a$ such that $\mathbf{x}_a \neq A^T\mathbf{x}_a$. But then the response segment corresponding to $\mathbf{x}(0)=\mathbf{x}_a$ is identical to the response segment corresponding to $\mathbf{x}(0)=A^T\mathbf{x}_a$, and such an FSM violates the elementary condition for an encoder. See also Ash.[9]

Application of Definition 1 to (16) yields the relation

$$\mathbf{Y}(\mathfrak{D}) = C(I_n - \mathfrak{D}A)^{-1}\mathbf{x}(0) = P(\mathfrak{D})\mathbf{x}(0)/h(\mathfrak{D}) \tag{17}$$

where $P(\mathfrak{D})$ is a matrix of polynomials in $\mathfrak{D}$ of degree $n-1$ or less and $h(\mathfrak{D})=\det(I_n-\mathfrak{D}A)=1+h_1\mathfrak{D}+\cdots+h_n\mathfrak{D}^n$. Note that $h_n=\det(-A)$. Letting

$$P(\mathfrak{D}) = P_0 + P_1\mathfrak{D} + \cdots + P_{n-1}\mathfrak{D}^{n-1} \tag{18}$$

it follows from (17) that

$$P_j = C(A^j + h_1A^{j-1} + \cdots + h_jI_n). \tag{19}$$

Equation (19) is simply deduced by solving the identity

$$(I_n - \mathfrak{D}A)(Q_0 + Q_1\mathfrak{D} + \cdots + Q_{n-1}\mathfrak{D}^{n-1}) = h(\mathfrak{D})I_n \tag{20}$$

recursively for Q_i and identifying $P_j=CQ_j$, analogous to the well-known procedure for the CDS (2).[10]

For the single-output case, C may be written as $\mathbf{c}'$ and $P(\mathfrak{D})$ as $\mathbf{p}'(\mathfrak{D})$. Accordingly from (18) it follows that

$$\mathbf{p}'(\mathfrak{D})\mathbf{x}(0) = \mathbf{p}_0'\mathbf{x}(0) + \mathbf{p}_1'\mathbf{x}(0)\mathfrak{D} + \cdots + \mathbf{p}_{n-1}'\mathbf{x}(0)\mathfrak{D}^{n-1} \tag{21}$$

is a polynomial of degree $n-1$ or less; furthermore, (21) can be made equal to any prescribed (degree $n-1$ or less) polynomial, for some $\mathbf{x}(0)$, if and only if the row vectors $\mathbf{c}', \mathbf{c}'A, \cdots, \mathbf{c}'A^{n-1}$ are linearly independent, which is the usual condition for n-step observability. If $\mathbf{p}'(\mathfrak{D})\mathbf{x}(0)=-b(\mathfrak{D})$, where $b(\mathfrak{D})$ may be viewed as an information polynomial, in one-to-one correspondence with $\mathbf{x}(0)$, and if T is the least integer such that $h(\mathfrak{D})$ divides $(\mathfrak{D}^T-1)$, and such a T always exists for any polynomial $h(\mathfrak{D})$ over $GF(q)$ such that $h(0)\neq 0$,[11] then (17) becomes

$$\mathbf{Y}(\mathfrak{D}) = -b(\mathfrak{D})/h(\mathfrak{D}) = b(\mathfrak{D})g(\mathfrak{D})/(1-\mathfrak{D}^T) \tag{22}$$

where

$$g(\mathfrak{D}) = (\mathfrak{D}^T - 1)/h(\mathfrak{D}). \tag{23}$$

But (22) can be rewritten in the form

$$\mathbf{Y}(\mathfrak{D}) = b(\mathfrak{D})g(\mathfrak{D})(1 + \mathfrak{D}^T + \mathfrak{D}^{2T} + \cdots) \tag{24}$$

which is displayed as the transform of a periodic sequence of period T whose first period has the transform $b(\mathfrak{D})g(\mathfrak{D})$. But the set of polynomials $b(\mathfrak{D})g(\mathfrak{D})$, as $b(\mathfrak{D})$ varies over all polynomials of degree $n-1$ or less, is just the cyclic code generated by $g(\mathfrak{D})$. Thus the specific cyclic code associated with a given FSM in Theorem 4 can be easily identified.

Instead of approaching the description of FSM's and cyclic encoders initially through Theorems 3 and 4, it is also possible to begin by construction, i.e., by (17) through (24). In this case the observability conditions arise in connection with undesirable cancellations in

$$\mathbf{Y}(\mathfrak{D}) = \mathfrak{D}^{-1}C(\mathfrak{D}^{-1}I_n - A)^{-1}\mathbf{x}(0). \tag{25}$$

Finally, it seems possible that the problem of solving for $\mathbf{x}(0)$, when $b(\mathfrak{D})$ is given, could be attacked by modifying the inner product lemmas of Liu and Leake[12] for application in $GF(q)$.

V. Controllability, Observability, and Burst Correction

The present discussion relates the basic burst correction property of convolutional codes to observability and controllability conditions and thereby outlines a procedure for the construction of FSM's with rigid restrictions in this regard.

Definition 5

The FSM (1) is i-step controllable if, for every prescribed state $\mathbf{e}$ and initial state $\mathbf{x}(0)$, there is an input sequence $\mathbf{u}(0), \mathbf{u}(1), \cdots, \mathbf{u}(i-1)$ such that $\mathbf{x}(i)=\mathbf{e}$. This is the case if and only if the $n\times im$ matrix

$$[B \mid AB \mid A^2B \mid \cdots \mid A^{i-1}B] \tag{26}$$

has rank n.[13] For $i=n$, Definition 5 is equivalent to that introduced by Kalman *et al.*[14] for the CDS (2). For $i<n$, Definition 5 is closely related to the recent correspondence by Chen *et al.*[15]

Consider now a convolutional code with finite input memory M.

Definition 6

A convolutional code with finite input memory M is said to be a *basic burst correcting code* if the information sequence can be recovered from the corrupted encoded sequence provided that, if $\mathbf{y}(i)$ has any corrupted digits, the following M outputs $\mathbf{y}(i+1), \mathbf{y}(i+2), \cdots, \mathbf{y}(i+M)$ are error free. Suppose now that $m=r-1$, that is, the output contains just one more digit than the input.

Furthermore, suppose that the first $r-1$ components of $y(i)$ are just the components of $u(i)$. A code with this property, in which the information digits appear unchanged among the encoded digits, is said to be *systematic*. Under these assumptions, the transfer matrix $T(\mathfrak{D})$ may be written

$$T(\mathfrak{D}) = [I_{r-1} \mid g(\mathfrak{D})]' \tag{27}$$

where $g(\mathfrak{D})$ is a column vector whose ith component $G_i(\mathfrak{D})$ is a polynomial of degree M or less. These polynomials may be specified by the $(M+1)\times r$ matrix S whose first column is $(1\ 0 \cdots 0)$ and whose $(i+1)$th column read downward gives the coefficients of $G_i(\mathfrak{D})$. If a basic burst correcting code with parameters M and r exists, Preparata[16] has shown that there exists such a code for which

$$S = [I_r \mid P']' \tag{28}$$

where P is some $(M+1-r)\times r$ matrix, and that a necessary condition for this existence is just $M \leqq 2r-1$.

Definition 7

A basic burst correcting convolutional code with parameters M and r is said to be *optimal* if $M=2r-1$. For an optimal code, S is a $2r\times r$ matrix.

Now let $\mathbf{0}$ be a $(2r-1)$-dimensional null column vector. Then a shifting matrix

$$L = \left[\begin{array}{c|c} \mathbf{0}' & 0 \\ \hline I_{2r-1} & \mathbf{0} \end{array}\right] \tag{29}$$

can be defined, and an optimal code is characterized by the following theorem.

Theorem 5 (Preparata)

A $2r\times r$ matrix S of the form (28) describes an optimal code if and only if the matrices

$$[S \mid L^i S], \qquad i = 1, 2, \cdots, r-1, \tag{30}$$

have rank $2r$, that is, are nonsingular. Accordingly, from Definition 5 and Theorem 5, Theorem 6 follows.

Theorem 6

A $2r\times r$ matrix S of the form (28) describes an optimal, systematic, basic burst correcting code if and only if the FSM (1), with $A=L^i$ and $B=S$, is 2-step controllable for $i=1, 2, \cdots, r-1$.

The utility of Theorem 6 follows from the fact that Berlekamp[17] and Preparata have independently shown that optimal codes exist for every r and have given constructive procedures for finding the associated S matrix. The discussion here, therefore, shows that this procedure is also a synthesis technique for linear, time-invariant FSM's with a very rigid controllability requirement. By duality, a similar argument establishes a procedure for determining FSM's with a rigid observability property.

VI. Structural Properties in the CDS

Since 1962, the CDS (2) has been repeatedly examined with respect to its structural properties. The results of such investigations are usually stated algebraically in terms of the matric properties of the characterizing arrays (A, B, C, D). For example, matrices of the form CA^kB have been used by Kreindler and Sarachik[18] to test for output controllability, by Brockett and Mesarović to test for functional reproducibility, by Cruz and Perkins to test for signal and parameter invariance[19] and more recently by Falb and Wolovich[20] to characterize their solution to the long studied problem of decoupling the variables in a multivariable system by means of state feedback. In this final section, it is shown that the clarity and simplicity of an elementary argument well known to analysts of the FSM (see, for example, Gill[21]) offers a conceptually simple alternative for the description of these properties.

For brevity of presentation, and with no loss of generality, let $D=0$. With each threetuple (A, B, C) associate an all-integrator block diagram in the manner indicated in Fig. 1.

Definition 8

The ijth element, B_{ij}, of the matrix B in (2) is the *path gain* from the input u_j to the input of the integrator whose output is denoted by x_i (Fig. 1). Analogously, A_{ij} is the path gain from the output of the integrator associated with x_j (more simply the jth integrator) to the input of the ith integrator, and C_{ij} is the path gain from the output of the jth integrator to the output y_i.

Lemma 2

The ijth element, $(A^kB)_{ij}$, of the matrix A^kB is the sum of the path gains from input u_j to integrator input i, where each path passes through exactly k integrators.

The proof is a straightforward induction. By definition B_{ij} is the path gain from u_j to integrator input i, where the path passes through no integrators ($k=0$). Next examine $(AB)_{ij}$, which corresponds to $k=1$. Writing $(AB)_{ij}$ as an $n$fold sum over m_1 of elements $A_{im_1}B_{m_1j}$, and identifying each such element as the product of the path gain from input u_j to integrator input m_1 with the path gain from integrator output m_1 to integrator input i, it follows that $(AB)_{ij}$ is just the sum of the path gains from input u_j to integrator input i, where each path passes through exactly one integrator. The lemma follows in an obvious manner and leads immediately to the following result.

Theorem 7

The ijth element, $(CA^kB)_{ij}$, of CA^kB is the sum of the path gains from input u_j to output y_i, where each path passes through exactly $k+1$ integrators.

Note that Definition 8 and Theorem 7 do not assume nontouching or nonrepetitive paths. The only restriction is on the number of integrators (not necessarily distinct) through which the path passes.

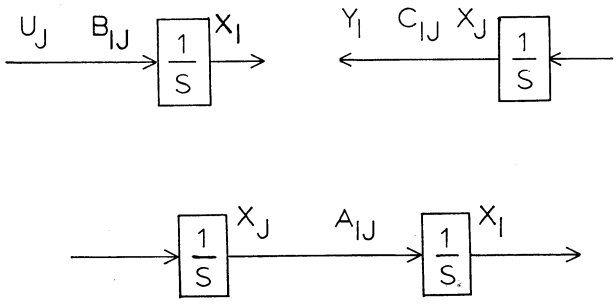

Fig. 1. Identification of the all-integrator block diagram associated with the threetuple (A, B, C).

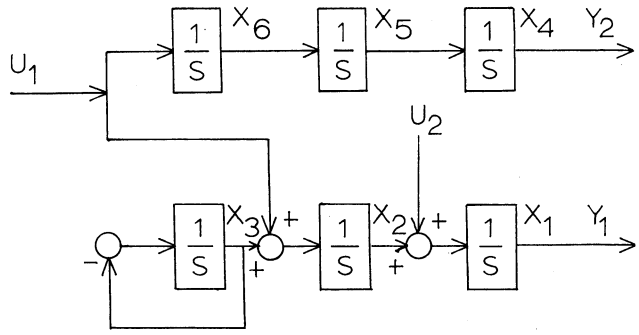

Fig. 2. All-integrator block diagram of multivariable system to be decoupled.

Because of space limitations, applications of Theorem 7 will be limited to two short examples. Further applications will be apparent to analysts familiar with the CDS (2). Consider first the question of state-variable decoupling in multivariable systems. It is known that the CDS (A, B, C), where $m=r$, can be decoupled (i.e., one input controls exactly one and only one output) by a state-variable feedback of the form $\mathbf{u}(t)=F\mathbf{x}(t)+G\boldsymbol{\omega}(t)$, where G is nonsingular and $\boldsymbol{\omega}(t)$ is the new input, if and only if the row vectors $\mathbf{c}_iA^{d_i}B$, $i=1, 2, \cdots, m$, are linearly independent. The row vector $\mathbf{c}_i$ is the ith row of C, and d_i is the minimum integer in the set $0, 1, \cdots, n-1$ such that $\mathbf{c}_iA^{d_i}B\neq\mathbf{0}'$. For a proof, see Falb and Wolovich.[20] From Theorem 7 it follows that the jth component of the row vector $\mathbf{c}_iA^{d_i}B$ is the sum of the path gains from input u_j to output y_i, where each path passes through exactly d_i+1 integrators. Accordingly, the vector $\mathbf{c}_iA^{d_i}B$ represents the relative effects of the m inputs u_j, $j=1, 2, \cdots, m$, on the output y_i after d_i+1 integrations, or equivalently after the minimum number of integrations which is necessary to observe an output at y_i. Thus Theorem 7 can be used to give alternate interpretations for structural conditions.

Another application of the result, for low-order systems in which the all-integrator block diagram description is useful for design, is in the construction and modification of such matric test arrays. The following example is due to Falb and Wolovich. The characterizing threetuple is chosen to be

$$\left(\begin{matrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}, \begin{matrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{matrix}, \begin{matrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{matrix}\right)$$

for which an all-integrator diagram in Fig. 2 has been drawn. By Theorem 7, $\mathbf{c}_1A^{d_1}B$ is immediately identified as $\mathbf{c}_1B$, since there is a nonzero gain path from u_2 to y_1, and is given by inspection from the diagram by the row vector $(0\ 1)$. Similarly, $\mathbf{c}_2A^{d_2}B=\mathbf{c}_2A^2B$, since the first path from any input to y_2 is from u_1 and passes through three integrators, and $\mathbf{c}_2A^2B=(1\ 0)$ also by inspection of Fig. 2. Thus Theorem 7 can be used in low-order systems to construct matric arrays for structural properties, in much the same way that the Mason gain formula is applied in such cases. In many cases, including the above example, application of Theorem 7 is simpler than direct matrix multiplications. The simplicity of Theorem 7 also points the way toward modification of the system to improve the structural properties.

VII. Conclusions

The purpose of this investigation was to establish explicit interconnections between the results of coding theory, automata theory, and continuous system theory. The paper achieves these objectives in a twofold manner. First, it establishes basic and characteristic results from which further work might proceed. In this regard, consider Theorems 1 and 4 on convolutional and cyclic codes, respectively. Second, it gives explicit examples of benefits accruing to each of these areas from the others; specific cases of these are Theorems 6 and 7 pertaining to structural conditions in automata and continuous systems. By pointing out specific parallelisms between coding, automata, and continuous system theories, the work clearly indicates the advantages of an increased exchange of ideas between these disciplines. By providing basic coding results in a framework which is more accessible to nonspecialists, the work increases the opportunity for such an exchange of ideas.

References

[1] M. A. Arbib, "A common framework for automata theory and control theory," *SIAM J. Control*, vol. 3, pp. 206–222, 1965.

[2] P. H. Swain and K. S. Fu, "On relationships between the theories of discrete-state and continuous-state linear time-invariant systems," *Proc. 4th Ann. Allerton Conf. on Circuit and System Theory* (Urbana, Ill., October 1966), pp. 794–804.

[3] D. A. Huffman, "The synthesis of linear coding networks," in *Information Theory*, C. Cherry, Ed. New York: Academic Press, 1953.

[4] P. Elias, "Coding for noisy channels," *1955 IRE Conv. Rec.*, vol. 3, pt. 4, pp. 37–46.

[5] R. W. Brockett and M. D. Mesarović, "The reproducibility of multivariable systems," *J. Math. Anal. and Appl.*, vol. 11, pp. 548–563, 1965.

[6] M. K. Sain, "Functional reproducibility and the existence of classical sensitivity matrices," *IEEE Trans. Automatic Control (Correspondence)*, vol. AC-12, p. 458, August 1967.

[7] W. W. Peterson, *Error Correcting Codes*. New York: Wiley, 1963, pp. 137–160.

[8] J. L. Massey, "Shift register synthesis and BCH decoding," *IEEE Trans. Information Theory* (to be published).
[9] R. Ash, *Information Theory*. New York: Interscience, 1965, ch. 5.
[10] L. A. Zadeh and C. A. Desoer, *Linear System Theory*. New York: McGraw-Hill, 1963, pp. 302–306.
[11] W. W. Peterson,[7] p. 138.
[12] R. W. Liu and R. J. Leake, "Exhaustive equivalence classes of optimal systems with separable controls," *SIAM J. Control*, vol. 4, pp. 678–685, 1966.
[13] M. Cohn, "Controllability in linear sequential networks," *IRE Trans. Circuit Theory*, vol. CT-9, pp. 74–78, March 1962.
[14] R. E. Kalman, Y. C. Ho, and K. S. Narendra, "Controllability of linear dynamical systems," *Contrib. Differential Equations*, vol. 1, pp. 189–213, 1962.
[15] C. T. Chen, C. A. Desoer, A. Niederlinski, and R. E. Kalman, "Simplified conditions for controllability and observability of linear time-invariant systems," *IEEE Trans. Automatic Control* (*Correspondence*), vol. AC-11, pp. 613–614, July 1966.
[16] F. P. Preparata, "Systematic construction of optimal linear recurrent codes for burst error protection," *Calcolo*, vol. 1, pp. 147–153, 1964.
[17] E. R. Berlekamp, "Note on recurrent codes," *IEEE Trans. Information Theory* (*Correspondence*), vol. IT-10, pp. 257–258, July 1964.
[18] E. Kreindler and P. E. Sarachik, "On the concepts of controllability and observability of linear systems," *IEEE Trans. Automatic Control*, vol. AC-9, pp. 129–136, April 1964.
[19] J. B. Cruz, Jr., and W. R. Perkins, "Conditions for signal and parameter invariance in dynamical systems," *IEEE Trans. Automatic Control*, vol. AC-11, pp. 614–615, July 1966.
[20] P. L. Falb and W. A. Wolovich, "On the decoupling of multivariable systems," Preprints, Joint Automatic Control Conf. (Philadelphia, Pa., June 1967), pp. 791–796. Also "Decoupling in the design and synthesis of multivariable control systems," this issue, pp. 651–659.
[21] A. Gill, *Finite State Machines*. New York: McGraw-Hill, 1962.

James L. Massey (S'54–M'55) was born in Wauseon, Ohio, on February 11, 1934. He received the B.S. degree in electrical engineering from the University of Notre Dame, Notre Dame, Ind., in 1956, and the S.M. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1960 and 1962, respectively.

From 1956 to 1959 he served on active duty as a communications officer in the U. S. Marine Corps. Since 1962 he has been on the faculty of the University of Notre Dame where his research interests have been in the areas of coding and finite-state automata, and where he is now Professor of Electrical Engineering. During the academic year 1966–1967 he held a visiting appointment in the Department of Electrical Engineering at M.I.T.

Dr. Massey is a member of Eta Kappa Nu, Sigma Xi, Tau Beta Pi, and the American Society for Engineering Education.

Michael K. Sain (S'57–M'65) was born in St. Louis, Mo., on March 22, 1937. He received the B.S. degree, summa cum laude, and the M.S. degree (research option) in electrical engineering from St. Louis University, St. Louis, Mo., in 1959 and 1962, respectively. He received the Ph.D. degree from the University of Illinois, Urbana, in 1965.

From 1955 to 1965 he studied under the scholarship and fellowship support of the General Motors Corp., the Motorola Corp., St. Louis University, the University of Illinois, the National Science Foundation, and the National Electronics Conference. Presently, he is Assistant Professor in the Department of Electrical Engineering at the University of Notre Dame, Notre Dame, Ind., where he teaches courses in control and systems theory. His current research interest is the optimization of stochastic systems.

Dr. Sain is a member of the Society for Industrial and Applied Mathematics and Sigma Xi.