The Journal of National Chiao Tung University
Vol. 8, July 1980
pp. 1-8

有限狀態機狀態識別與控制之矩陣方法

On a Matrix Approach to State Identification and Control of Finite-State Machines *

達斯，盛慶琜，陳稔 **Sunil R. Das, Ching-Lai Sheng, and Zen Chen**

Institute of Computer Science, N. C. T. U.

(Received February 9, 1979)

Abstract—The response of a nontrivial sequential machine to specified excitations becomes unpredictable if the state of the machine is unknown. On the other hand, the response of the machine can always be predicted if the initial state is known. Hence one of the basic problems in the study of sequential machines is to identify the state of the machine under investigation. Once the state is identified, the behavior of the machine under all future circumstances becomes predictable, and definite steps may then be taken to force the machine into various modes of operation at the discretion of the investigator. The former class of problems comes under the broad category of problems usually termed the state identification problems, and the latter problem is commonly known as the control problem in sequential machines. One of the most important state identification problems is that of identifying the unknown initial state of the machine, called the initial state identification problem or diagnosing problem; whereas, another important state identification problem is relating to that of identifying the terminal state of the machine, known as the terminal state identification problem or homing problem, of which the special case is the synchronizing problem. The solution to either of these state identification problems constitutes the solution to the basic problem of rendering the machine predictable to the investigator. In the present paper, instead of resorting to the conventional procedure of using the transition table and the corresponding response tree, use is made of the transition matrix representation of the machine and its higher-order forms to solve the aforementioned state identification and control problems. The developed approach is not only simple, but very systematic, and completely algorithmic, and thus lends itself to easy computer implementation.

I. Introduction

While choosing experimental programs for conducting experiments in sequential machines, considered as a black box with only accessible input and output terminals, an experimenter has to keep in view the type of problem he has to solve. The experimenter may have to deal with a situation in which he knows very little about the device he experiments upon, excepting only that the device is a sequential machine with a given input alphabet, belonging to a general class of machines. This problem is the *machine identification problem,* which is usually a rather difficult problem to solve. In this case the experimenter is required to know not only the machine's complete input alphabet, but also a bound on the maximum number of states that the machine can have; in addition, the machine under investigation has to be strongly connected. Alternatively, the experimenter may have to consider a class of problems, known as the *measurement and control problems,* that are less formidable to solve, because in this case the experimenter is required to conduct experiments on a machine of which the transition table is supplied. In this latter case, the experimenter's interest is primarily in measuring and controlling the various machine parameters [1].

The kind of experiment that an experimenter can perform is usually limited by the number of copies of the machine available for investigation, the amount of flexibility allowed to the experimenter, and also the amount of a priori information available regarding the machine's internal behavior. In this paper we consider the measurement and control problems of sequential machines with the constraint that we conduct our experiments on a single copy

* The research reported in this paper was supported in part by the National Science Council of the Republic of China. The paper was originally accepted for presentation at the Twelfth Annual Asilomar Conference on Circuits, Systems, and Computers, Pacific Grove, California, U. S. A., November 6-8, 1978.

of the machine, thereby confining ourselves to only *simple experiments*; we further assume that the input sequences to be applied to the machine are fixed in advance so that we are required to perform *preset experiments.* A simple preset experiment is generally easy to implement, though this type of experiments suffers from the disadvantage that it tends to be lengthy, and sometimes does not provide the experimenter with the desired information [1-5].

Assume that in the present study of the measurement and control problems, we are dealing with a finite, deterministic, completely specified, synchronous sequential machine defined by the quintuple $M = <I,S,0,f,g>$ where $I = I_1, I_2, ..., I_u$ denotes the input alphabet, $S = S_1, S_2,...,S_v$ denotes the state alphabet, and $0 = 0_1, 0_2,...,0_w$ the output alphabet, and f and g denote the two *characterizing functions* of machine M given by $S_{N+1} = f(I_N, S_N)$, and $0_N = g(I_N, S_N)$. For a Mealy machine, 0_N is the corresponding output of I_N and $(I_N, 0_N)$ forms an *input-output pair.* For a Moore machine [2], the output corresponding to I_N is 0_{N+1} and $(I_N, 0_{N+1})$ forms an *input-output pair.* An *input sequence* $\overline{I}_i = I_{i1} I_{i2} ... I_{iL}$, of *length* L, is a number of L inputs successively applied to the machine M in a certain initial state S_j. An *output sequence* $\overline{0}_k = 0_{k1}0_{k2}... 0_{kL'}$, of *length* L', is a number of L' outputs successively produced by the machine M when an input sequence is applied. An output sequence $\overline{0}_k$ is called the *corresponding output sequence* of an input sequence $\overline{I}_i$ if and only if $L = L'$, and $(I_{ih}, 0_{kh})$, $h=1,2,...,L$, is an input-output pair [11]. Assume further that the machine M under consideration is a minimal machine. Now if we allow $\overline{I}_i$ to represent any possible input sequence of M, we can always evaluate the functions $f(\overline{I}_i, S_r)$ and $g(\overline{I}_i, S_r)$ for every state S_r in the state set S, $f(\overline{I}_i, S_r)$ denoting the terminal state reached, and $g(\overline{I}_i, S_r)$ denoting the output sequence produced, on application of $\overline{I}_i$ at S_r of M [1].

The easiest control problem arises when we know that the machine M is in an initial state S_i and we intend to change its state to some other state S_k. To accomplish this, we are required to find an input sequence $\overline{I}_i$ such that $f(\overline{I}_i, S_i) = S_k$. While dealing with a strongly connected machine, we know that such an input sequence always exists, and can also be readily found using the transition diagram, or even the transition table of the machine.

The initial state of the machine M, however, is usually unknown, or sometimes may only be partially known. The task of bringing the machine M to a specified terminal or final state in such circumstances generally involves a control process that is a two-step adaptive process. Initially, an input sequence is applied that brings the machine M from its unknown initial state to a known intermediate state, which is identifiable by observing the resulting output sequence. Once this intermediate state is identified, in the next phase, a second input sequence is selected which is to take the machine to the desired final state. Thus the general machine control problem can be viewed as being comprised of two distinct subproblems : a measurement problem followed by a simple control problem.

There may be many different measurement problems of concern in sequential machines depending upon which parameters of the machine we assume to be known, which parameters we assume to be unknown, and also which of the machine parameters we can vary in a controlled manner. One important measurement problem in sequential machines happens to be the *initial state identification problem*, also called the *diagnosing problem,* which deals with the problem of determination of the unknown initial state of the machine. This kind of problem may be of immense significance while we are trouble-shooting a machine. In case we can identify the state of the machine after an error has disrupted the machine's operation, we may have a clue to the cause of the error resulting in the malfunction. While solving this problem we apply a predetermined input sequence $\overline{I}_i$ to the machine M and observe the corresponding output sequence $g(\overline{I}_i, S_k)$, and then, on the basis of the information provided by the observed output sequence, we are able, possibly, to define the unknown initial state S_k. Unfortunately, not all initial state identification problems have unique solutions, or rather solvable, to say more explicitly [12], as we shall see later. Another measurement problem of much interest in sequential machines is the *terminal state identification problem*, also called the *homing problem.* We assume in this case that the machine M under investigation is in some initial state S_k. We then apply a known input sequence $\overline{I}_i$ to the machine and observe the resulting output sequence

$g(\bar{I}_i, S_k)$. Based on this observation we are then able to specify the terminal state $S_m = f(\bar{I}_i, S_k)$. Fortunately, for every minimal machine M, the terminal state identification problem is solvable [12]. A very special case of the terminal state identification problem is termed the *synchronizing problem*, which is also another measurement problem of considerable importance. There are some machines for which it is possible to use a single input sequence to take the machine from any unknown initial state to a specific, predefined known state.

The measurement and control problems of sequential machines, including the many different aspects of these problems have been studied by several authors [1-12], which include such pioneering works like that of Moore, of Gill, and of Hennie in particular. The most usual approach to the solution of these problems is to make use of the information contained in the transition table representation of the machine in conjunction with the machine's response tree. A *response tree* is basically a graphical presentation of the results obtained when different input sequences are applied to the machine. The different paths through this tree correspond to the possible input sequences that might be used in an experiment. The *nodes* of this tree correspond to the possible states that the machine can be in, after the application of the input sequences that lead to those nodes. The *level* of a node corresponds to the *length* of the input sequence required to reach the node. A path through a response tree terminates whenever certain termination rules are satisfied. The response tree approach is hence in effect an exhaustive tree search process. In the present paper, instead of using the transition table and also the corresponding response tree of the machine, use is made of the transition matrix representation of the machine and of its higher-order forms to develop an approach that effectively solves the measurement and control problems in synchronous sequential machines. The developed approach is extremely systematic and completely algorithmic, and hence can be very readily implemented on a computer as well.

II. Transition Matrices of Mealy and Moore Machines and Their Higher-Order Forms

Conventionally, a transition matrix is viewed as the mathematical counterpart of the transition diagram of a sequential machine. The use of a transition matrix in the determination of the paths and cycles in the transition diagram, in the classification of machine states and in finding the different submachines, in testing whether a machine is strongly connected or not, in finding the set of equivalent states of a machine, and in machine identification is already well known [3,6-8, 11, 13].

For a v-state machine M, the *transition matrix* is composed of v rows and v columns, and is denoted by [M]. For ease of understanding, it is usual to attach the label of the kth state S_k to the kth row and kth column, and refer to the row and column as *row* S_k and *column* S_k, respectively. The (i,j) entry, that is, the entry common to the ith row and jth column of [M] is b_{ij}, if and only if, there exists an input that takes the machine M from the state S_i to the state S_j in the transition diagram of M, and is zero otherwise. For a Mealy machine M, $b_{ij} = \sum_k (I_N, 0_N)$, where I_N is the present input that takes M from S_i to S_j, and 0_N is its corresponding present output, and hence b_{ij} is simply an input-output pair, the summation being over all such input-output pairs, with the interpretation OR or union for the sum. It is therefore obvious that for a Mealy machine M, the (i,j) entry is the label of the branch (or the branch weight) in the corresponding transition diagram of M that points from node S_i to node S_j, and is in accordance with the conventional definition of transition matrix for a Mealy machine [3,6]. But for a Moore machine M, $b_{ij} = \sum_k (I_N, 0_{N+1})$, where I_N is the present input that takes M from S_i to S_j as before, but 0_{N+1}, its corresponding output, is the next output, the summation being over all such input-output pairs; hence b_{ij} is not simply the label of the branch (or the branch weight) in the corresponding transition diagram of M that points from node S_i to node S_j, as conventionally defined.

Note that the transition matrix as defined in this paper is somewhat different from the transition matrix originally defined by Seshu et al. [8], who considered Moore's model of sequential machines. According to [8], the transition matrix describes the *distribution* of edges of a *given weight* (input) in the transition diagram of a machine M; with each input i (permissible input) of M is associated a transition matrix T^i defined by $T^i = [t^i_{kj}]_{v \times v}$, where $t^i_{kj} = 1$ if the input i takes the state S_k into the state S_j; $t^i_{kj} = 0$ otherwise. T^i is thus a square matrix as usual, and has an order which is the same as the number of states of M. In particular, every row of the transition matrix T^i contains exactly one nonzero entry 1, according to this definition. The transition matrix as defined for a Mealy machine in this paper corresponds in fact to the connection matrix $C = [c_{ij}]_{v \times v}$ as defined originally by Hohn et al. [7], and subsequently by Aufenkamp and Hohn [13], where c_{ij} gives the union of the branch weights, that is, input-output pairs, on all branches that point from node S_i to node S_j in the transition diagram of M, and $c_{ij} = 0$ otherwise. On the other hand, the transition matrix for a Moore machine as defined in this paper corresponds to $C\,\Omega_0$ where $C = \sum_i w_i T^i$ is the connection matrix for a Moore machine, and Ω_0 is the output vector [8] defined by $\Omega_0 = [\omega_1, \omega_2, \ldots, \omega_v]^T$, where T stands for the matrix transpose.

The undernoted theorems follow obviously from the definition of a transition matrix.

Theorem 1: If u is the size of the input alphabet of a machine M, irrespective of whether M is a Mealy machine or a Moore machine, every row in the transition matrix [M] contains exactly u input-output pairs, each pair exhibiting a different input symbol.

Theorem 2: In the transition matrix [M] of a Moore machine M, in any column S_k, the output symbol in the input-output pair of every row is the same, and is the output associated with the state S_k in the corresponding transition diagram or transition table.

Example: Table 1 shows the transition table of a Mealy machine M. The corresponding transition matrix [M] is shown in (1).

Table 1. Mealy Machine M

State \ Input	0	1
S_1	S_2,0	S_1,0
S_2	S_2,1	S_3,1
S_3	S_1,1	S_4,0
S_4	S_3,0	S_1,1

$$[M] = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} (1,0) & (0,0) & 0 & 0 \\ 0 & (0,1) & (1,1) & 0 \\ (0,1) & 0 & 0 & (1,0) \\ (1,1) & 0 & (0,0) & 0 \end{bmatrix} \end{array} \qquad (1)$$

$$[M]^2 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} (11,00) & (10,00)+(00,01) & (01,01) & 0 \\ (10,11) & (00,11) & (01,11) & (11,10) \\ (01,10)+(11,01) & (00,10) & (10,00) & 0 \\ (11,10)+(00,01) & (10,10) & 0 & (01,00) \end{bmatrix} \end{array} \qquad (2)$$

For transition matrices, multiplication is defined in the usual way. Let [A] be a vxv transition matrix with the (i,j) entry x_{ij}, and [B] be a vxv transition matrix with the (i,j) entry y_{ij}. Then [C] = [A][B] is also a vxv matrix of which the (i,j) entry, z_{ij}, is

$$z_{ij} = x_{i1}y_{1j} + x_{i2}y_{2j} + \dots + x_{iv}y_{vj} = \sum_{k=1}^{v} x_{ik}y_{kj}$$

where multiplication of the entries x_{ik} and y_{kj}, each of which is, in general, a sum of input-output pairs, or input-output polynomials according to Aufenkamp and Hohn [13], is associative and distributive with respect to addition but not commutative, with the interpretation AND or, more specifically, concatenation for multiplication, and OR or union for addition. Thus multiplication of transition matrices is the same as that of ordinary matrices, except that the order of the factors in each product $x_{ik}y_{kj}$ must be preserved : $x_{ik}y_{kj}$ is not necessarily equivalent to $y_{kj}x_{ik}$. Note that the (i,j) entry z_{ij} of [C] vanishes if certain of the x_{pq}'s or y_{rs}'s are zero. Assume now that both [A] and [B] represent the transition matrix [M] of a v-state sequential machine. Then the matrix [C] is simply equivalent to $[M]^2$ with the (i,j) entry, to be denoted by b_{ij}^2, given by

$$b_{ij}^2 = \sum_{k=1}^{v} b_{ik}b_{kj} \cdot$$

As in the above, b_{ij}^2 vanishes if some of the b_{pq}'s are zero. $[M]^2$ is called a *second-order transition matrix*. Thus for a v-state sequential machine M, a second-order transition matrix is denoted by $[M]^2$ and is composed of v rows and v columns, which are labeled as in [M]. In general, the (i,j) entry of the *rth-order transition matrix*, denoted by $[M]^r$, is given as

$$b_{ij}^r = \sum_{k1,k2,\dots,k(r-1)=1}^{v} b_{ik1}b_{k1k2}\cdots b_{k(r-1)j}$$

which, as usual, vanishes if some of the b_{pq}'s are zero. The rows and columns of $[M]^r$ are also labeled as in [M]. Notice that $[M]^1$ is simply [M].

The following theorems are evident [13].

Theorem 3 : The (i,j) entry of $[M]^2$ gives all input-output sequences $(\overline{I}_k, \overline{0}_m)$ of length two, such that $\overline{I}_k$ takes M from the state S_i to the state S_j passing through some intermediate state S_n, not necessarily distinct from S_i or S_j, producing the corresponding $\overline{0}_m$.

Theorem 4 : The (i,j) entry of $[M]^r$ gives all input-output sequences $(\overline{I}_k, \overline{0}_m)$ of length r, such that $\overline{I}_k$ takes M from the state S_i to the state S_j, passing through r-1 intermediate states, not necessarily all distinct, or distinct from S_i or S_j, producing the corresponding $\overline{0}_m$.

Theorem 5 : For an $[M]^r$, no two of its entries in the same row can have terms that involve the same input sequence $\overline{I}_k$.

Given the kth-order transition matrix $[M]^k$, the next higher-order transition matrix $[M]^{k+1}$ can be formed quite readily. We use the following theorem.

Theorem 6 : $[M]^{k+1} = [M][M]^k$.

Example : The second-order transition matrix $[M]^2 = [M][M]$ corresponding to machine M in Table 1 is shown in (2).

III. Terminal State Identification and Machine Control

Assume that the initial state of a minimal v-state machine M is unknown. We intend to find an input sequence $\bar{I}_k$ such that $\bar{I}_k$ will take the machine M to a known final state which can be uniquely identified by observing the resulting output sequence. In using the transition matrix representations to solve the terminal state identification problem of sequential machines, the undernoted theorem is of prime importance.

Theorem 7: An input sequence $\bar{I}_k$ of length r is a homing sequence for a sequential machine M, if and only if, in the rth-order transition matrix $[M]^r$ of M, whenever $\bar{I}_k$ appears in the entries of two or more columns which are all distinct, the corresponding output sequences of $\bar{I}_k$ in the entries of those columns are all distinct.

Corollary 7.1: An input sequence $\bar{I}_k$ of length r is not a homing sequence for a sequential machine M, if in the rth-order transition matrix $[M]^r$ of M, whenever $\bar{I}_k$ appears in the entries of two or more columns which are all distinct, the corresponding output sequences of $\bar{I}_k$ in the entries of at least two of those columns are nondistinct or identical.

Example: Consider the second-order transition matrix $[M]^2$ of machine M of Table 1 as given in (2). From an inspection of the entries in different rows and columns of $[M]^2$ we see that the input sequence $\bar{I}_k$ =01 of length two is a homing sequence for M.

Once we find a homing sequence $\bar{I}_k$ of length r for the machine M, the terminal state of M can be readily identified from an inspection of the rows and columns of $[M]^r$ where $\bar{I}_k$ appears and then using the appropriate cor---n-dence. Based on the knowledge of the terminal state gained from this part of the experiment, a second input sequence can next be applied to bring the machine M to any desired state. The machine control problem can thus be solved by using an adaptive experiment of *order* two.

There are upper bounds on the lengths of the smallest simple preset homing experiments. The following two theorems [3] are relevant in that context. Before we formally state these theorems, we first introduce a pertinent definition.

The set of states S, one of which is, to the experimenter's knowledge, the initial state of the machine M, is called the *admissible set* of M, and is denoted by A(M). The states in A(M) are called the *admissible states*. Both the homing and diagnosing problems become trivial when A(M) is a singleton (m=1).

Theorem 8: The homing problem for a v-state sequential machine M with m admissible states can always be solved by a simple preset experiment of length L_{hs}, where $L_{hs} \leq (v-1)(m-1)$.

Theorem 9: Let M be a sequential machine in which every pair of states is k-distinguishable. The homing problem for M with m admissible states can always be solved by a simple preset experiment of length L_{hs}, where $L_{hs} \leq k\,(m-1)$.

IV. Synchronization Problem

A sequential machine M is said to possess a synchronizing sequence, if and only if, there exists at least one input sequence $\bar{I}_k$ such that $f(\bar{I}_k,S_i) = S_j$ for all $S_i \epsilon$ S. This signifies that $f(\bar{I}_k,S)$ is a mapping of the state set S onto a single state $S_j \epsilon$ S.

To solve the synchronization problem of sequential machines through the use of transition matrices, consider the following theorem.

Theorem 10: An input sequence $\bar{I}_k$ of length r is a synchronizing sequence for a sequential machine M, if and only if, in the rth-order transition matrix $[M]^r$ of M, $\bar{I}_k$ appears in the entries of all the rows for a particular column j, $1 \leq j \leq v$.

Corollary 10.1: An input sequence $\bar{I}_k$ of length r is not a synchronizing sequence for a sequential machine M, if in the rth-order transition matrix $[M]^r$ of M, there exists at least one row for every column j, $1 \leq j \leq v$, where $\bar{I}_k$ does not appear in the entries.

The following theorems deal with upper bounds on the lengths of synchronizing sequences for sequential machines.

Theorem 11: If a v-state sequential machine M possesses a synchronizing sequence $\bar{I}_k$, then the length of the sequence $\bar{I}_k$ is L_{ss}, where $L_{ss} \leq 2^v-(v+1)$.

For $v \geq 8$, a better bound is provided by the following theorem.

Theorem 12: If a synchronizing sequence $\bar{I}_k$ exists for a v-state sequential machine M, then the length L_{ss} of the sequence $\bar{I}_k$ is at most $v(v-1)^2/2$.

V. Initial State Identification

In solving the initial state identification problem we need to find an input sequence $\bar{I}_k$ such that there exists a unique relationship between the observed output sequence $g(\bar{I}_k,S_i)$, and the unknown initial state of the machine S_i. To solve the diagnosing problem by using transition matrix representations, we make use of the following theorem.

Theorem 13: An input sequence $\bar{I}_k$ of length r is a diagnosing sequence for a sequential machine M with admissible set A(M) = S, the set of states of M, if and only if, in the rth-order transition matrix $[M]^r$ of M, for all of the v rows in each of which $\bar{I}_k$ appears in the entry of some column j, $1 \leq j \leq v$, the corresponding output sequences of $\bar{I}_k$ in the entries of all the columns are distinct.

Corollary 13.1: An input sequence $\bar{I}_k$ of length r is a diagnosing sequence for a sequential machine M with admissible set $A(M) = S_{a1}, S_{a2},...,S_{ak} \subset S$, the set of states of M, if and only if, in the rth-order transition matrix $[M]^r$ of M, for all of the ak rows in each of which $\bar{I}_k$ appears in the entry of some column j, $1 \leq j \leq v$, the corresponding output sequences of $\bar{I}_k$ in the entries of all the columns are distinct.

Corollary 13.2: An input sequence $\bar{I}_k$ of length r is not a diagnosing sequence for a sequential machine M with admissible set $A(M) = S_{a1}, S_{a2},..., S_{ak} \subseteq S$, the set of states of M, if in the rth-order transition matrix $[M]^r$ of M, of the ak rows in each of which $\bar{I}_k$ appears in the entry of some column j, $1 \leq j \leq v$, the corresponding output sequences of $\bar{I}_k$ in at least two of the entries are nondistinct or identical.

Example: Consider the second-order transition matrix $[M]^2$ of machine M of Table 1 as given in (2) again. From an inspection of the entries in its different rows and columns we find that the input sequence $\bar{I}_k = 01$ of length two, which also happens to be a terminal state identification sequence, can be selected as a preset initial state identification sequence or diagnosing sequence for M.

In the following we first define, and discuss certain properties of, a class of sequential machines which definitely possess preset initial state identification sequences.

A sequential machine M = $< I,S,0,f,g >$ is said to be *I_i-mergeable*, if and only if, for each input symbol $I_i \in I$ there exist at least two states $S_i,S_j \in S$ such that $f(I_i,S_i) = f(I_i,S_j)$, and $g(I_i, S_i) = g(I_i, S_j)$. Otherwise, the machine M is called *non-I_i-mergeable*. We now state an important theorem [5].

Theorem 14: The sequential machines which are I_i-mergeable do not possess any diagnosing sequences.

We next state the following results.

Theorem 15: The greatest lower bound on the length of a diagnosing sequence $\bar{I}_k$, $GLB(L_{ds})$, of a v-state sequential machine M is $\lceil \log_w v \rceil$, where w denotes the size of the output alphabet of M.

The upper bounds of the lengths of diagnosing sequences can be stated through the following theorems.

Theorem 16: The class of minimal, non-I_i-mergeable, v-state sequential machines each possesses at least one diagnosing sequence $\overline{I}_k$ whose length L_{ds} is no greater than $v(v-1)/2$ input symbols.

Theorem 17: The diagnosing problem for a v-state sequential machine M with only two admissible states can always be solved by a simple preset experiment of length L_{ds}, where $L_{ds} \leq v-1$.

Theorem 18: The diagnosing problem for a v-state sequential machine M with m admissible states, if at all solvable by simple preset experimentation, is solvable by a simple preset experiment of length L_{ds}, where $L_{ds} \leq (m-1)v^m$.

References

1. T. L. Booth, "Sequential machines and automata theory", John Wiley, New York, NY 1967.
2. E. F. Moore, "Gedanken-experiments on sequential machines", in Automata Studies, C. E. Shannon and J. McCarthy, Editors, Princeton Univ. Press, Princeton, NJ 1956, pp.129-153.
3. A. Gill, "Introduction to the theory of finite-state machines", McGraw-Hill, New York, NY 1962.
4. A. Gill, "State-identification experiments in finite automata", Information and Control, 4, 132-154 (1961).
5. S. C. Lee, "Digital circuits and logic design", Prentice-Hall, Englewood Cliffs, NJ 1976.
6. C. L. Sheng, "Introduction to switching logic", Intext, Scranton, PA 1972.
7. F. E. Hohn, S. Seshu, and D. D. Aufenkamp, "The theory of nets", IRE Transactions on Electronic Computers, EC-6, 154-161 (1957).
8. S. Seshu, R. E. Miller, and G. Metze, "Transition matrices of sequential machines", IRE Transactions on Circuit Theory, CT-6, 5-12 (1959).
9. S. Ginsburg, "On the length of the smallest uniform experiment which distinguishes the terminal states of a machine", Jol. of the Asso. for Computing Machinery, 5, 266-280 (1958).
10. T. N. Hibbard, "Least upper bounds on minimal terminal state experiments for two classes of sequential machines", Jol. of the Asso. for Computing Machinery, 8, 601-612 (1961).
11. C. L. Sheng and S. R. Das, "On identification of synchronous sequential machines", Automatica, 8, 357-360 (1972).
12. F. C. Hennie, "Fault detecting experiments for sequential circuits", in Proc. of the Fifth Annual Symposium on Switching Circuit Theory and Logical Design, Princeton, NJ 1964 , pp. 95-110.
13. D. D. Aufenkamp and F. E. Hohn, "Analysis of sequential machines", IRE Transactions on Electronic Computers, EC-6, 276-285 (1957).