SEL 75-036

DETERMINISTIC SEQUENTIAL NETWORKS UNDER RANDOM CONTROL

by

Sandor Várszegi

Technical Report No. 97

September 1975

The author is with the Computer and Automation Institute of the Hungarian Academy of Sciences, Budapest, Hungary. This work was performed while the author was a visiting scholar under the Inter-Academy Exchange Program at the Digital Systems Laboratory, Stanford University, Stanford, California, and was partially supported by National Science Foundation grant GK-43322. This paper was also submitted to the 1975 IEEETC Special Issue on Fault-Tolerant Computing.

DIGITAL SYSTEMS LABORATORY

STANFORD ELECTRONICS LABORATORIES

STANFORD UNIVERSITY . STANFORD, CALIFORNIA

SEL 75-036

DETERMINISTIC SEQUENTIAL NETWORKS UNDER RANDOM CONTROL

by

Såndor Vårszegi

Technical Report no. 97

September 1975

Digital Systems Laboratory

Departments of Electrical Engineering and Computer Science

Stanford University

Stanford, California

The author is with the Computer and Automation Institute of the Hungarian Academy of Sciences, Budapest, Hungary. This work was performed while the author was a visiting scholar under the Inter-Academy Exchange Program at the Digital Systems Laboratory, Stanford University, Stanford, California, and was partially supported by National Science Foundation grant GK-43322. This paper was also submitted to the 1975 IEEETC Special Issue on Fault-Tolerant Computing.

DETERMINISTIC SEQUENTIAL NETWORKS UNDER RANDOM CONTROL

Sándor Várszegi

Technical Report no. 97

September 1975

Digital Systems Laboratory
Departments of Electrical Engineering and Computer Science
Stanford University
Stanford, California

ABSTRACT

This paper presents a network-oriented approach for the treatment of deterministic sequential networks under random control. Considered are the cases of multinomial, stationary Markov and arbitrary input processes. Probabilities of the state and output processes are directly derived from the primary information of the network and the source. Coded networks are treated using the logic ciruits or Boolean functions. The isomorphism between Boolean and event algebras is made use of, and the probabilities of the response processes are obtained in the form of algebraical probability expressions interpreted over the determining (i.e., input and initial state) minterm or signal joint probabilities.

Key words: Markov process, multinomial process, output behavior, probabilistic model, probability, random control, random testing, sequential network.

Introduction

The configuration in which a deterministic sequential network is controlled by a random source while the network output is monitored and evaluated appears in various aspects in the computer field. Most indebted to the analysis of the configuration is the random testing of sequential networks, in which the network response to a known random input process is utilized to provide diagnostic information. The analysis of the effect of intermittent faults on the network behavior, the reliability analysis of digital networks, the simulation of the operation of mass-data processing networks, the analysis of the operation of a sequential decoder driven by a noisy channel, etc. can also work with the above configuration as a model.

The concentrated study of the behavior of deterministic sequential network under random control is due to Booth, mid-60's. In [1] he investigated the state and output processes excited by Markov or linearly dependent input processes. By other authors, elements of the topic were touched upon in connection with Markov chains, communication theory, etc. From the beginning of the 70's, the topic has also been important for the random testing of sequential networks [2,3,4]. In the majority of the publications the probabilistic treatment of deterministic networks makes use of matrix tools, or derives probabilities while operating over the state graph of the network. At present, the use of a realization-oriented information base for good and faulty networks, and the increase in the size of networks make it desirable to derive probabilities associated with signals (symbols) in the network by using methods which operate over logic circuits or Boolean functions.

It is the aim of this paper to present a new point of view and methods by which the stochastic behavior of a network can be followed from any given point of time, and also, by which the probabilities associated with the network can be directly derived by using the primary information of the network and the source.

In this paper we consider completely specified synchronous sequential networks, and assume that the random source generates exactly one symbol at every time quantum. The results are also valid for asynchronous networks within the control range for which the behavior of the network corresponds to a network of the above type. In part 1 we examine the case of uncoded networks and obtain general relationships which will be satisfied in any particular network realization. We show that for general input sources the determination of the stochastic behavior of the network requires a separate computation for each point of time whose complexity increases with increasing time. However, we also show that for stationary Markov input sources a recursive method with fixed parameters and complexity for each successive point of time can be used. The probabilistic treatment is specialized for coded networks in part 2.

Notation

upper case letters	symbols or events
lower case letters	(Boolean) variables or functions
superscripts	time quantum identifier
subscripts	identifier of an element (of a set)
P(...)	probability of
$\rightarrow$ ($\not\rightarrow$)	projected (not projected) into
$\Rightarrow$ ($\not\Rightarrow$)	enables (does not enable) to derive

1. Case of Uncoded Networks

Sequential networks considered in this paper are assumed to have the Mealy form, defined by the 5-tuple $<X,Y,Z,\delta,\omega>$, where

X,Y,Z	sets of input, state and output symbols, respectively
δ,ω	next-state and output mappings, respectively.

In our discussion, the Mealy model is controlled by a random source, hence the appearance of any symbol from X,Y or Z has to be characterized by a probability parameter assigned to the symbol.

Making use of the 5-tuple, we wish to investigate how the symbol probabilities are related in the model, and also, which probabilities and correlations characterize the stochastic status of the network and the network itself.

First we examine, in general, how mappings and probabilities are related. When doing so, we use the term "sources".

If a set of symbols possesses the property that at every time quantum exactly one symbol appears, the set is said to form a source, while the probability distribution of the symbols is said to describe the state of the source. We will denote the state of the source U: $\{U_i\}$ by $P(U):\{P(U_i)\}$.

In order to characterize the correlation between the appearance of symbols of two sources, U: $\{U_1,U_2,..,U_{N_U}\}$ and V: $\{V_1,V_2,..., V_{N_V}\}$, we can use the set of their joint probabilities,

(1.1) $P(U_iV_j)$, enough for N_uN_v-1 combinations of i and j.

(Note: the reduction-by-one of the number of probabilities appears whenever all of the probabilities sum to 1.) It can be seen. this set of probabilities also describes the state of the composite source $U \times V$.

Independent of whether there exists a directed deterministic relationship between the appearance of the elements U_i and V_j, we can also describe the correlation in a directed form. For example, if $\{U_i\}$ were projected into $\{V_j\}$, in general, we need to assume a projection which is not deterministic, but which can be characterized by a set of probability projection functions

$$(1.2) \quad P(V_j) = \sum_{i=1}^{N_u} k_{ij} P(U_i), \text{ enough for } N_v - 1 \text{ values of } j.$$

The coefficient k_{ij} is equal to the conditional probability $P(V_j | U_i)$.

The characterization forms (1.1) and (1.2) are equivalent. If we associated inputs of an object with U and outputs with V, form (1.1) would characterize the object by the input-output pairs, while form (1.2) by a set of functions. This object could be a memoryless network (stochastic or deterministic).

The symbol joint probability and probability projection characterization types can be easily related as

$$(1.3) \quad P(U_i V_j) = k_{ij} P(U_i)$$

$$(1.4) \quad P(V_j) = \sum_{i=1}^{N_u} P(U_i V_j)$$

For the two-source case, let us suppose a deterministic (one-to-one or many-to-one) projection $\phi: \{U_i\} \rightarrow \{V_j\}$ exists a priori. Then the specific property arises that $k_{ij} \varepsilon \{0,1\}$, $\forall i,j$, namely

$$(1.5) \quad k_{ij} = 1(0) \text{ if by } \phi,\ U_i \rightarrow (\not\rightarrow) V_j$$

Having the knowledge of the probability projection functions (1.2), the deterministic projection can be characterized in an information lossless manner. (One way of restoring the deterministic information is the "freezing", i.e. to set the input probabilities to the permitted combinations of extremes.) From the other side, the coefficients k_{ij} can be determined by merely knowing the properties of the deterministic projection. Hence, we have two projections

$$(1.6)\quad \phi:\{U_i\}\longrightarrow\{V_j\} \text{ and } \phi_p:\{P(U_i)\}\longrightarrow\{P(V_j)\}$$

such that ϕ and ϕ_p cover the same information.

Returning to the Mealy model, we can speak of input, state and output sources X, Y and Z, having a number N_x,N_y and N_z of symbols, and of two probability projections δ_p and ω_p, for which we have

$$(1.7)\quad \delta_p:P(X^n\times Y^n)\longrightarrow P(Y^{n+1})$$

$$(1.8)\quad \omega_p:P(X^n\times Y^n)\longrightarrow P(Z^n)$$

The behavior of the model at $t=t^n$ is determined by the projections (1.7) and (1.8). Since in general

$$(1.9)\quad P(X^n),\ P(Y^n) \not\Longrightarrow P(X^n\times Y^n)$$

we are to conclude

- though $P(X^n)$ and $P(Y^n)$ contain important information, they do not determine $P(Z^n)$,
- according to equation (1.4)

$$(1.10)\quad P(X^n\times Y^n)\Longrightarrow P(X^n),\ P(Y^n),$$

- knowing the 5-tuple

$$(1.11)\quad P(X^n\times Y^n)\Longrightarrow P(Y^{n+1}),\ P(Z^n).$$

Summarizing, $P(X^n\times Y^n)$ can be considered to describe the stochastic status of the network at $t=t^n$, which means, $P(X^n\times Y^n)$ allows us to derive all possible symbol probabilitcs at $t=t^n$. So, the minimum amount of information for determining the network responses $P(Z^n)$ and $P(Y^{n+1})$ is the knowledge of $P(X^n \times Y^n)$, δ_p and ω_p,

We continue with chaining the descriptors of network behavior along the time axis. According to the relation (1.10) we would know the long-term network behavior if we were able to generate the series

$$(1.12) \quad P(X^r \times Y^r),\ P(X^{r+1} \times Y^{r+1}),\ P(X^{r+2} \times Y^{r+2}),\ldots$$

assuming $t=t^r$ is the initial point of time. Having the knowledge of the network and the source

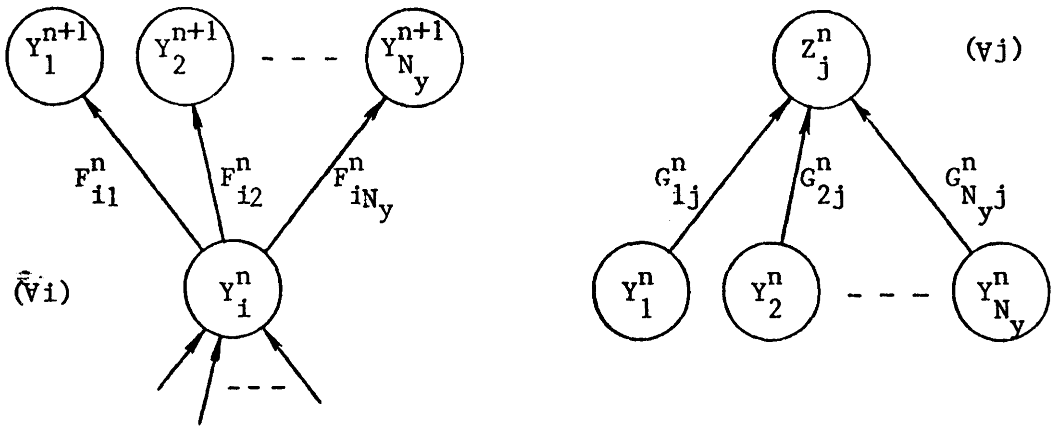
$$(1.13) \quad P(X^n \times Y^n) \Rightarrow P(Y^{n+1}) \text{ and } P(X^n) \Rightarrow P(X^{n+1})$$

however, in general, the relation (1.9) also holds for $t=t^{n+1}$. This implies that the information we used up to now is insufficient for deriving the correlation between P(X) and P(Y). Apparently, we need to know more about the source. In order to determine $P(X^n \times Y^n)$ for arbitrary values of n, we present two methods, called methods of growing and of recursion.

A. The Method of Growing

If we possess the information which exactly describes the network and the input process, and the necessary initial conditions are known, we can follow the stochastic status of the network along the time axis. For deriving the status at $t=t^n$, it is required to know the initial status at $t=t^r$ and the history of the control between t^r and t^n. Such an approach is applicable for any kind of input process.

In the consequent discussion we use the event representation given in Figure 1, where



Event representation for sequential network

Figure 1

Y_i^n, Y_i^{n+1}, Z_j^n	symbol events, in correspondence with previous use
F_{ij}^n	alternative event at $t=t^n$, formed of the appearance of input symbols which can cause a Y_i^n-Y_j^{n+1} state transition
G_{ij}^n	alternative event at $t=t^n$, formed of the appearance of input symbols which make the output symbol Z_j^n appear provided the network is in the state Y_i^n.

The set of symbols belonging to F_{ij}^n/G_{ij}^n can be derived from the mapping δ/ω. If the set is empty, F_{ij}^n/G_{ij}^n equals the impossible event.

For the events, the network establishes the relationships

$$(1.14)\quad Y_i^n\, Y_j^{n+1} = Y_i^n\, F_{ij}^n$$

$$(1.15)\quad Z_j^n = \sum_i Y_i^n\, G_{ij}^n$$

and hence, the following equivalence relations hold

$$(1.16)\quad P(Y_j^{n+1} | Y_i^n) = P(F_{ij}^n | Y_i^n)$$

$$(1.17)\quad P(Z_j^n | Y_i^n) = P(G_{ij}^n | Y_i^n).$$

Equations (1.14) through (1.17) imply the network is deterministic.

Making use of general probability equalities, letting $r+\ell=n$, we obtain

$$(1.18)\quad P(Y_{i_\ell}^{r+\ell}) = \sum_{\forall i_0, i_1, \ldots, i_{\ell-1}} P(Y_{i_0}^r\, F_{i_0 i_1}^r\, F_{i_1 i_2}^{r+1} \cdots F_{i_{\ell-1} i_\ell}^{r+\ell-1})$$

$$(1.19)\quad P(Z_j^{r+\ell}) = \sum_{\forall i_0 i_1, \ldots, i_\ell} P(Y_{i_0}^r\, F_{i_0 i_1}^r\, F_{i_1 i_2}^{r+1} \cdots F_{i_{\ell-1} i_\ell}^{r+\ell-1}\, G_{i_\ell j}^{r+\ell})$$

$$(1.20)\quad P(X_k^{r+\ell}\, Y_{i_\ell}^{r+\ell}) = \sum_{\forall i_0, i_1, \ldots, i_{\ell-1}} P(Y_{i_0}^r\, F_{i_0 i_1}^r\, F_{i_1 i_2}^{r+1} \cdots F_{i_{\ell-1} i_\ell}^{r+\ell-1}\, X_k^{r+\ell})$$

Introducing $\#\{Z_j\} = N_z$, the range of variables in the above equation will be $1 \le \ell < \infty$, $1 \le k \le N_x$, $1 \le i_0, i_1, \ldots, i_\ell \le N_y$, $1 \le j \le N_z$, while r is an integer constant.

It can be seen, that an F-chain in the equations (1.18) through (1.20), for a particular combination of values of $i_0, i_1, \ldots, i_{\ell-1}$, describes a particular wequence of states $Y^r_{i_0}$ through $Y^{r+\ell}_{i_\ell}$ which occurs in ℓ input steps. To an !-step state transition belongs a set of distinct sequences of input symbols which can perform the given transition, and naturally, this set can be smaller than that of all e-length input sequences. The length of the F-chains grows as ℓ increases.

Let $T^{(\ell)}_{i_0 i_\ell}$ denote the event that an ℓ-symbol input sequence starts at $t=t^r$ and results in a state sequence beginning with $Y^r_{i_0}$ and ending with $Y^{r+\ell}_{i_\ell}$. Then we can rewrite equations (1.18) through (1.20) as

$$(1.21)\quad P(Y^{r+\ell}_{i_\ell}) = \sum_{\forall i_0} P(Y^r_{i_0}\; T^{(\ell)}_{i_0 i_\ell})$$

$$(1.22)\quad P(Z^{r+\ell}_j) = \sum_{\forall i_0, i_\ell} P(Y^r_{i_0}\; T^{(\ell)}_{i_0 i_\ell}\; G^{r+\ell}_{i_\ell j})$$

$$(1.23)\quad P(X^{r+\ell}_k\; Y^{r+\ell}_{i_\ell}) = \sum_{\forall i_0} P(Y^r_{i_0}\; T^{(\ell)}_{i_0 i_\ell}\; X^{r+\ell}_k)$$

The term $T^{(\ell)}_{i_0 i_\ell}$ could also be derived as an element of the product matrix equal to $\prod_{i=r}^{r+\ell-1} S^i$, where S^i is the state transition matrix associated with the network, and the elements of which are interpreted at $t=t^i$. Though the arguments of probabilities at the right side of equations (1.21) through (1.23) can be algorithmically generated as ℓ increases, the probabilities themselves are affected by the source, at any point of time,. It is this property that allows us to handle non-stationary input processes.

In order to interpret equations (1.18) through (1.20), it should be noted that the random process generated by the input source is completely defined if and only if from its description, starting with the initial point of time $t=t^r$, the stochastic source-state

(1.24) $$P(X^r_{i_0} \times X^{r+1}_{i_1} \times \ldots \times X^{r+\ell}_{i_\ell}),\ 0 \le \ell,\ 1 \le i_0, i_1, \ldots, i_\ell \le N_x,\ \forall \ell$$

can be generated. This implies the knowledge of all possible symbol sequence probabilities.

The right side of equations (1.18) through (1.20) have a similar structure. so all of these sides could be rewritten as

(1.25) $$\sum P(Y^r_{i_0} \ldots) = \sum P(\ldots | Y^r_{i_0})\ P(Y^r_{i_0})$$

Hence, the behavior of the Mealy model can only be well-defined if all of the sequence probabilities of the input source, conditioned upon the initial state of the model, are determined anyway. Let this topic be analyzed. We have certain conditions imposed by the source/network configuration under discussion:

<u>Logical Conditions 1.1</u>

a) For the network: At every time quantum the network must stay in a single state.

b) For the configuration: The network cannot influence the input source in any way.

c) For the initial state: The uncertainty about the network initial state $Y^r_{i_0}$ is expressed by the probability $P(Y^r_{i_0})$, $\forall i_0$, while $\sum_{\forall i_0} P(Y^r_{i_0}) = 1$.

d) For the experinrnton the configuration: The network can only behave in N_y different ways at $t=t^r$ (we have the initial state uncertainty combined with single states allowed). We assume that the initial state of the network is not determined or influenced by the input source in any way. We have to think that the overall stochastic behavior of the network, which we describe in the form of the state and output processes, can be decomposed into N_Y different modes of stochastic behavior, in accordance with the possible initial states, and the probability of each mode of behavior to appear should equal that of the corresponding initial state.

The logical conditions lead to the probabilistic condition given in Theorem 1.1.

Theorem 1.1

From Logical Conditions 1.1 follow the unique probabilistic condition that the appearance of any input symbol sequence is stochastically independent of any of the network initial states. This probabilistic condition is mathematically contradiction-free.

***

(Proofs for theorems throughout this paper will be omitted; however, they exist.)

The stochastic independence given in Theorem 1.1 will be assumed to hold from now on. If the stochastic independence is utilized in equations (1.18) through (1.20), which up to now only supposed the network was deterministic, we obtain

$$P(Y^{r+\ell}_{i_\ell}) = \sum_{\forall i_0, i_1, \ldots, i_\ell} P(Y^r_{i_0})\, P(F^r_{i_0 i_1}\, F^{r+1}_{i_1 i_2} \cdots F^{r+\ell-1}_{i_{\ell-1} i_\ell}) \tag{1.26}$$

$$P(Z^{r+\ell}_j) = \sum_{\forall i_0, i_1, \ldots, i_\ell} P(Y^r_{i_0})\, P(F^r_{i_0 i_1}\, F^{r+1}_{i_1 i_2} \cdots F^{r+\ell-1}_{i_{\ell-1} i_\ell}\, G^{r+\ell}_{i_\ell j}) \tag{1.27}$$

$$P(X^{r+\ell}_k Y^{r+\ell}_{i_\ell}) = \sum_{\forall i_0, i_1, \ldots, i_{\ell-1}} P(Y^r_{i_0})\, P(F^r_{i_0 i_1}\, F^{r+1}_{i_1 i_2} \cdots F^{r+\ell-1}_{i_{\ell-1} i_\ell}\, X^{r+\ell}_k) \tag{-1.28}$$

In view of equations (1.26) through (1.28), we have an explicit solution for the network behavior along the time axis. Knowing the initial state distribution and the input source sequence probabilit'ies, we are given the possibility of computing the network behavior by using a mapping-typed network representation. It should be seen that it is enough to compute $P(X^n \times Y^n)$

according to equation (1.28), while $P(Z^n)$ and $P(Y^n)$ flow in a "static" way. When deriving the left side of the above equations, as ℓ increases, our computational work also increases. In the next section we show that for stationary Markov input processes, the amount of work could be moderate, and lessens as the order of the Markov process becomes smaller.

Equations (1.26) and (1.27) also allow us to describe the state and output processes generated by the network. For the state symbol sequences we have

$$(1.29) \qquad Y^{r+\ell}_{i_\ell} Y^{r+\ell+1}_{i_{\ell+1}} \cdots Y^{r+m}_{i_m} = Y^{r+\ell}_{i_\ell} F^{r+\ell}_{i_\ell i_{\ell+1}} \cdots F^{r+m-1}_{i_{m-1} i_m}$$

which leads to

$$(1.30) \qquad P(Y^{r+\ell}_{i_\ell} Y^{r+\ell+1}_{i_{\ell+1}} \cdots Y^{r+m}_{i_m}) = \sum_{\forall i_0, i_1, \ldots, i_\ell} P(Y^r_{i_0}) \, P(F^r_{i_0 i_1} F^{r+1}_{i_1 i_2} \cdots F^{r+\ell-1}_{i_{\ell-1} i_\ell} F^{r+\ell}_{i_\ell i_{\ell+1}} \cdots F^{r+m-1}_{i_{m-1} i_m})$$

For the output symbol sequences we can derive

$$(1.31) \qquad Z^{r+\ell}_{j_\ell} Z^{r+\ell+1}_{j_{\ell+1}} \cdots Z^{r+m}_{j_m} = \sum_{\forall i_\ell, i_{\ell+1}, \ldots, i_m} Y^{r+\ell}_{i_\ell} F^{r+\ell}_{i_\ell i_{\ell+1}} \cdots F^{r+m-1}_{i_{m-1} i_m} G^{r+\ell}_{i_\ell j_\ell} G^{r+\ell+1}_{i_{\ell+1} j_{\ell+1}} \cdots G^{r+m}_{i_m j_m})$$

which leads to

$$(1.32) \qquad P(Z^{r+\ell}_{j_\ell} Z^{r+\ell+1}_{j_{\ell+1}} \cdots Z^{r+m}_{j_m}) = \sum_{\forall i_0, i_1, \ldots, i_m} P(Y^r_{i_0}) \, P(F^r_{i_0 i_1} F^{r+1}_{i_1 i_2} \cdots F^{r+m-1}_{i_{m-1} i_m} G^{r+\ell}_{i_\ell j_\ell} G^{r+\ell+1}_{i_{\ell+1} j_{\ell+1}} \cdots G^{r+m}_{i_m j_m})$$

Applying the Method of Growing to the case of multinomial (also known as 0-order Markov) input processes, we could achieve a significant reduction in the complexity of the above formulas caused by the fact that the stochastic independence of input symbols at different points of time is transferable to groups of input symbols. Analyzing equation (1.28), the following important properties arise.

Theorem 1.2

For multinomial input processes, whatever be the network, it holds

$$P(X_k^n Y_i^n) = P(X_k^n)\, P(Y_i^n) \;, \forall i, k, n \;.$$

***

Moreover, we also have

$$P(Y_j^{n+1}) = \sum_i P(Y_i^n)\; P(F_{ij}^n) \tag{1.33}$$

$$P(Z_j^n) = \sum_j P(Y_i^n)\; P(G_{ij}^n) \tag{1.34}$$

B. The Method of Recursion

In case of deterministic control, the derivation of the behavior of the Mealy model is greatly simplified by the fact that the present and next states of the network can be related by means of a finite set of time-invariant difference equations, derivable from the projection 6. For random control, we desire to find a set of difference equations for probabilities, such that they only contain a finite number of time-invariant coefficients, and also, $P(X^n \times Y^n)$ can be computed making use of the value, at $t=t^n$, of variables of the equations. It can be easily seen that the existence of such a set of equations depends on the type of the input process. Without analyzing the case for various input sources, we will present an approach which holds for stationary Markov input processes (combined with no restrictions on the network), and which requires a minimal number of probability variables.

If the above set of probability equations exists, we can consider the equations as the stochastic state equations of the configuration, and their variables, S, as the stochastic state variables. Possessing the state equations, from any point of time, $t=t^n$, we can move forward, $t>t^n$, by performing a recursion for S^{n+1}, or move backward, $t<t^n$, by recursion for S^n, possibly step-by-step up to the initial point of time, $t=t^r$.

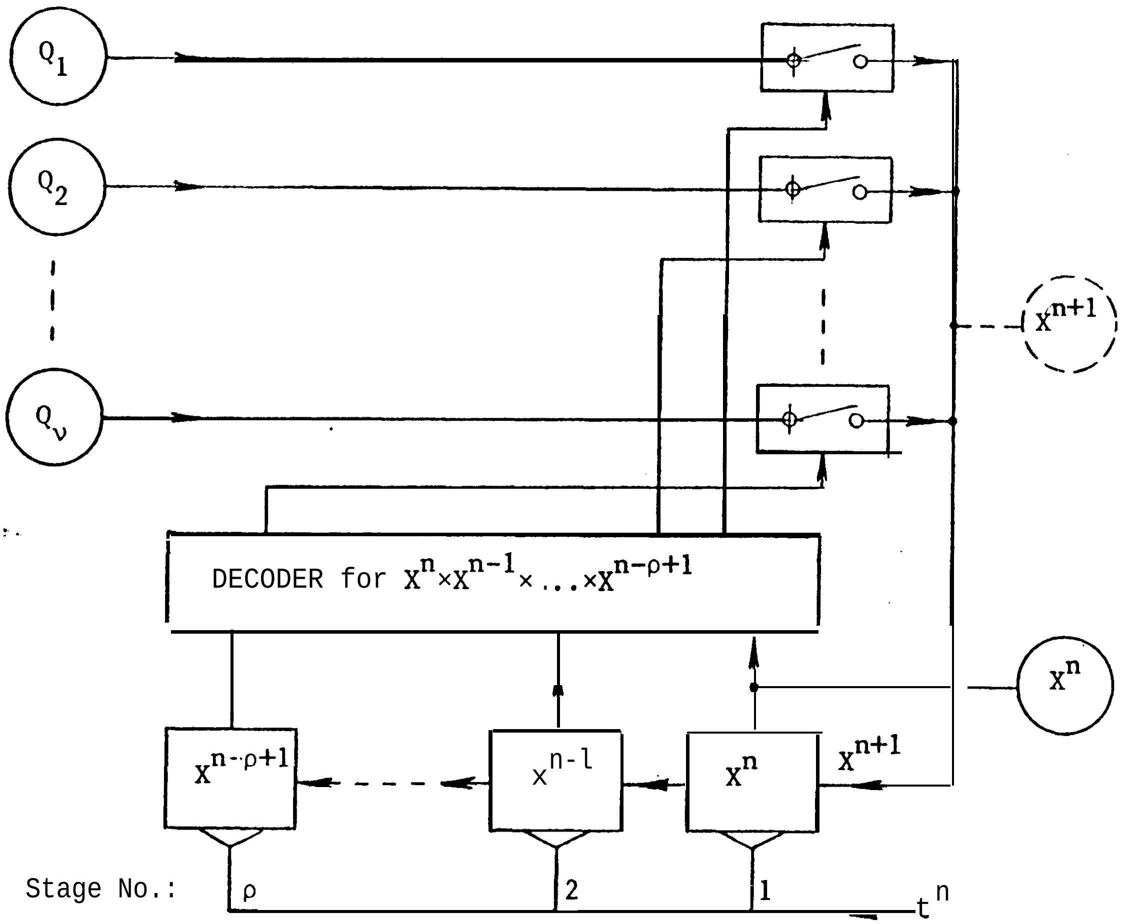
It would be the simplest case, if $P(Y_i^n)$ itself behaved as a state variable. Considering equation (1.34) and Theorem 1.2, we see this holds for a stationary multinomial process. Not restricting the network, the minimum number of state variables in N_Y-1.

Assuming the network is given, making use of general probability equalities we can derive

$$(1.35) \qquad P(Y_i^{n+1}) = \sum_{j=1}^{N_Y} \sum_{k=1}^{N_X} \alpha_{ijk} P(X_k^n | Y_j^n) \, P(Y_j^n)$$

where $\alpha_{ijk} \varepsilon \{0,1\}$, and $\alpha_{ijk} = 1(0)$ if by 6, $X_k^n \times Y_j^n \longrightarrow (\not\longrightarrow) \; Y_i^{n+1}$.
The value of the conditional probabilities in equation (1.35) is not apparent (excluding the case of the multinomial process), because the network more or less memorizes the past control history, and the "forecast " on the present symbol depends on the past input sequences. From this it follows that, even for Markov input processes, we can only have more sophisticated state variables.

The above preferable property of multinomial processes suggests to reduce the case of non-zero-order Markov input processes to that of multinomial processes, in an information lossless manner. To perform the reduction, we construct the universal O-transform network, given in Figure 2, whose structure is applicable for any ρ-order stationary Markov process. (From the other side, the transform network could establish any ρ-order Markov



Universal O-transform network

Figure 2

source, making use of O-order sources.) The network inputs are $Q_1, Q_2, \ldots, Q_\nu$ random sources. The output is the contents of the first stage of the symbol shift register. The network can be identified as a Moore machine. The principle of network operation is that the contents of the shift register uniquely select an input source and the symbol at that source is fed into the shift register. The input sources are multinomial sources, and form a stochastically totally independent group of sources. The Q_i sources have the same set of symbols, however the symbol probabilities are determined source-by-source, by the process to be generated. In rough words, the network will work properly, if the sequence of past symbols stored in the shift register selects the corresponding "forecast" on the next symbol.

The parameters and the probability relations of the O-transform network will be given in Theorem 1.3. The p-order stationary Markov process is defined by

$$P(X_{k_0}^{n} | X_{k_1}^{n-1} X_{k_2}^{n-2} \ldots X_{k_\rho}^{n-\rho} \ldots) = P(X_{k_0}^{r} | X_{k_1}^{r-1} X_{k_2}^{r-2} \ldots X_{k_\rho}^{r-\rho}) \tag{1.36}$$

$$\forall n, \forall k_0, k_1, \ldots k_\rho, \ldots \quad , \quad r\text{:constant}$$

and ρ is the smallest number for which all this is satisfied,

while the operation of a Markov source is determined if in addition we know

$$P(X_{k_1}^{r} X_{k_2}^{r-1} \ldots X_{k_\rho}^{r-\rho+1}) \ , \forall k_1, k_2, \ldots, k_\rho \tag{1.37}$$

because either these probabilities are unique, or they are not unique but given.

<u>Theorem 1.3</u>

Consider the operation (1.37) of a p-order Markov source (1.36). Let it be satisfied that

a) The number of Q_i sources is $\nu=(N_x)^{\rho}$, and the i-th source is labelled by $k_1k_2...k_{\rho}$ so that $i=\sum_{j=1}^{\rho}(N_x)^{j-1}k_j$

b) The set of symbols of Q_i sources is $\{X^{(Q_i)}\} = \{X_1,X_2,...,X_{N_x}\}$, $\forall Q_i$.

c) The symbol probabilities, according to (1.36), at the Q_i sources are

$t=t^n$, $\forall n$: $P(X_k^{(Q_i)}) = P(X_k^r|X_{k_1}^{r-1}\, X_{k_2}^{r-2}\, ...\, X_{k_{\rho}}^{r-\rho})$, $\forall Q_i, k$.

d) The Q_i sources are considered to form a composite source X_{μ}, having the set of symbols $\{X_{j_1}^{(Q_1)} \times X_{j_2}^{(Q_2)} \times ... \times X_{j_{\nu}}^{(Q_{\nu})}\}$.

e) The Q_i sources are defined to be totally uncorrelated, i.e.

$$P(\Pi X_{j_i}^{(Q_i)}) = \Pi P(X_{j_i}^{(Q_i)}),\ \forall j_1,j_2,...,j_{\nu},\ 1\le j_1,j_2,...,j_{\nu}\le N_x$$

and the multiplication is extended over any group of the sources.

f) The contents of the shift register at $t=t^n$ is denoted by

$$Y_{\mu}^n = Y_{k_1}^n \times Y_{k_2}^n\ ... \times Y_{k_{\rho}}^n,$$

and at the same time

$$Y_{\mu}^n = X_{k_1}^n \times X_{k_2}^{n-1} \times ... \times X_{k_{\rho}}^{n-\rho+1}$$

where $X_{k_1}^n$ is the network output symbol at $t=t^n$.

g) At each time $t = t^n$, exactly one source Q_i is selected, where i is determined by Y_{μ}^n by the method of part (a).

h) At some $t=t^r$, the symbol probability distribution of the Y_{μ} source was set to

$$P(Y_{k_1}^r\, Y_{k_2}^r\ ...\ Y_{k_{\rho}}^r) = P(X_{k_1}^r\ X_{k_2}^{r-1}\ ...X_{k_{\rho}}^{r-\rho+1}),\ \forall k_1,k_2,...,k_{\rho}$$

according to condition (1.37).

Then the universal O--transform network, upon the control X_{μ}, for any $t\ge t^r$, will provide an output which is a true realization of the operation of the ρ-order Markov source considered above.

***

We illustrate Theorem 1.3 by means of an example.

Example 1.1

Assume a source has two symbols, X_0 and X_1, which are generated in a second order stationary Markov process. Given the set of conditional probabilities

(1.38) $$P(X_0^r|X_0^{r-1}\ X_0^{r-2}),\ P(X_0^r|\ X_1^{r-1}\ X_0^{r-2}),\ P(X_0^r|X_0^{r-1}\ X_1^{r-2}),\ P(X_0^r|X_1^{r-1}\ X_1^{r-2}),$$

this also determines

(1.39) $$P(X_1^r|X_0^{r-1}\ X_0^{r-2}),\ P(X_1^r|X_1^{r-1}\ X_0^{r-2}),\ P(X_1^r|\ X_0^{r-1}\ X_1^{r-2}),\ P(X_1^r|X_1^{r-1}\ X_1^{r-2})$$

since the "column" sums in this order of writing must equal 1. The operation of the source is defined by the set of probabilities

(1.40) $$P(X_0^r\ X_0^{r-1}),\ P(X_1^r\ X_0^{r-1}),\ P(X_0^r\ X_1^{r-1}),\ P(X_1^r\ X_1^{r-1}).$$

Determine the parameters of the O-transform network.

According to Theorem 1.3, we have a 2-stage shift register whose cells are capable of storing X_0 and X_1. We have 4 random sources which are selected by the contents Y_μ^n: $Y_{\mu_1}^n Y_{\mu_2}^n$ of the shift register. The sources Q_0 through Q_3 are selected by the register contents

(1.41) $$Y_\mu^n:X_0^n\ X_0^{n-1},\ X_1^n\ X_0^{n-1},\ X_0^n\ X_1^{n-1},\ X_1^n\ X_1^{n-1}$$

respectively. The symbols generated by the Q sources are X_0 and X_1, while the symbol probabilities at the sources are

(1.42) $$Q_0:\ P(X_0)\ =\ P(X_0^r|X_0^{r-1}\ X_0^{r-2})$$

(1.43) $$Q_1:\ P(X_0)\ =\ P(X_0^r|X_1^{r-1}\ X_0^{r-2})$$

(1.44) $$Q_2:\ P(X_0)\ =\ P(X_0^r|X_0^{r-1}\ X_1^{r-2})$$

(1.45) $$Q_3:\ P(X_0)\ =\ P(X_0^r|X_1^{r-1}\ X_1^{r-2})$$

and $P(X_1) = 1-P(X_0)$, for each Q source. The initial state probability distribution of the shift register, $P(Y_{\mu_1}^r\ Y_{\mu_2}^r)$, is given by (1.40) so that X^r

and X^{r-1} refer to $Y^r_{\mu_1}$ and $Y^r_{\mu_2}$, respectively.

Theorem 1.4

Given a configuration formed of a p-order stationary Markov input source and a Mealy model. Defined the operation of the input source. Performed the p-to-0 reduction according to Theorem 1.3, assuming $t=t^r$ is the initial point of time an experiment begins on the configuration. $P(Y^r)$ and $P(Y^r_\mu)$ are defined to be stochastically independent of each other and Theorem 1.1 holds for $P(Y^r \times Y^r_\mu)$. Then

a) upon the effect of the replacement, the state and output processes of the Mealy model remain unchanged,

b) the stochastic state variables of the configuration are $P(Y^n \times Y^n_\mu)$, enough $N_y (N_x)^\rho - 1$ of them, and this number of variables is an overall minimum, considering the configuration is parametrically unspecified,

c) $P(X^n_{i_0} \; Y^n_j) = \sum_{\forall i_1, i_2, \dots, i_{\rho-1}} P(Y^n_j \; X^n_{i_0} \; X^{n-1}_{i_1} \cdots X^{n-\rho+1}_{i_{\rho-1}})$

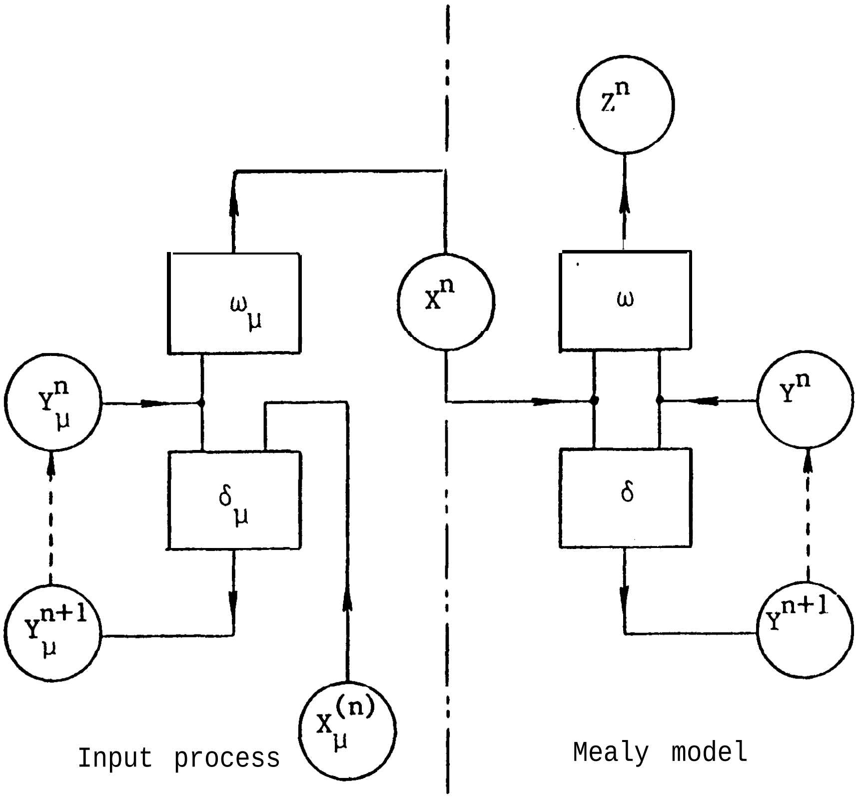
included, the variable left out in point b) equals the one-complement of the sum of the others.

***

The replacing configuration is illustrated in Figure 3.

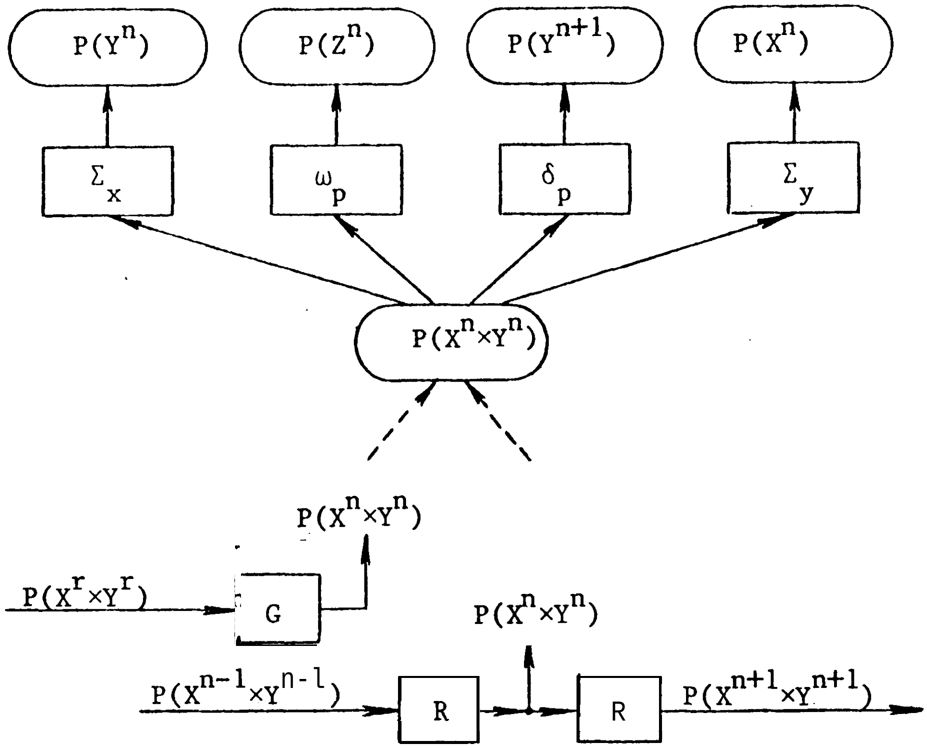
By means of Theorem 1.4 we reduced the treatment of the case of a higher-order Markov source to that of a multinomial source, and simultaneously, we can generate $P(X^n \times Y^n)$ for the original configuration. Not going into details, the knowledge of the Mealy model and the value of the stochastic status variables of the replacing configuration also enable us to derive the state and output sequence probabilities.

Closing the probabilistic treatment of uncoded networks, in Figure 4 we summarized the relationships between the states of the characteristic sources of the configuration.



Replacing configuration

Figure 3



G: Growing R: Recursion

Relationship between states of sources

Figure 4

2. Case of Coded Networks

We wish to examine how the tools of probabilistic treatment can be specialized if the Mealy model is further specified by binary coding for the symbols. We assume, the sets X, Y and Z of coded symbols are binarily complete, i.e. contain $N_x = 2^{n_x}$, $N_Y = 2^{n_y}$ and $N_z = 2^{n_z}$ elements, where n_x, n_y and n_z are integers. (If the case originally were not such, then properly extending the 5-tuple over dummy symbols we could satisfy the assumption, and later, by using zero probabilities, we could disregard the dummies.) The assumption on the completeness of the set of coded symbols is met if we have a completely specified binary network.

Introducing the coding, we have a set of input Boolean functions

$$(2.1) \qquad y_i^{n+1} = f_i(y_1^n,\dots,y_{n_Y}^n \ ; \ x_1^n,\dots,x_{n_X}^n), \quad i: 1,2,\dots,n_y$$

and a set of output Boolean functions

$$(2.2) \qquad z_i^n = g_i(y_1^n,\dots,y_{n_Y}^n; \ x_1^n,\dots,x_{n_X}^n), \quad i: 1,2,\dots,n_z$$

and correspondingly, we speak of the input and output logic.

The isomorphism between the Boolean and event algebras links the probabilistic terms with those we use to describe a coded network. There are two considerable possibilities of using the Boolean calculus to help the probability calculations.

One possibility is to calculate probabilities by using minterms assigned to the symbols. We denote this approach by "minterm probability calculus". The correspondence between symbols and minterms establishes a strict parallelism between the probabilistic treatment of uncoded and coded networks. However, minterms are associated with the canonical form of Boolean functions, and therefore, in general, are not immediately at hand.

The usual matrix-oriented methods of probability calculations operate over minterm probabilities.

The other possibility for calculating probabilities is to make use of the logic variables assigned to the signals of a network. If q is a logic variable, we denote the events that the corresponding signal is at HIGH/LOW level by $q/\overline{q}$ and correspondingly, we introduce the probabilities $P(q)/P(\overline{q})$. Since $P(\overline{q})$ = l-P(q), we shall only speak of "the signal probability" interpreted as P(q). However, even if the stochastic state of a source is to be described, it is not enough to determine the signal probabilities assigned to the lines where the symbols appear. It is also required to know the spatial dependencies,i.e. the probabilistic correlations of the signals at the same point of time. The spatial dependencies can be properly characterized by using conditional or joint probabilities. Since the multiplication is inherent in the Boolean calculus, we shall make use of the joint probabilities, and correspondingly, we can speak of the "joint term probability calculus". Joint terms are composed of uncomplemented Boolean variables, which makes this approach preferable over the minterm approach if we have to derive probabilities for sources of functions as in the case of P(Y) or P(Z). It is another favorable feature of the joint term probability calculus that it explicitly works with the signal probabilities, an important matter of our interest.

For using the joint term probability calculus, it is needed to determine which joint probabilities can provide exhaustive information of the sources. This will be given in Theorems 2.1 through 2.3, in parallel with the probabilities necessary in the minterm calculus. In the theorems we refer to the "complete set of joint probabilities" associated with $\#n_s$ signal lines.

By this term we mean the $(2^{n_s}-1)$-element set of probabilities, composed of the $\#n_s$ signal probabilities and all of the probabilities that logic ones simultaneously appear at pairs, triples,...,n_s-tuple of the lines.

Theorem 2.1

The stochastic state of the source S, whose symbols appear at $\#n_s$ binary lines, can be exhaustively described by any $2^{n_s}-1$ minterm probabilities, or by the complete set of joint probabilities associated with $\#n_s$ signal. lines.

Theorem 2.2

The correlation between the stochastic states of sources U and V, whose symbols appear at $\#n_u$ and $\#n_v$ binary lines, can be exhaustively described by any $2^{n_u n_v}-1$ composite minterm probabilities, obtained by replacing symbols by minterms in the set of probabilities (1.1), or by the complete set of joint probabilities associated with $\#(n_u+n_v)$ signal lines.

Theorem 2.3

The probability projection $\phi_p: \{P(U_i)\} \rightarrow \{P(V_j)\}$, where Ui and Vj are symbols of sources U and V having $\#n_u$ and $\#n_v$ binary lines, can be exhaustively described by $2^{n_v}-1$ minterm probability projection functions obtained by replacing symbols by minterms in functions (1.2), or by the complete set of joint probabilities associated with $\#n_v$ signal lines so that the elements of the latter set are expressed as functions of the U-line (joint term or minterm) probabilities.

***

The equivalent power of characterization possessed by the sets of minterm and joint term probabilities is due to the fact that an element of either set is computable from (several) elements of the other set (see [5], or the methods we later use for combinational networks), moreover, either complete set can be generated in such a way if and only if we know each element of the other set. Hence, without specifying a particular case of use, we generally have to work with the same number of probability parameters in both the minterm and joint term probability calculi, however to generate and to use the joint probabilities is usually more convenient. It is natural that the joint term and minterm probability calculi can be mixed if it provides ease in a particular application.

We continue with specializing the probability relations for the coded Mealy model. As we saw for uncoded networks, if $P(X^n \times Y^n)$ is available, we can perform the probability calculations at $t = t^n$ making use of δ_P and ω_P. In the coded form, this requires the ability of treating multi-output combinational networks under general probabilistic input conditions. Therefore we need to extend the results obtained in [6], and with the latter the reader is assumed to be familiar.

Combinational networks

The input and output combinational networks of a sequential network are usually given by a logic circuit or Boolean functions. For single-output networks, [6] presents two algorithms for calculating the output signal probability. First we extend these algorithms for arbitrary input conditions. We treat two cases:

a) Logic circuit is given.

Algorithm No. 2 [6] results in a function for the output signal probability in terms of the input signal probabilities. The extension of the algorithm is given in Theorem 2.4.

<u>Theorem 2.4</u>

Derive the output probability function by means of Algorithm No. 2 [6], assuming that the network is controlled by spatially independent signals. Write the function so obtained in a parenthesis-free form. If each summand in the parenthesis-free form is replaced as

$$\prod_{i=1}^{n} P(q_i) \rightarrow P(\prod_{i=1}^{n} q_i) \qquad (2.3)$$

where q_i are the input logic variables, the probability function is valid for any input source.

***

As an illustration for Theorem 2.4, if Algorithm No. 2 [6] presented a probability function $P(q_1)[1-P(q_2)]$ then the application of the theorem would result in $P(q_1)-P(q_1 q_2)$.

We can see that the function obtained by applying Theorem 2.4 follows the joint term probability calculus.

After the general form of a particular output probability function is determined, we can perform some reductions if we know that a signal or group of signals is stochastically independent of another signal, or group of signals. Then the terms of the form $P(\Pi q_i)$ can be broken into a product of factors, as $\Pi P(\Pi q_i)$, in correspondence with the existing spatial independencies. (Naturally, we could use these independencies in the first place by grouping the logic variables according to their stochastic dependencies in the parenthesis-free form.) Hence, we can say that the case of a source with spatially independent signals spans a parenthesis-free probability expression which, by properly inserting parentheses, can be adjusted to arbitrary source.

b) Boolean function is given.

If we derive the sum-of-minterms form of a Boolean function, following the instruction of Algorithm No. 1 [6], then by summing the probabilities assigned to the minterms, we have a method for deriving the output probability valid for any input sources. This method follows the minterm probability calculus. In the general case, the minterm probabilities cannot be broken into a product of probabilities associated with signals and signal-complements.

If we have a sum-of-products form of a Boolean function another method of obtaining the output probability function can have advantages. This method follows the joint term probability calculus, and makes use of the associated two-level circuit realization, i.e. the case reduces to that where a logic circuit is given.

For multi-output networks, we have to describe the stochastic state of the source formed by the network output lines. Here we also consider the two cases.

a) Logic circuit is given

Making use of the joint term probability calculus, we need the output signal probabilities and the output joint probabilities. If joint probabilities are to be derived, we can feed the corresponding output signals into an AND gate, so that the output probability of the AND gate equals the required joint probability. If the output minterm probabilities are to be derived, we can insert network-output-driven inverters before AND-ing. In such a way, cases are reduced to the probabilistic treatment of single-output networks.

b) Boolean function is given

Assume the Boolean functions $q_1, q_2, \ldots, q_n$ are associated with the output signals of a multi-output network. For stochastically describing the output source, referring to Theorem 2.1, we need either the probabilities

(2.4) $P(\bar{q}_1\bar{q}_1\ldots\bar{q}_n)$, $P(q_1\bar{q}_2\ldots\bar{q}_n)$, $P(\bar{q}_1q_2\ldots\bar{q}_n),\ldots,P(q_1q_2\ldots q_n)$

or

(2.5) $P(q_1)$, $P(q_2),\ldots,P(q_n)$, $P(q_1q_2)$, $P(q_1q_3),\ldots,P(q_1q_2\ldots q_n)$

Both series of probabilities require the Boolean functions to be placed in the arguments, and the case reduces to that of the single-output networks.

Continuing with the probabilistic treatment of the coded Mealy model at $t = t^n$, we have to derive the stochastic output states of the input and output networks both fed by the signals of the X^n and Y^n sources. We can represent $P(X^n \times Y^n)$ according to Theorem 2.2, and the necessary output probabilities can be derived following the above considerations on combinational networks.

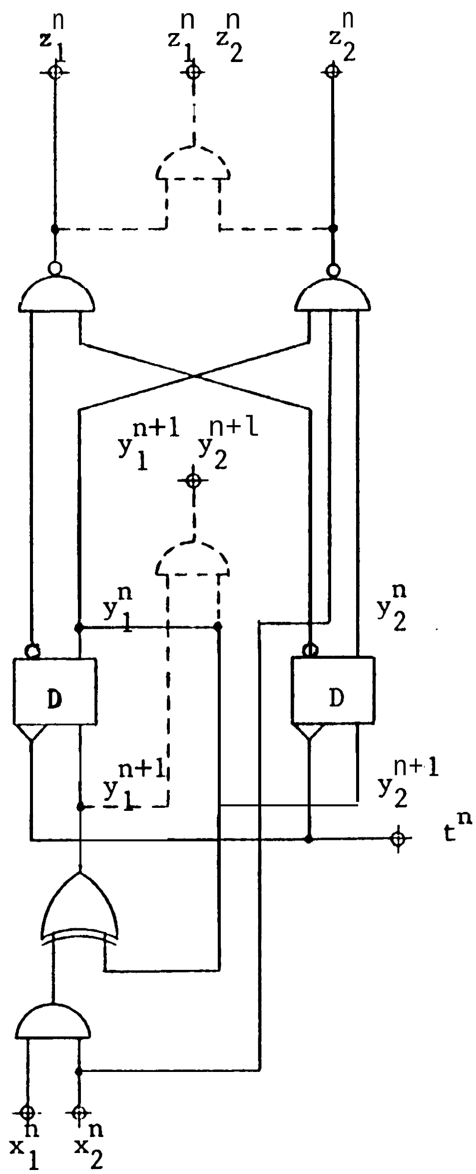
When representing $P(X^n \times Y^n)$ in the joint term probability calculus, we can have independence of signals, with a time-independent validity, if

- the input and state processes are stochastically independent, i.e. the case of a multinomial source,
- some of the input signals may be stochastically independent of each other, determined by the source.

We illustrate the probability calculations at $t = t^n$ by an example.

Example 2.1

Consider the network given in Figure 5. Derive the stochastic states $P(Z^n)$ and $P(Y^{n+1})$, assuming $P(X^n \times Y^n)$ is given.



Example network

Figure 5

For obtaining $P(Z^n)$ and $P(Y^{n+1})$ we use the joint term probability calculus. Also considering the two AND gates of dashed lines in Figure 5, using Algorithm No. 2 [6], eliminating parentheses we have

(2.6) $P(z_1^n) = P(y_1^n) + P(y_2^n) - P(y_1^n)P(y_2^n)$

(2.7) $P(z_2^n) = 1 - P(y_1^n)P(y_2^n)P(x_2^n)$

(2.8) $P(z_1^n z_2^n) = P(y_1^n) + P(y_2^n) - P(y_1^n)P(y_2^n) - P(y_1^n)P(y_2^n)P(x_2^n)$

(2.9) $P(y_1^{n+1}) = P(y_1^n) + P(x_1^n)P(x_2^n) - 2P(y_1^n)P(x_1^n)P(x_2^n)$

(2.10) $P(y_2^{n+1}) = P(y_1^n)$

(2.11) $P(y_1^{n+1} y_2^{n+1}) = P(y_1^n) - P(y_1^n)P(x_1^n)P(x_2^n)$

Merging the arguments in the products we obtain the general form as

(2.12) $P(z_1^n) = P(y_1^n) + P(y_2^n) - P(y_1^n y_2^n)$

(2.13) $P(z_2^n) = 1 - P(y_1^n y_2^n x_2^n)$

(2.14) $P(z_1^n z_2^n) = P(y_1^n) + P(y_2^n) - P(y_1^n y_2^n) - P(y_1^n y_2^n x_2^n)$

(2.15) $P(y_1^{n+1}) = P(y_1^n) + P(x_1^n x_2^n) - 2P(y_1^n x_1^n x_2^n)$

(2.16) $P(y_2^{n+1}) = P(y_1^n)$

(2.17) $P(y_1^{n+1} y_2^{n+1}) = P(y_1^n) - P(y_1^n x_1^n x_2^n)$

If the control source were a (general) multinomial source we could also perform the following factorization in equations (2.12) through (2.17')

(2.18) $P(y_1^n y_2^n x_2^n) \longrightarrow P(y_1^n y_2^n)P(x_2^n)$

(2.19) $P(y_1^n x_1^n x_2^n) \longrightarrow P(y_1^n)P(x_1^n x_2^n)$

and if in addition, the source provided spatially independent signals, we could further factorize as

(2.20) $P(x_1 x_2) \longrightarrow P(x_1)P(x_2)$

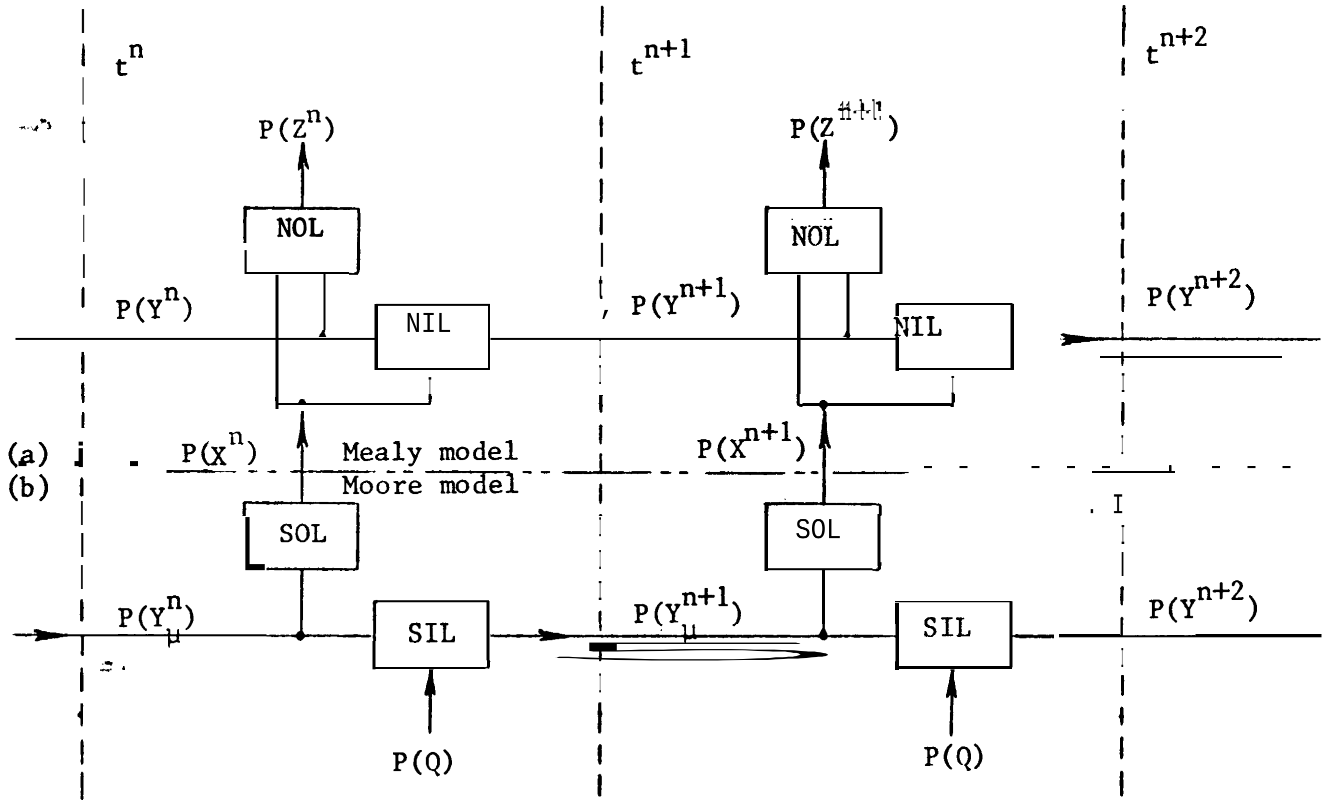
***

In order to derive the stochastic behavior of the network along the time axis, we can make use of the iterative expansion of sequential networks, given in Figure 6a. In such a way, we only have to deal with a combinational network for which the probability calculation methods are possessed.

If the iterative expansion is originated at $t = t^r$, we can derive $P(Z^n)$ and $P(Y^n)$ provided the initial state probability distribution of the network and the stochastic description of the source for the time interval $t^r \leq t \leq t^n$ are known, and Theorem 1.1 is supposed to hold. Then the network to treat consists of the combinational segments for t^r, $t^{r+1}, \ldots, t^n$. In each segment we have network inputs while outputs appear in the segment for t^n. We can also derive any state or output sequence probability within the above time interval, and then we treat combinational network outputs in some consequent segments. This approach corresponds to the Method of Growing for uncoded networks.

If the sequential network were fed by a multinomial input source, $P(X \times Y)$ could be generated for each segment, having the knowledge of the stochastic outputs of the preceding segment. In such a way, for any time interval, we could follow the state and output processes in the iterative network, proceeding segment-by-segment. It means we simultaneously have an iterative expansion for both the deterministic and the stochastic behavior of the network.

For the p-order stationary Markov input sources we can also consider Figure 6b. The introduction of the universal O-transform network makes it possible to treat a periodic combinational structure which has a stochastic input source in each segment, while the sources are stochastically identical multinomial sources, spatially totally independent of each other for any



NIL: network input logic
NOL: network output logic

SIL: source input logic
SOL: source output logic

Iterative expansion

Figure 6

group of the segments. In correspondence with the Method of Recursion we can derive the network state and output processes by treating a minimum time-invariant structure which, in the iterative expansion, appears as the source/network combinational periodic pattern. The set of stochastic state variables of the configuration (see Theorem 1.4) can be identified as the set of input minterm probabilities interpreted over the joint combinational network which is formed of the Moore network output and Mealy network input logics, Figure 6. If we follow the joint term probability calculus, another equivalent set of stochastic state variables is obtained, namely, the complete set of joint probabilities associated with the input lines of the above joint combinational network.

Making use of the iterative expansion, the state and output sequence probabilities could be derived by treating the corresponding number of source/network combinational segments. It is apparent that it is the periodic combinational structure and the set of stochastically identical sources that allows us to determine the sequence probabilities for $t \geq t^n$ if we only know the value of the state variables for $t = t^n$.

Conclusion

We have presented a network-oriented approach for the probabilistic treatment of sequential networks. When discussing the case of uncoded networks, we developed the Methods of Growing and of Recursion. The former method is valid for any type of input process, the latter well suited the case of stationary Markov input processes. Introducing the universal O-transform network, we could represent a stationary Markov process by a network with multinomial control. For networks with a stationary Markov input source we have generated a replacing configuration composed of a replacing network and a multinomial input source. The states of the replacing network are the composite states formed of the original network states and the Markov process states while the probabilities of the composite states appeared as the stochastic state variables of the configuration.

For coded networks, we have extended the applicability of known probability-calculation algorithms [6] of combinational networks, and sequential networks became treatable by the iterative expansion. The use of the universal O-transform network allowed us to integrate a stationary Markov process in the iterative expansion, and in this case, we only had to stochastically treat a periodic combinational network pattern. The joint term probability calculus for coded networks proved to be favorable over the minter-m probability calculus, since its terms are easier to generate and have explicit relations with the signals of the network. The case of multinomial control proved to be easy to work with in many aspects throughout this paper.

For computational purposes we did not generate any state graph or matrix, instead we have worked with algebraical probability expressions derived from the primary information of the network and the source.

Results in the paper are unconditionally valid for completely specified synchronous sequential networks. If the operation of an asynchronous network, with a specified input range, corresponds to that of a network of the above type, the results also hold. We can control the range of inputs by suitably specifying the source, namely, input patterns or sequences of them can be prohibited by setting their probabilities to zero. For the sequence elimination we need a properly sophisticated source.

The obtained results can find application, among other fields, in the random testing of sequential networks, e.g. [7].

Acknowledgements

The author is grateful to the Hungarian Academy of Sciences, the National Academy of Sciences (USA) and the Digital Systems Laboratory, Stanford University, Stanford, California, for supporting his research. The author's thanks are due to Professor J. F. Wakerly who read the manuscript and made helpful comments on it.

REFERENCES

[1] T. L. Booth, Sequential Machines and Automata Theory, Mew York: Wiley, 1967.

[2] J. C. Rault, "A Graph Theoretical and Probabilistic Approach to the Fault Detection of Digital Circuits", Dig. 1971, International Symposium on Fault-Tolerant Computing.

[3] R. Tellez-Giron and R. David, "Random Fault Detection in Logical Networks", Dig. of the International Symposium on Discrete Systems, IFAC Riga, USSR, October 1974.

[4] J. J. Shedletsky and E. J. McCluskey, "The Error Latency of a Fault in a Sequential Digital Circuit", Dig. 1975, International Symposium on Fault-Tolerant Computing.

[5] E. Parzen, Modern Probability Theory and Its Applications, New York: Wiley, 1964.

[6] K. P Parker and E. J. McCluskey, "Probabilistic Treatment of General Combinational Networks", IEEE Transactions on Computers, June 1975, pp. 668-670.

[7] S. Vårszegi, "Source Evaluation for the Random Testing of Sequential Circuits", in preparation.

Biography

Sándor Várszegi was born in Budapest, Hungary, on January 7, 1945. He received his diploma in electrical engineering from the Budapest University of Technology. Since 1970 he has been with the Department of Digital Systems, Computer and Automation Institute of the Hungarian Academy of Sciences. He has been engaged with the design of computer controlled digital test equipment and the computer-aided test generation for digital printed circuit boards. From 1971 to 1974 he was responsible for the TESTOMAT project. His current research interests include fault diagnosis of digital systems, design of digital systems, swithcing and automata theory.