A PROBABILISTIC APPROACH TO THE WORKING PHASE OF A TWO-MODE THRESHOLD LEARNING PROCESS

by

MONDY INGHEL

Submitted to the Department of Electrical Engineering in partial fulfilment of the requirements for the degree of Master of Science.

Department of Electrical Engineering
Faculty of Pure and Applied Science,
The University of Ottawa,
Ottawa, Canada.

1967

ABSTRACT

To keep the performance above some specified minimum level, train-work cycles have been proposed for a two-mode threshold learning process. While during the training phase the threshold is changed according to the correct answer, during the working phase it is kept fixed. Since there is a degradation in the performance index during this last phase a better approach was sought.

The approach to be considered in the present thesis is probabilistic. It consists of changing the threshold at a predeterminated rate according to the information provided by the Markov process theory.

The method leads to a constant performance index. This is proved with the aid of the z-transform method. This index is not always higher than the one obtained with a fixed threshold, depending on the specific characteristics of the information source and channel statistics.

A computer program is proposed for finding out which method leads to a higher index for a given set of parameters.

ACKNOWLEDGMENT

I wish to express sincere thanks to professor G. S. Glinski for his guidance and supervision. I am especially grateful for the suggestions, comments and advice that made possible the present thesis.

Since our research is, in some respect, a continuation of Dr. J. Sklansky's work, much credit should be directed to him also.

I am also indebted to Mr. S. G. S. Shiva for his help.

My thanks for financial assistance are due to the Defence Research Board. (Grant No. 5501-45).

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

Much work has been done during the recent years in the field of adaptive and learning systems. The increasing number of papers published shows the growing interest in finding a more general theoretical basis for this type of systems[1,2,3]. Useful applications might be found in such fields as: automatic control, communication systems, reliability control, numerical computation, etc.

In communications, the "threshold learning process" (T. L. P.) has been proposed as a model of adaptive signal detection for radar systems. In fact the T. L. P. is one form of trainable threshold logic. Sklansky[1,4] analyses the T. L. P. using the Markov chain theory as a mathematical tool. His approach is summarized in chapter 2. Those familiar with Sklansky's works will recognize the slight change in the basic model and the somewhat independent method of presentation.

In chapter 3 some basic concepts of Markov chain theory and z-transform method are briefly reviewed, before proceeding with the explanation and analysis of the proposed model.

A comparison of the two approaches, fixed threshold, as assumed by Sklansky, and changing threshold, as proposed in this work is found in chapter 4. In that chapter we prove also that under reasonable conditions both methods lead to the same result. (The equations used for the fixed threshold performance that are not explained here may be found in references 1 and 4). In general, however, the performances will differ. This fact is proved with the aid of a computer program, as explained in chapter 5.

We would like to mention that the model analyzed here will usually represent a part of a more sophisticated system. For example, assuming that the values of the mode to mode transition probabilities are changing and that one may know these values as a function of time, a computer can be used "on-line" to decide which method will lead to a higher performance index. Similarly for other channel parameters. It is in this context that we keep the "working phase" definition for our model.

Figure 1, slightly changed, is taken from reference 5 where the relation between real systems and their possible simulation is discussed. We use the terms "deterministic approach" and "probabilistic approach" with respect to system realization although in both cases probability is used as a mathematical tool. Hence Sklansky's work will be located in square 2 of that figure and ours in square 4. Chapters 4 and 5, especially, are to be read keeping this in mind.

In chapter 6 we point out a possible relation between our analysis and the Information Theory. The background in this field was taken mainly from reference 6.

Finally some conclusions and remarks are given in chapter 7.

We would like to mention that the model analyzed here will usually represent a part of a more sophisticated system. For example, assuming that the values of the mode to mode transition probabilities are changing and that one may know these values as a function of time, a computer can be used "on-line" to decide which method will lead to a higher performance index. Similarly for other channel parameters. It is in this context that we keep the "working phase" definition for our model.

Figure 1, slightly changed, is taken from reference 5 where the relation between real systems and their possible simulation is discussed. We use the terms "deterministic approach" and "probabilistic approach" with respect to system realization although in both cases probability is used as a mathematical tool. Hence Sklansky's work will be located in square 2 of that figure and ours in square 4. Chapters 4 and 5, especially, are to be read keeping this in mind.

In chapter 6 we point out a possible relation between our analysis and the Information Theory. The background in this field was taken mainly from reference 6.

Finally some conclusions and remarks are given in chapter 7.

CHAPTER II

MODEL DESCRIPTION

2.1 Introduction,

The "threshold learning process" (T. L. P.) model is illustrated in Figure 2. It consists of an information source, a channel, a binary source and an observer.

The information source transmits a discrete, stationary, statistically independent sequence $\tilde{u}$ of 0's and 1's. We denote the frequency of occurrence of 0's by ρ, i.e. $P(0's) = \rho$. This implies $P(1's) = 1 - \rho$.

As each digit, 0 or 1, passes through the channel it is mixed with noise and the resulting sequence $\tilde{v}$ will in general be different from the transmitted one.

If the statistics describing the noisy channel are stationary we call the process one-mode T. L. P. The non-stationary process is discussed in section 2.4.

The one mode T. L. P. is characterized by two conditional probability densities:

1) $p_o(v) \equiv p(v/_u = 0)$

2) $p_1(v) \equiv p(v/_u = 1)$

where, e.g., $p(v/_u = 0)$ is to be understood as the probability of a signal v being output from the channel given that a digit 0 was transmitted.

For convenience we assume, as it happens in some communications systems, that a digit 1 corresponds to a "pulse" and a digit 0 to a "no-pulse" being transmitted.

Since, by assumption, our system is synchronous (the same time reference at both the information source and the observer end), the observer will look during each interval T at the received signal v and will have to decide if a pulse, or no-pulse, was transmitted.

Actually the observer compares the received signal v with a threshold level k predetermined by a "feedback policy". The sign detector will decide whether:

$y = 1$ (i.e. pulse if $v-k \geq 0$)

$y = 0$ (i.e. no-pulse if $v-k < 0$)

During the training phase the information and the binary sources emit an identical and properly chosen sequence. No meaningful information is conveyed through the channel during this phase, except that it provides the observer with the possibility of comparing his guess with the correct answer. The "feedback policy" is the mechanism of threshold adjustment as a result of this comparison. The policy used in the present work is called "simple incremental" and it is implemented as follows:

1) Leave the threshold at its resent value if the observer's guess is correct.

2) Move the threshold to the next higher value if a false-alarm was detected.

3) Move the threshold to the next lower value if a false-rest was detected.

Although this policy implies no memory, it proves to be fairly effective. This fact, when compared with its simplicity of to a "no-pulse" being transmitted.

Since, by assumption, our system is synchronous (the same time reference at both the information source and the observer end), the observer will look during each interval T at the received signal $\dot{v}$ and will have to decide if a pulse, or no-pulse, was transmitted.

Actually the observer compares the received signal $\dot{v}$ with a threshold level k predetermined by a "feedback policy". The sign detector will decide whether:

$\dot{y} = 1$ (i.e. pulse if $v-k \geqslant 0$)

$\dot{y} = 0$ (i.e. no-pulse if $v-k < 0$)

During the training phase the information and the binary sources emit an identical and properly chosen sequence. No meaningful information is conveyed through the channel during this phase, except that it provides the observer with the possibility of comparing his guess with the correct answer. The "feedback policy" is the mechanism of threshold adjustment as a result of this comparison. The policy used in the present work is called "simple incremental" and it is implemented as follows:

1) Leave the threshold at its resent value if the observer's guess is correct.

2) Move the threshold to the next higher value if a false-alarm was detected.

3) Move the threshold to the next lower value if a false-rest was detected.

Although this policy implies no memory, it proves to be fairly effective. This fact, when compared with its simplicity of implementation makes the policy worthy of study.

The definitions for false-alarm and false-rest are given in the next section.

2.2 Detection fundamentals.

Statistical signal detection, as treated in communication theory [7,8], is actually one aspect of the more general statistical decision theory.

What the observer of the proposed model does is known in statistics as "hypothesis testing" and this method is due originally to Neyman and Pearson[9].

In its simplest version the observer measures the received signal and on the basis of the outcome of this measurement he has to choose between two hypotheses:

1) H_0 - a no-pulse was transmited

2) H_1 - a pulse was transmited

This special case treated here is called "simple hypothesis testing".

Under the assumption that the conditional probability density functions describing the channel are Gaussian, the problem might be visualized as in Figure 3.

As mentioned in the preceding section, if the received signal is greater than k, the observer decides that a pulse was transmitted. The exact value of k depends upon the criterion used. Whatever this criterion might be, there is always some probability of wrong detection. Using statistical terms we get an "error of first kind" or false-alarm when hypothesis H_1 is chosen - H_o being true.

The value of this probability is:

$$\text{Prob. [pulse detected when no-pulse transmitted]} = \int_k^{\infty} p_o(v)dv$$

The probability of an "error of the second kind" or false-rest, i.e., H_0 decided upon -H_1 being true is:

$$\text{Prob. [no-pulse detected when pulse transmitted]} = \int_{-\infty}^{k} p_1(v)dv$$

2.3 One-mode T.L.P.

For the one-mode T.L.P. each threshold represents a state of the Markov process. In our model no memory is involved and the Markov process is homogeneous[10] (the transition probabilities are time-independent), with the next state of the process depending only on its present state.

For simplicity this process is allowed to have only three states and the corresponding state transition graph is illustrated in Figure 4. It should be noted that p_{ij} $i, j = 1, 2, 3$ represents the probability that the next state is j, given that the present one is i.

Because of the chosen feedback policy no transition may take place directly, i.e. in one step, between states one and three.

To compute the different transition probabilities p_{ij} we notice that the shaded area in Figure 3 represents the conditional probability that the received signal has a value between v and $v + \Delta v$ given that a pulse was transmitted.

Let us assume now, for example, that the present threshold of operation is $k = 2$ and that by comparing his guess with the correct answer the observer finds out that a false rest has occurred. According to our feedback policy the threshold has to be moved to $k = 1$. Therefore the probability of false rest occurring given that the present threshold is $k = 2$ is given by:

$$p_{21} = (1-\rho) \int_{-\infty}^{2} p_1(v)dv$$

Similarly one may proceed to compute each of the transition probabilities p_{ij}, $i \neq j$. To compute p_{ii}, we may use the fact that for a stochastic matrix the sum of all elements in a row must be unity or $p_{ii} = 1 - \sum_j p_{ij}$, $j \neq i$. Otherwise using the direct method this probability may be computed as follows:

$$p_{ii} = \text{Probability of correct guess}$$

$$= \text{Probability [pulse transmitted and } v - k \geq 0]$$

$$+ \text{Probability [no-pulse transmitted and } v - k < 0]$$

$$= \rho \int_{-\infty}^{k} p_o(v)dv + (1-\rho) \int_{k}^{\infty} p_1(v)dv.$$

For simplicity the conditional probability densities are assumed to be constant in the interval between the thresholds with discontinuities occurring at the thresholds as illustrated in Figure 5. Such a model is called "the quantal" T. L. P. The number of thresholds of our model is still restricted to three.

The above restrictions are necessary only if we want to keep the computation to a minimum and yet to obtain significant results.

The notation used in Figure 5 is explained by the following definitions:

$f_0(v) \equiv \rho p_0(v)$

$f_1(v) \equiv (1-\rho)p_1(v)$

$\alpha \equiv$ Amplitude of the central part of $p_0(v)$

$\beta \equiv$ Amplitude of the central part of $p_1(v)$

By using the method described previously to compute the transition probabilities for the model illustrated in Figure 5 and assuming, for simplicity, that $\alpha = \beta$, the following stochastic matrix is obtained:

$$\overline{P} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1-\frac{\rho}{2}(1+\alpha) & \frac{\rho}{2}(1+\alpha) & 0 \\ \frac{1-\rho}{2}(1-\alpha) & \frac{1}{2}(1+\alpha) & \frac{\rho}{2}(1-\alpha) \\ 0 & \frac{1-\rho}{2}(1+\alpha) & 1-\frac{1-\rho}{2}(1+\alpha) \end{pmatrix} \end{array} \qquad 2.3.1$$

It should be noted that under no condition an independent trials process can be obtained. Such a process might occur in the probabilistic model for the working phase (chapter 3).

We define the performance index Z(n) of the T. L. P. as the probability of a correct guess at time n. This index is computed with the aid of the following formula:

$$Z(n) \equiv \overline{r}(o)\,\overline{P}^{\,n}\,\overline{q} \qquad 2.3.2$$

where for a m-state (threshold) process $\overline{r}(o)$ is the vector of initial distribution of those states, $\overline{P}$ is the transition probability matrix and $\overline{q}$ is a m-dimensional column vector whose elements represent the probability of correct guess given that the process is at a given threshold. Notice that since $\overline{r}(o)$ is a m-dimensional row vector, the matrices in (2.3.2) are comformable[11].

One could write (2.3.2) in a more familiar form as:

$$Z(n) \equiv \overline{r}(n)\,\overline{q}$$

where:

$$\overline{r}(n) = \overline{r}(o)\,\overline{P}^{\,n}$$

with this last equation representing the state probability distribution at time n.

It should be noted that Z(n) is a scalar.

2.4 Two-mode T. L. P.

A practical example of a two-mode Markov process is described by E. N. Gilbert[12], who computes the capacity of a burst-noise telephone communication channel. In one mode, called "A", the telephone channel has "good" transmission characteristics and in the other, called "B", it has relatively "bad" ones.

In our analysis no specific meaning is attached to either mode for the sake of generality, but we still leave the statistical parameters to fluctuate between two modes, A and B. As a result, not only each mode is represented by a Markov process, but the two modes are related by the following stochastic matrix:

$$\overline{\Gamma} \equiv \begin{array}{c c} & \begin{array}{cc} A & B \end{array} \\ \begin{array}{c} A \\ B \end{array} & \begin{pmatrix} 1-\delta & \delta \\ \epsilon & 1-\epsilon \end{pmatrix} \end{array} \qquad 2.4.1$$

where δ represents the probability of transition from mode A to B, etc. .

By properly defining the states of this process Sklansky[4] has shown that it is equivalent to a Markov chain with constant transition probabilities, as illustrated in Figure 6.

The states of this new process are not simply the thresholds, as in the case of an one-mode process. Although we have assumed only three thresholds we have now a six state process. The notation used for those states is to be read as follows: The process is in state A1 if the channel is in mode A and the threshold is $k = 1$, etc. .

2.4 Two-mode T. L. P.

A practical example of a two-mode Markov process is described by E. N. Gilbert[12], who computes the capacity of a burst-noise telephone communication channel. In one mode, called "A", the telephone channel has "good" transmission characteristics and in the other, called "B", it has relatively "bad" ones.

In our analysis no specific meaning is attached to either mode for the sake of generality, but we still leave the statistical parameters to fluctuate between two modes, A and B. As a result, not only each mode is represented by a Markov process, but the two modes are related by the following stochastic matrix:

$$\overline{\Gamma} \equiv \begin{array}{c} \\ A \\ \\ B \end{array} \begin{array}{c} \begin{array}{cc} A & \quad B \end{array} \\ \begin{pmatrix} 1-\delta & \delta \\ & \\ \epsilon & 1-\epsilon \end{pmatrix} \end{array} \qquad 2.4.1$$

where δ represents the probability of transition from mode A to B, etc. .

By properly defining the states of this process Sklansky[4] has shown that it is equivalent to a Markov chain with constant transition probabilities, as illustrated in Figure 6 .

The states of this new process are not simply the thresholds, as in the case of an one-mode process. Although we have assumed only three thresholds we have now a six state process. The notation used for those states is to be read as follows: The process is in state A1 if the channel is in mode A and the threshold is $k = 1$, etc. .

The stochastic matrix of the process has the form:

$$\overline{\Gamma}\,\overline{P} = \begin{pmatrix} (1-\delta)\,\overline{A} & \delta\,\overline{B} \\ \epsilon\,\overline{A} & (1-\epsilon)\,\overline{B} \end{pmatrix} \qquad 2.4.2$$

where $\overline{\Gamma}$ is defined by (2.4.1) and

$$\overline{P} \equiv \begin{pmatrix} \overline{A} & \overline{O} \\ \overline{O} & \overline{B} \end{pmatrix}$$

$\overline{A}$ and $\overline{B}$ being the matrices of the constituent one-mode processes and $\overline{O}$ a third order matrix with zero elements.

It should be noted that $\overline{\Gamma}\,\overline{P}$ is not a standard matrix multiplication and has to be taken in the sense stated by the equation (2.4.2).

CHAPTER III

A PROBABILISTIC MODEL FOR THE WORKING PHASE

3.1 Basic concepts.

In his analysis for the working phase of the two-mode process[4] Sklansky states the following theorem:

Theorem 1: "The final success probability in a working phase of any two-mode Markov chain cannot be less than one-half of the success probability at the beginning of the working phase".

This assumption is that once the training phase is terminated with the achievement of a predeterminated performance index, the working phase begins during which the threshold remains fixed and the correct answer is not known to the observer any more. To use the terminology of automatic control theory, one may say that the controller is idle now and not only that it cannot help the plant, by some adaptive action, to improve its performance index, but also has no means to prevent its degradation. This degradation results directly from the fact that, as time passes, the average mismatch between the actual state of the channel and the value of the threshold increases.

To overcome this undesired degradation of performance, an active role was sought for the controller in the working phase as well. One may conceive different tests for the incoming sequences of bits to see if they carry meaningful information and, if not, some corrective measure is taken. Such an example of adaptive communication is the A.R.Q System[13].

The approach to be taken here is to use the information that we have about the constitutent one-mode processes to improve the working phase performance. The assumption is that it is better to switch the thresholds at some predetermined rate rather than to keep the controller at one fixed threshold. The approach is probabilistic and the information used is concerned with the question of how long, on the average, we expect an one-mode process to be in a specified state. To find these averages we base our analysis on the following theorems[14] :

Theorem 2: "If P is a regular transition matrix then

(i) The powers P^n approach a probability matrix A.

(ii) Each row of A is the same probability vector $\alpha = \{a_1, a_2, \ldots, a_n\}$, that is $A = \xi\alpha$.

(iii) The components of α are positive."

Theorem 3: "If P is a regular transition matrix and A and α are as given in theorem 2, then

(i) For any probability vector π, πP^n approaches the vector α as n tends to infinity.

(ii) The vector α is the unique probability vector such that $\alpha P = \alpha$.

(iii) $PA = AP = A$"

Theorem 4: "(The Law of Large Numbers) Consider a regular Markov chain with limiting vector $\alpha = \{a_1, a_2, \ldots a_r\}$.
For any initial vector π,

$$M_\pi [\bar{v}^{(n)}_j] \rightarrow a_j$$

and for any $\epsilon > 0$

$$Pr_\pi [\,|v^{(n)}_j - a_j| > \epsilon\,] \rightarrow 0$$

as n tends to infinity."

The function $v_j^{(n)}$ is, by definition, the fraction of time in the first n steps that the process moves to state S_j and M_π denotes mean for a chain started with the initial vector π. The theorem implies independence of the starting state. Actually we are interested only in the last theorem and theorems 2 and 3 are given to define and explain terms needed to understand the last one.

Notice that α, ξ and ϵ, as used in the above theorems, are not used in the following analysis as such and must not be confused with parameter α of the conditional probability density functions, ξ the performance index defined in (3. 3. 28) and ϵ the probability of transition from mode B to A as defined in (2. 4. 1.).

As yet, we have restricted ourselves to regular Markov chains. Similar theorems can be proved[14] for ergodic chains. An ergodic chain is either regular or cyclic. For a cyclic chain a geometric bound is found in a different way to prove the Law of Large Numbers[14]. A pertinent theorem is given in reference 15. Notice that we are limited in the possibility of periodic behaviour which is obtained for $\alpha = 0$ and $\rho = 1/2$. In this case the column sum of the elements in (2. 3. 1) is also unity and the limiting vector is (1/3, 1/3, 1/3).

In reference 14 the following theorem is proved:

Theorem 5: "Every Markov chain with a single ergodic set has a unique probability vector fixed point. This vector has positive components for the ergodic states, and zero for the transient states".

Now if we examine the fundamental matrix of the one mode process given by (2. 3. 1) we see that the only case of ergodic set that may arise is when $\alpha = 1$ and then states 1 and 3 are transient and state 2 is absorbing (trapping). No other possibilities exist, since we cannot allow $\rho = 0$ or $\rho = 1$ which would mean that the information source transmits only pulses or only "no-pulses". In such a case our problem of detection does not exist in the first place. Therefore, for $\alpha = 1$, the unique probability fixed point vector will have the form (0, 1, 0). The interpretation that we have for the limiting vector of an ergodic chain does not apply to this one. We simply say that after a large number of steps the process will necessarily be trapped in state 2. This is equivalent to keeping the threshold permanently in position $k = 2$.

For an ergodic chain the fixed point probability vector is obtained from the solution of the following matrix equation:

$$\bar{g}\,\bar{P} = \bar{g}$$

where $\bar{P}$ is given by (2.3.1) and

$$\bar{g} \equiv (a_1, a_2, a_3) \qquad 3.1.1$$

or stated otherwise we have to solve the following three simultaneous linear equations:

$$\begin{aligned} -\frac{\rho}{2}(1+\alpha)\,a_1 + \frac{1-\rho}{2}\,a_2 \qquad\qquad &= 0 \\ \frac{\rho}{2}\,a_2 - \frac{1-\rho}{2}(1+\alpha)\,a_3 &= 0 \\ a_1 + a_2 + a_3 &= 1 \end{aligned}$$

The result is:

$$(a_1, a_2, a_3) = \frac{1}{(1-\alpha)+\rho(1-\rho)(3\alpha-1)}\left[(1-\rho)^2(1-\alpha), \rho(1-\rho)(1+\alpha), \rho^2(1-\alpha)\right] \qquad 3.1.2$$

We observe that by setting $\alpha = 1$ we get the absorbing case discussed previously. Also by substituting in equation (3.1.2) $\alpha = 0$ and $\rho = 1/2$, we obtain the limiting vector mentioned already in connection with the possible periodic behaviour .

Special attention must be paid to the fact that because of the particular conditions explained above concerning the matrix $\bar{\bar{P}}$, no recurrent chains may occur (the chain is indecomposable). The only case when an ergodic (closed) set is observed corresponds to the absorbing chain ($\alpha = 1$).

3.2 <u>The z-transform method.</u>

The basis for the use of signal-flow graphs and transform methods to solve stochastic problems with special emphasis on Markov processes was given by Hüggins[16] and Sittler[17].

To avoid some confusion that has arisen in the definition[18] and relation of the z-transform to the Laplace transform, we will use the z-transform for the discrete Markov processes as defined by:

$$f(z) = \sum_{n=0}^{\infty} f(n)\, z^n \qquad 3.2.1$$

and leave the Laplace transform as such for the solution of continuous-time processes[19,20].

The above definition of the z-transform, i.e., the relationship between f(n) and f(z) is unique and it is applicable to Markov processes since the transition probabilities generate geometric sequences, condition required by the definition.

In Appendix I we list some z-transform[19] pairs which are to be used later.

Now let $\bar{r}(n)$ be a row vector of the states distribution probabilities at time n. The set of linear difference equations with constant coefficients describing the process is:

$$\bar{r}(n+1) = \bar{r}(n)\,\bar{P} \qquad 3.2.2$$

where $\bar{P}$ is the stochastic matrix of the process.

In references 17 and 19 it is shown that by taking the z-transform of (3.2.2) the following matrix equation is obtained:

$$\bar{r}(z) = \bar{r}(o)\,(\bar{I} - z\bar{P})^{-1} \qquad 3.2.3$$

where I is the indentity matrix.

In the time domain the probability distribution of states for a Markov chain is given by:

$$\bar{r}(n) = \bar{r}(0)\,\bar{P}^{n} \qquad 3.2.4$$

Let the matrix $\bar{H}(n)$ be the inverse transform of the matrix $(\bar{I} - z\bar{P})^{-1}$.

Now by taking the inverse transform of equation (3.2.3) and comparing it with equation (3.2.4) we have:

$$\bar{P}^{n} = \bar{H}(n) \qquad 3.2.5$$

Hence the z-transform provides us with a method for studying the behaviour of the process, method which proves to be convenient.

The matrix $\bar{H}(n)$ can be expressed as a sum of matrices at least one of which is a stochastic matrix. For a completely ergodic process there is only one such matrix having identical rows and representing the asymptotic state probabilities, which are not a function of time (n). For such a process the stochastic matrix has one characteristic value of 1 and this is equivalent to saying that the determinant of $(\bar{I} - z\bar{P})$ vanishes for $z = 1$. This fact will be used in the next section. Transform pair 7 in Appendix I tells us that in the transform domain each element of this matrix will have the form $p/(1-z)$, where p is the corresponding asymptotic probability. The other matrices of $H(n)$ will depend on n in such a way that,for an ergodic process, each element will vanish as time goes on. The characteristic values in this case are limited to $|z| < 1$, with the exact values depending on the specific process under study.

Another interesting fact, that is used in the following analysis, is that the row sum of elements of a transient (dependent on n) matrix is zero. Matrices having this property are called differential matrices.[19]

3.3 Analysis of the working phase.

In accordance with the explanation given in section 3.1, and taking into account the mode to mode transition probabilities, the Markov process of the proposed working phase will be described by the following matrix:

$$\bar{P} = \begin{pmatrix} \bar{e}' & \bar{b}' \\ \bar{c}' & \bar{f}' \end{pmatrix} \qquad 3.3.1$$

where:

$$\bar{e}' = \bar{U}\ \bar{W}_1 ; \quad \bar{b}' = \bar{U}\bar{W}_4 ; \quad \bar{C}' = \bar{U}\bar{W}_7 ; \quad \bar{f}' = \bar{U}\ \bar{W}_{10} \qquad 3.3.2$$

$$\bar{U} \equiv (1,\ 1,\ 1)^T$$

$$\bar{W}_1 \equiv (p_1, p_2, p_3) \qquad \bar{W}_4 \equiv (p_4, p_5, p_6) \qquad 3.3.3$$

$$\bar{W}_7 \equiv (p_7, p_8, p_9) \qquad \bar{W}_{10} \equiv (p_{10}, p_{11}, p_{12}) \qquad 3.3.4$$

$$p_1 = (1-\delta)a_1 \qquad p_2 = (1-\delta)a_2 \qquad p_3 = (1-\delta)a_3$$

$$p_4 = \delta b_1 \qquad p_5 = \delta b_2 \qquad p_6 = \delta\ b_3 \qquad 3.3.5$$

$$p_7 = \epsilon\, a_1 \qquad p_8 = \epsilon\, d_2 \qquad p_9 = \epsilon a_3$$

$$p_{10} = (1-\epsilon)b_1 \qquad p_{11} = (1-\epsilon)b_2 \qquad p_{12} = (1-\epsilon)b_3$$

ϵ and δ are the mode to mode transition probabilities as defined by (2.4.1) and a_i, b_i, $i = 1, 2, 3$ are respectively the limiting probabilities of the two one-mode processes as given by equation (3.1.2).

The notation used for the $\overline{P}$ matrix in (3.3.1) is for convenience in analysis. A complete form is given in Appendix II.

Notice that although the matrix $\overline{P}$ looks simple this chain is not lumpable[14], fact that could reduce the order of the matrix and enable us to obtain the same conclusions with less computational effort.

To find the matrix $\overline{H}(n)$ we have first to compute its transform $(\overline{I} - z\overline{P})^{-1}$.

For reasons to be explained soon, we denote:

$$(\overline{I} - z\overline{P}) = \begin{pmatrix} \overline{a} & \overline{b} \\ \overline{c} & \overline{d} \end{pmatrix} \qquad 3.3.6$$

where:

$$\overline{a} = \overline{I} - z\overline{e}' \qquad \overline{d} = \overline{I} - z\overline{f}' \qquad 3.3.7$$

$$\overline{b} = -z\overline{b}' \qquad \overline{c} = -z\overline{c}' \qquad 3.3.8$$

Observe that $\overline{I}$ in equation (3.3.6) is a sixth order identity matrix while in (3.3.7) it is a third order one.

It appears difficult to find the inverse of the matrix given by (3.3.6), since it is a sixth order matrix. Nevertheless, by using the method given in reference 11, the fact that instead of thirty-six elements we have only twelve different ones, symmetry, some properties of matrices, and the mentioned properties of the z-transform of stochastic matrices, we will find that our task is quite easy.

The method referred to above makes possible to work with third order matrices, instead of the original sixth order one, to find $(\bar{I} - z\bar{P})^{-1}$. It consists of solving four simultaneous matrix equations. In a way similar to (3.3.6) let us denote:

$$(\bar{I} - z\bar{P})^{-1} = \begin{pmatrix} \bar{A} & \bar{C} \\ \bar{B} & \bar{D} \end{pmatrix} \qquad 3.3.9$$

Now by multiplying respectively both sides of (3.3.6) and (3.3.9) we obtain:

$$(\bar{I} - z\bar{P})(\bar{I} - z\bar{P})^{-1} = \begin{pmatrix} \bar{a} & \bar{b} \\ \bar{c} & \bar{d} \end{pmatrix}\begin{pmatrix} \bar{A} & \bar{C} \\ \bar{B} & \bar{D} \end{pmatrix}$$

The four matrix equations are:

$$\bar{a}\bar{A} + \bar{b}\bar{B} = \bar{I} \quad ; \quad \bar{c}\bar{A} + \bar{d}\bar{B} = \bar{O}$$

$$\bar{a}\bar{C} + \bar{b}\bar{D} = \bar{O} \quad ; \quad \bar{c}\bar{C} + \bar{d}\bar{D} = \bar{I}$$

And the desired matrices are:

$$\bar{A} = [\bar{a} - \bar{b}\bar{d}^{-1}\bar{c}]^{-1} \quad ; \quad \bar{B} = -\bar{d}^{-1}\bar{c}\bar{A}$$

$$\bar{D} = [\bar{d} - \bar{c}\bar{a}^{-1}\bar{b}]^{-1} \quad ; \quad \bar{C} = -\bar{a}^{-1}\bar{b}\bar{D}$$

Notice that only the inversion of matrices $\bar{a}$ and $\bar{d}$ is needed. No such requirement exists for matrices $\bar{b}$ and $\bar{c}$.

To further simplify our task, we maintain that the solution will hold for any number of thresholds of the one mode process. This fact will become clear towards the end of the inversion process. Meanwhile, under the above assumption, we will let the matrices $\bar{a}$, $\bar{b}$, $\bar{c}$, $\bar{d}$, $\bar{A}$, $\bar{B}$, $\bar{C}$ and $\bar{D}$ to be of second order. Notice that the four matrix equations and their solution are not affected by this assumption.

Now instead of inverting the sixth order $(\overline{I} - z\overline{P})$ matrix we will find the inverse of a fourth order $(\overline{I} - z\overline{R})$ matrix where $\overline{R}$ is assumed to have otherwise the same properties as $\overline{P}$.

Hence:

$$(\overline{I} - z\overline{R}) = \begin{pmatrix} 1-r_1 z & -r_2 z & -r_3 z & -r_4 z \\ -r_1 z & 1-r_2 z & -r_3 z & -r_4 z \\ -r_5 z & -r_6 z & 1-r_7 z & -r_8 z \\ -r_5 z & -r_6 z & -r_7 z & 1-r_8 z \end{pmatrix}$$

$$\overline{d}^{-1} = \frac{1}{1-(r_7+r_8)} \begin{vmatrix} 1-r_8 z & r_8 z \\ r_7 z & 1-r_7 z \end{vmatrix}$$

$$\overline{b}\,\overline{d}^{-1}\overline{c} = \frac{\delta z^2}{1-(1-\epsilon)z} \begin{pmatrix} r_5 & r_6 \\ r_5 & r_6 \end{pmatrix}$$

where use has been made of the fact that:

$$r_3 + r_4 = \delta \quad ; \quad r_7 + r_8 = 1 - \epsilon$$

$$(\overline{a}-\overline{b}\overline{d}^{-1}\,\overline{c}) = \begin{pmatrix} 1 & -1 \\ -r_1 z & 1-r_2 z \end{pmatrix} + \frac{\delta z^2}{1-(1-\epsilon)z} \begin{pmatrix} 0 & 0 \\ r_5 & r_6 \end{pmatrix}$$

Now:

$$\overline{A} = \frac{M}{\Delta} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{Mz}{\Delta} \begin{pmatrix} -r_2 & r_2 \\ r_1 & -r_1 \end{pmatrix} + \frac{\delta z^2}{\Delta} \begin{pmatrix} -r_6 & r_6 \\ r_5 & -r_5 \end{pmatrix}$$

where:

$$M \equiv 1 - (1-\epsilon)z$$

$$\Delta \equiv 1 - [\,2-(\epsilon+\delta)\,]\,z + [\,1-(\epsilon+\delta)\,]\,z^2 \;.$$

Since we know that one characteristic value is unity, it follows immediately that:

$$\Delta = (1-z)\ \{1 - [1-(\epsilon+\delta)]z\}$$

Partial fraction expansion will lead to:

$$A_{11} = \frac{1}{1-z} + \frac{[\epsilon r_1 + \delta r_5 - (\epsilon+\delta)]/(\epsilon+\delta)}{1-z} z + \frac{\delta(r_1 - r_5)/(\epsilon+\delta)}{1-[1-(\epsilon+\delta)]z} z \qquad 3.3.10$$

$$A_{12} = \frac{(\epsilon r_1 + \delta r_5)/(\epsilon+\delta)}{1-z} z + \frac{\delta(r_1 - r_5)/(\epsilon+\delta)}{1-[1-(\epsilon+\delta)]z} z \qquad 3.3.11$$

By replacing r_1 and r_5 with r_2 and r_6 in (3.3.10) and (3.3.11) we obtain the elements A_{22} and A_{21} respectively.

Since we have used only the stochastic properties of the one-mode and two-mode processes we can write in a closed form the matrix $\overline{A}$ as a third order matrix:

$$\overline{A} = \frac{1}{1-z}\ \overline{I} - \frac{z}{1-z}\ \overline{I} + \frac{\nu z}{1-z}\ \overline{e}' + \frac{(1-\nu)z}{1-z}\ \overline{c}' + \frac{(1-\nu)z}{1-[1-(\epsilon+\delta)]z}\ (\overline{e}' - \overline{c}') \qquad 3.3.12$$

where:

$$\nu \equiv \frac{\epsilon}{\epsilon+\delta}$$

Because of symmetry, the matrix $\overline{D}$ is similar to matrix $\overline{A}$ where only the transition probabilities have to be changed in accordance with (3.3.1). The result is:

$$\overline{D} = \frac{1}{1-z}\ \overline{I} - \frac{z}{1-z}\ \overline{I} + \frac{\nu z}{1-z}\ \overline{b}' + \frac{(1-\nu)z}{1-z}\ \overline{f}' + \frac{\nu z}{1-[1-(\epsilon+\delta)]z}\ (\overline{f}' - \overline{b}') \qquad 3.3.13$$

Matrices $\overline{B}$ and $\overline{C}$ are:

$$\overline{B} = \frac{\nu z}{1-z}\overline{e'} + \frac{(1-\nu)z}{1-z}\overline{c'} + \frac{\nu z}{1-[1-(\epsilon+\delta)]z}(\overline{c'} - \overline{e'})$$

$$\overline{C} = \frac{\nu z}{1-z}\overline{b'} + \frac{(1-\nu)z}{1-z}\overline{f'} + \frac{(1-\nu)z}{1-[1-(\epsilon+\delta)]z}(\overline{b'} - \overline{f'})$$

Hence:

$$(\overline{I}-z\,\overline{P})^{-1} = \frac{1}{1-z}\overline{I} - \frac{z}{1-z}\overline{I} + \frac{\nu z}{1-z}\begin{pmatrix}\overline{e'} & \overline{b'}\\ \overline{e'} & \overline{b'}\end{pmatrix} + \frac{(1-\nu)z}{1-z}\begin{pmatrix}\overline{c'} & \overline{f'}\\ \overline{c'} & \overline{f'}\end{pmatrix}$$

$$+ \frac{z}{1-[1-(\epsilon+\delta)]z}\begin{pmatrix}(1-\nu)(\overline{e'}-\overline{c'}) & (1-\nu)(\overline{b'}-\overline{f'})\\ \nu\,(\overline{c'}-\overline{e'}) & \nu\,(\overline{f'}-\overline{b'})\end{pmatrix} \quad 3.3.14$$

Here $\overline{I}$ is a sixth order indentity matrix.

As a check, we observe that the last matrix in (3.3.14) will become a zero matrix in the case that $p_i = p_{i+6}$, $i = 1, 2, \ldots 6$ i.e. for an independent trial process.

Using the transform pairs 2, 3, 4, 6 and 7 given in Appendix I for (3.3.14) we obtain the following inverse transform matrix:

$$\overline{H}(n) = \nu\begin{pmatrix}\overline{e'} & \overline{b'}\\ \overline{e'} & \overline{b'}\end{pmatrix} + (1-\nu)\begin{pmatrix}\overline{c'} & \overline{f'}\\ \overline{c'} & \overline{f'}\end{pmatrix} + [1-(\epsilon+\delta)]^{n-1}\begin{pmatrix}(1-\nu)(\overline{e'}-\overline{c'}) & (1-\nu)(\overline{b'}-\overline{f'})\\ \nu\,(\overline{c'}-\overline{e'}) & \nu\,(\overline{f'}-\overline{b'})\end{pmatrix} \quad 3.3.15$$

$$n \geq 1$$

Notice that for $n=1$ we obtain (3.3.1) as expected.

Similarly to (3.1.1) let us denote for the second one-mode process:

$$h \equiv (b_1, b_2, b_3)$$

as the fixed point probability vector and also denote:

$$\bar{G} = \bar{U}\,\bar{g} \quad ; \quad \bar{H} = \bar{U}.\,\bar{h} \qquad 3.3.16$$

Substituting (3.3.16) in (3.3.15) we have:

$$\bar{H}(n) = \begin{pmatrix} \nu\bar{G} & (1-\nu)\bar{H} \\ \nu\bar{G} & (1-\nu)\bar{H} \end{pmatrix} + [1-(\epsilon+\delta)]^{n} \begin{pmatrix} (1-\nu)\,\bar{G} & -(1-\nu)\bar{H} \\ -\nu\bar{G} & \nu\,\bar{H} \end{pmatrix} \qquad 3.3.17$$

$$n \geq 1$$

Since $\bar{g}$ and $\bar{h}$ are stochastic matrices, the first matrix component of $\bar{H}(n)$ is also stochastic and each row of the non-stochastic matrix sums to zero as explained in section 3.2.

We also observe that the stochastic matrix has the form required for a limiting state probability matrix and that the differential matrix will vanish in time.

We will prove next that the performance index of this model is constant.

During the training phase the process is described by equation (2.4.2). Since in general $\bar{\Gamma}\bar{P}$. will converge quite fast to the limiting matrix we may assume that the initial state distribution for the working phase is given by the fixed point probability vector (steady state probabilities) of $\bar{\Gamma}\bar{P}$.

Let us denote by a_{ij} and b_{ij} respectively the transition probabilities of the two third order matrices in (2.4.2) and let the fixed point probability vector be:

$$\bar{X} \equiv (x_1, x_2, x_3, x_4, x_5, x_6) \qquad 3.3.18$$

We have to solve six simultaneous linear equations to find the elements of $\overline{X}$, i.e. to solve the matrix equation:

$$\overline{X} = \overline{\Gamma}\,\overline{P}\,\overline{X} \qquad 3.3.19$$

However we are not interested in computing each limiting probability individually but want only to show that as a direct result of the construction of $\overline{\Gamma}\,\overline{P}$:

$$\sum_{i=1}^{3} x_i = \nu \text{ or } \sum_{j=4}^{6} x_j = 1 - \nu \qquad 3.3.20$$

where one of the equations in (3.3.20) is obvious since $\overline{X}$ is a probability vector.

The first three simultaneous equations implied by (3.3.19) are:

$$(1-\delta) \sum_{i=1}^{3} a_{i1} x_i + \epsilon \sum_{i=1}^{3} a_{i1} x_{i+3} = x_1 \qquad 3.3.21$$

$$(1-\delta) \sum_{j=1}^{3} a_{j2} x_j + \epsilon \sum_{j=1}^{3} a_{j2} x_{j+3} = x_2 \qquad 3.3.22$$

$$(1-\delta) \sum_{k=1}^{3} a_{k3} x_k + \epsilon \sum_{k=1}^{3} a_{k3} x_{k+3} = x_3 \qquad 3.3.23$$

Summing these three equations and rearranging terms we have:

$$\delta \sum_{i=1}^{3} x_i = \epsilon \sum_{j=4}^{6} x_j$$

Now since $\overline{X}$ is a probability vector we obtain:

$$\sum_{i=1}^{3} x_i = \nu \qquad 3.3.24$$

Pre-multiplying the differential matrix given in (3.3.17) with $\overline{X}$ we have:

$$\overline{X} \begin{pmatrix} (1-\nu)\overline{G} & -(1-\nu)\overline{H} \\ -\nu\,\overline{G} & \nu\,\overline{H} \end{pmatrix} = \overline{O}$$

where $\overline{O}$ is a sixth order zero row vector.

Hence the state probability distribution of the proposed process is independent of time and given by:

$$\overline{Y} \equiv \overline{X} \begin{pmatrix} \nu\overline{G} & (1-\nu)\overline{H} \\ \nu\overline{G} & (1-\nu)\overline{H} \end{pmatrix} \qquad 3.3.25$$

Or:

$$\overline{Y} = (\nu\overline{g},\ (1-\nu)\,\overline{h}) \qquad 3.3.26$$

The probability of correct guess at time n for each state is given by:

$$\overline{Q} \equiv (q_1, q_2, q_3, q_4, q_5, q_6)^T \qquad 3.3.27$$

where:

$$q_i = (1-\delta)a_{ii} + \delta b_{ii} \qquad i = 1,2,3$$

$$q_{j+3} = \epsilon a_{jj} + (1-\epsilon)b_{jj} \qquad j = 1,2,3$$

Therefore the performance index (P. I.) as defined in (2. 3. 2) is:

$$\text{P.I.} \equiv \xi = \nu \sum_{i=1}^{3} a_i\, [(1-\delta)a_{ii} + \delta b_{ii}] + (1-\nu) \sum_{j=1}^{3} b_j\, [\epsilon a_{jj} + (1-\epsilon)b_{jj}] \qquad 3.3.28$$

By this we have completed the proof that the performance index of the proposed model is constant and depends only on the conditional probability densities and the mode-to-mode transition probabilities.

Notice that the performance of an one-mode process during the working phase is also constant.

3.4 Some general remarks.

At this stage we would like to discuss the different modes of behavior that our process might show taking into account what was already said in connection with the one-mode process.

First we should note that periodic behaviour is obtained when $\alpha = 0$, $\rho = 1/2$ and $\epsilon = \delta$ ($\nu = 1/2$).

Not as in the case of one mode processes, this one cannot be trapped in one specific state. Although ergodic set might be formed, no recurrent chains are possible (no multiple characteristic values are possible). When one or both of the one-mode processes are absorbing, our model has transient states leading into an ergodic set. When both one-mode processes are absorbing, the proposed model is equivalent to that of keeping the threshold fixed during the working phase. Renumbering the states, the stochastic matrix $\bar{P}$ in this case is:

$$\bar{P} = \begin{pmatrix} \bar{\Gamma} & \bar{O} \\ \bar{\Gamma} & \bar{O} \\ \bar{\Gamma} & \bar{O} \end{pmatrix}$$

where $\bar{\Gamma}$ is defined by (2.4.1) and $\bar{O}$ denotes a 2 x 4 zero matrix.

This matrix shows that we have obtained a closed set, states 1 and 2, representing threshold 2. The other four are transient states. We notice that the first step in the working phase will necessarily move the process to threshold 2 where it will remain indefinitely.

CHAPTER IV

COMPARISON OF THE TWO APPROACHES

4.1 Identical probability densities.

Next we will show that the probabilistic approach is equivalent to the deterministic one (fixed threshold) for the case when the conditional probability densities are identical. This is to be expected since for this particular case the probability of success does not depend upon the mode-to-mode transition probabilities but on the threshold only. Observe that identity of the probability densities implies under our assumption that the parameter α has the same value for the two one-mode processes. The elements of the vector $\overline{Q}$ are now:

$$q_i = q_{i+3} = a_{ii} \qquad i = 1, 2, 3 \qquad 4.1.1$$

Using a different method (dealing with "roots" which are the reciprocals of the characteristic values) to analyse a two-mode chain, Feller[21] shows that the following time-dependence exists:

$$\overline{\Gamma}^n = \frac{1}{\epsilon + \delta}\begin{pmatrix} \epsilon & \delta \\ \epsilon & \delta \end{pmatrix} + \frac{(1 - \epsilon - \delta)^n}{\epsilon + \delta}\begin{pmatrix} \delta & -\delta \\ -\epsilon & \epsilon \end{pmatrix} \qquad 4.1.2$$

Since now the threshold is fixed we have to compute the performance index by means of the following formula :

$$\text{P.I.} = \sum_{i=1}^{3} \overline{R}_i \; \overline{\Gamma}^n \overline{Q}_i \qquad 4.1.3$$

where :

$$\overline{R}_i = (x_i, x_{i+3}) \;;\quad \overline{Q}_i = (q_i, q_{i+3})^T \quad i = 1, 2, 3$$

Now we notice that by postmultiplying the differential matrix of (4.1.2) by Q_i and using (4.1.1) we obtain :

$$\begin{pmatrix} \delta & -\delta \\ -\epsilon & \epsilon \end{pmatrix} \begin{pmatrix} q_i \\ q_{i+3} \end{pmatrix} = (0, 0) \qquad 4.1.4$$

Hence the P.I., as given in (4.1.3), is not a function of time. The contribution of one threshold to the P.I. is :

$$(x_i, x_{i+3}) \begin{pmatrix} \nu & 1-\nu \\ \nu & 1-\nu \end{pmatrix} \begin{pmatrix} q_i \\ q_i \end{pmatrix} = q_i \, (x_i + x_{i+3}) \qquad 4.1.5$$

substituting in (4.1.3) and summing, we have :

$$\text{P.I.} = \sum_{i=1}^{3} a_{ii} \, (x_i + x_{i+3}) \qquad 4.1.6$$

To see that the same performance is obtained with the probabilistic approach we substitute in equation (3.3.28)

$$a_i = b_i \, ; \; a_{ii} = b_{ii} \qquad i = 1, 2, 3$$

obtaining :

$$\xi = \sum_{i=1}^{3} a_{ii} a_i \qquad 4.1.7$$

To show that (4.1.6) and (4.1.7) give the same result we have to prove that :

$$a_i = x_i + x_{i+3} \qquad i = 1, 2, 3 \qquad 4.1.8$$

We will use the same method as in deriving equation (3.3.28), i.e. by showing that (4.1.8) leads to a solution of (3.3.19) for the particular case we are dealing with.

To this end we rewrite equation (3.3.21) and the fourth of the six linear simultaneous equations not mentioned explicitly as yet under the assumption that :

$$a_{ij} = b_{ij} \quad i, j = 1, 2, 3 \qquad 4.1.9$$

and have :

$$(1 - \delta) \sum_{i=1}^{3} a_{i1} x_i + \epsilon \sum_{i=1}^{3} a_{i1} x_{i+3} = x_1$$

$$\delta \sum_{i=1}^{3} a_{i1} x_i + (1 - \epsilon) \sum_{i=1}^{3} a_{i1} x_{i+3} = x_4$$

Summing up these two equations we obtain :

$$\sum_{i=1}^{3} a_{i1} (x_i + x_{i+3}) = x_1 + x_4 \qquad 4.1.10$$

Since $a_{31} = 0$ for any one-mode process, equation (4.1.10) leads to :

$$\frac{a_{21}}{1 - a_{11}} = \frac{x_1 + x_4}{x_2 + x_5} \qquad 4.1.11$$

Making use of (2.3.1), (3.1.2) and 4.1.8) we observe that equation (4.1.11) is actually an identity.

The direct and complete proof is not given because of the lengthy algebraic operations involved.

Another possibility of getting constant performance in the deterministic case is when $\epsilon + \delta = 1$, fact that can be seen from (4.1.2) .

4.2 "Deterministic" Source

It might be relevant to show that one can obtain:

P. I. = 1

or 100% performance under reasonable assumptions. As the title of this section suggests, this will be the case of $\rho = 0$ or $\rho = 1$, i.e. if the information source emits only "no pulses" or only "pulses".

Let us take for example the case $\rho = 0$.

From (2.3.1) we have:

$$p_{11} = 1 \qquad 4.2.1$$

and from equation (3.1.2) :

$$(a_1, a_2, a_3) = (1, 0, 0) \qquad 4.2.2$$

Since equations (4.2.1) and (4.2.2) are independent of α we obtain for the two one-mode processes :

$$a_{11} = b_{11} = 1 \qquad 4.2.3$$

$$a_1 = b_1 = 1 ; a_2 = a_3 = b_2 = b_3 = 0 \qquad 4.2.4$$

Substituting in equation (3.3.28) we obtain :

$$P.I. = \nu \times 1 \times [(1 - \delta) \times 1 + \delta \times 1] + (1 - \nu) \times 1 \times [\epsilon \times 1 + (1 - \epsilon) \times 1]$$
$$= 1 \qquad 4.2.5$$

For the case $\rho = 1$ we obtain, instead of equations (4.2.3) and (4.2.4) :

$$a_{33} = b_{33} = 1 \qquad 4.2.6$$

$$a_3 = b_3 = 1 ; a_1 = a_2 = b_1 = b_2 = 0$$

which leads again to (4.2.5).

It is obvious that this P.I. must be already attained during the training phase. Notice that for $\rho = 0$ equation (2.3.1) becomes :

$$\overline{P} = \begin{pmatrix} 1 & 0 & 0 \\ x & x & 0 \\ 0 & x & x \end{pmatrix}$$

where x stands for transition probability (not 1 or 0) whose actual value is irrelevant for our present purpose.

Using equation (2.4.2) we have :

$$\bar{\Gamma}\bar{P} = \begin{array}{c} \\ A1 \\ A2 \\ A3 \\ B1 \\ B2 \\ B3 \end{array} \begin{array}{c} \begin{array}{cccccc} A1 & A2 & A3 & B1 & B2 & B3 \end{array} \\ \begin{pmatrix} 1-\delta & 0 & 0 & \delta & 0 & 0 \\ x & x & 0 & x & x & 0 \\ 0 & x & x & 0 & x & x \\ \epsilon & 0 & 0 & 1-\epsilon & 0 & 0 \\ x & x & 0 & x & x & 0 \\ 0 & x & x & 0 & x & x \end{pmatrix} \end{array}$$

The notation used for the states is explained in section 2.4 and it is illustrated in Figure 6.

In the last stochastic matrix we observe that states {A1, B1} form a closed set, i.e., once the process has entered one of these states it will never leave them. It is also easy to see that there is a finite probability to enter this set of states and actually it may be shown that it will be entered quite fast. Anyway, we have assumed infinite training and therefore there is no need to find exactly when this will happen.

Since equation (4.2.3) is also valid for this case, and the definition of the vector $\bar{Q}$ given in (3.3.27) does apply for the training phase too, it follows that for either state A1 or B1 the probability of correct guess is 1.

For $\rho = 1$ the same reasoning applies with the difference that now equation (4.2.6) is true and the closed set is {A3, B3}.

Notice that these results are independent of the four statistics characterizing the channel, as should be. It is also possible to deduce the results from the very definition of the model, but they have been explicitly derived to show the consistency with our analysis.

4.3 Asymptotic performance

For the probabilistic approach, the performance at any time is given by (3.3.28).

In the fixed threshold approach, equation (4.1.3) leads to :

$$\text{P.I.} \equiv \lim_{n \to \infty} Z(\infty) = \nu \sum_{i=1}^{3} (x_i + x_{i+3})\, q_i + (1-\nu) \sum_{i=1}^{3} (x_i + x_{i+3})\, q_{i+3} \qquad 4.3.1$$

Let us denote by Z (0) the P.I. at the beginning of the working phase (end of the training phase). Then from the very definition of this index we may write :

$$Z(0) = \sum_{i=1}^{6} x_i\, q_i \qquad 4.3.2$$

With the aid of (4.1.2), (4.1.3), (4.3.1) and (4.3.2) it can be shown[4] that the P.I. dependence on time is :

$$Z(n) = Z(\infty) + (1 - \epsilon - \delta)^n \,[\, Z(0) - Z(\infty)\,] \qquad 4.3.3$$

The last equation shows the degradation of the working phase.

Now if we were to compare the two approaches we would see the difficulties involved. The method suggested is to compare the performances after a long working period. To estimate this we may use, as a reasonable criterion, ξ and $Z(\infty)$. To this end, equations (3.3.28) and (4.3.1) have to be expressed in terms of the elementary parameters of the process : ρ, $\alpha 1$, $\alpha 2$, δ and ϵ. This done, bounds may be found telling us which of the methods is better for a given quintuple of parameters.

Because of the large number of algebraic operations involved (especially for the solution of the six simultaneous linear equations) and the possibility that no closed formula for comparison could be found, a computer program solution is proposed. The utility of such a program becomes more evident when we take into account that with only slight changes it may be used for an n-mode process, case, where the difficulty of computing the threshold distribution becomes evident.

CHAPTER V

COMPUTER PROGRAM

The program is written in the Fortran 2 language[22] and was run on the I.B.M. 1620 computer located at the Computing Center, University of Ottawa.

It consists of the main program called "Performance Computation" and two subroutines :"Solve Z" and "Basic Z". The block diagram of the main program is illustrated in Figure 9. As it can be seen from it and the program itself, the computation is straightforward. The block diagram of the two subroutines are not shown because of their simplicity.

The subroutine "Solve Z" performs the solution of the six simultaneous equations by the Gauss-Jordan method and takes care of special conditions that might arise (possibility of a zero valued "pivot"). Similar programs are quite common in a computer subroutine library.

The second subroutine "Basic Z" computes the elements for the different matrices as given in the text and which are required for the computation of $Z(0)$, $Z(\infty)$ and ξ. Notice that $S(n)$ and $T(n)$ are actually a_i and b_i (see equation (2.4.1)). The third order matrices A and B represent the two one-mode processes and the sixth order matrix C is needed for the computation of the vector X.

As to the main program DET (deterministic) is used to denote $Z(\infty)$ and PROB (probabilistic) for ξ. The arrays S, T and C (vector X) have the same meaning as in the subroutine "Basic Z".

Only two out of the five parameters are varied through a reasonable range of values to show some significant features. In the Program No. 1 these parameters are $\alpha 2$, the amplitude of the central part of $p_o(v)$ for the second one-mode process and ϵ. In other words, changes in the channel statistics only are allowed. In Program No. 2

the influence of a change in a channel statistics $\alpha 2$, and of the information source statistic was investigated. Otherwise both programs are identical.

Looking at the results of Program 1, different possibilities of performance might be seen, including the special cases discussed in section 4.1. The relevant ones are :

1) (Z0 >) DET > PROB e.g. rows : 7, 78, etc.

2) (Z0 >) DET < PROB e.g. rows : 19, 64, etc.

3) (Z0 =) DET = PROB e.g. rows : 37 - 45

Cases 1) and 2) show us that one approach might be better than another depending on the parameter values.

In accordance with the proof given in section 4.1, case 3) shows that equal performance is obtained when the conditional probability densities of the two one-mode processes are identical (in this case $\alpha 1 = \alpha 2 = 0.5$).

Another feature that follows from equation (4.3.3) is observed whenever in the program $\epsilon = 0.9$ (since $\delta = 0.1$, $\epsilon + \delta = 1$). This agrees also with the remark at the end of section 4.1.

A quick look through all the results reveals some general trends of the P.I.'s. For a specific value of ϵ, all three P.I.'s decrease as $\alpha 2$ increases given that $\alpha 2 > 0.5$. For a given value of $\alpha 2$, Z0 and PROB decrease as ϵ increases but the behaviour of DET is more complicated. This last P.I. behaves like the others for $\epsilon < 0.5$ but changes very slightly for $\epsilon > 0.5$. These observations agree with our expectation and remark about a simple formula for comparison of the P.I.'s. Notice that we are talking as yet only about changes of two parameters. In the above conclusions we do not mention especially the case $\alpha 1 = \alpha 2$ which otherwise fits the general trend.

One more thing to be mentioned with respect to this program is that although PROB is a constant, it differs from Z0 since a change in performance occurs during the transfer from the training phase to the working one.

Program No. 2 was run to show additional features. This time we see that although ρ changes through a considerable range of values, we always have PROB > DET i.e. the probabilistic approach is better.

Another reason for running this program is to substantiate what was already said about a closed comparative formula. Only a quick look at the values of P.I.'s for $\rho = 0.1$ and $\rho = 0.2$ is required to see that the patterns found in Program No. 1 become much more complicated. There is no need for trying to find those new ones since otherwise we would not have suggested the use of a computer program.

One technical observation, from many possible, is to be made about the computer's way of working. In this last program we would expect also that for $\alpha 1 = \alpha 2$ to have DET =PROB, according to the proof given in section 4.1. Nevertheless, since different operations are performed for the computation of DET than for PROB the printed results on line 41 differ because of the truncation error. There is no round-off error if not so specified in the program. For the same reason one has to look with more confidence to the first four numbers of the P.I.'s.

It should be noted that the degradation in performance can be much higher than for those computed, as stated in Theorem 1 and explained thereafter in Section 3.1.

The explanation for that part in the program shown in Figure 9 as "Compute : H SI, HSS" or appearing in the main program itself in the 200 series statements, will be found in the next section.

CHAPTER VI

RELATION TO INFORMATION THEORY

6.1 "Information source"

One might take a more philosophic point of view and lump together the information source and the channel of our model and look at them as a "source" for the observer. We however will not go so far, although the principle of such a lumping exists. Instead we will show a result that might be of interest to research workers in the field of Information Theory. The background in this field was taken mainly from reference 6.

First, we will introduce a Markov Information Source.

Let us denote :

Z_t ≡ state of the chain at time t

Γ ≡ the alphabet of the source

W ≡ $(w_1, \ldots\ldots w_r)$ i.e. steady state distribution

X_n ≡ the n-th random variable of a source sequence.

Given:

1) a finite Markov chain $Z_o, Z_1 \ldots\ldots,$

2) a function of whose domain is the set of states of the chain and whose range is a finite set Γ.

Assume :

3) Z_o is chosen in accordance with W, i.e., Prob. $\{Z_o = s_j\} = w_j$.

Then the stationary sequence $X_n = f(Z_n)$ is said to be a Markov information source corresponding to the above three statements.

Now we will define the uncertainty of a source and the Unifilar Source.

Definition 1 : Given an information source $X_o, X_1, \ldots,$ the uncertainty of the source denoted by $H\{X\}$ is defined as :

$$\lim_{n \to \infty} H(X_n / X_o, X_1, \ldots X_{n-1}).$$

Definition 2 : Consider a Markov information source. For each state s_k, let s_{k1}, s_{k2}, s_{kn_k} be the states that can be reached in one step from s_k, that is, the states s_j such that $p_{kj} > 0$. The source is said to be unifilar if for each state s_k, the letters $f(s_{k1})$, ... $f(s_{kn_k})$ are distinct.

Figure 7 illustrates a simple example of unifilar Markov source with :

1) $S = \{s_1, s_2, s_3\}$

2) $\Gamma = \{A, B, C\}$

3) $f(s_1) = A$, $f(s_2) = B$, $f(s_3) = C$

Now that we know formally what is an unifilar Markov source, and having defined the uncertainty of a source, we state the following theorem :

Theorem : The uncertainty of an unifilar Markov source is given by :

$$H\{X\} = \sum_{k=1}^{r} w_k H_k$$

where, by definition, the uncertainty of the state s_k is :

$$H_k = - \sum_{j=1}^{n_k} p_{kj} \log p_{kj}$$

Since we have felt that there is a basic difference between the two stochastic matrices representing the two-mode T.L.P., one for the training phase given by Sklansky and the other, as proposed by us, the uncertainty associated with each one of them, under the assumption that they represent an unifilar source, was computed. The results are listed under HSS and HSI in Appendix III, where HSS represents the uncertainty associated with (2.4.2) and HSI that associated with (3.3.1). It should be noted that always HSI > HSS. Actually this result was tested for hundreds of other possible values. This was easy to perform since it required only one additional DØ loop in the program. The outcome of this last test will not be found in Appendix III for a simple reason ; the computer was asked to print only if HSS > HSI, i.e., to decide by itself and spare us the trouble of running through hundreds of numbers.

The proof of the above finding is not immediate but there is some motivation why it should be done. In this context we would like to stress the fact that the two-mode T.L.P., although a non-stationary process, can be analysed as a stationary one. Is there such a possibility for a non-stationary source? To be more precise : may we devise a model of two sources such that if at time $t = n$ one of them emits symbols then at $t = n + 1$ there is a finite probability that still it emits and some given probability that the second one will come in action?

We believe that this problem should be further investigated since :

1) For a given arbitrary source it is possible to construct a unifilar Markov source whose uncertainty is as close as we wish to that of the given source.

2) In the absence of noise it is easier to communicate using a source with greater uncertainty.

3) In both cases the number of distinct symbols of the source must be the same for the sources to be unifilar.

6.2 Finite-state channel

This section is intended to show how by using the burst noise channel model[12] refered to in section 2.4 a channel matrix having the form given in (3.3.1) is obtained. This fact is relevant in itself and for this reason is briefly discussed here, although it may be found in reference 6, where one more model leading to the same form is included.

The model may be visualized with the aid of Figure 8. The error probability for a digit transmitted through the channel during "Bad" conditions is β, and p_i, q_i, $i = 1, 2$ are as defined on that figure. For the two possible inputs 0 or 1 we assume four possible states : G0, G1, B0, B1, where G and B stands for "Good" or "Bad" channel conditions and 0 or 1 for the previous output. Under these assumptions the following matrices are obtained :

$$\text{Input} = 0 \qquad \begin{array}{c} \\ G0 \\ G1 \\ B0 \\ B1 \end{array} \begin{array}{c} \begin{array}{cccc} G0 & G1 & B0 & B1 \end{array} \\ \left(\begin{array}{cccc} q_1 & 0 & p_1(1-\beta) & p_1\beta \\ q_1 & 0 & p_1(1-\beta) & p_1\beta \\ p_2 & 0 & q_2(1-\beta) & q_2\beta \\ p_2 & 0 & q_2(1-\beta) & q_2\beta \end{array} \right) \end{array}$$

$$\text{Input} = 1 \qquad \begin{array}{c} \\ G0 \\ G1 \\ B0 \\ B1 \end{array} \begin{array}{c} \begin{array}{cccc} G0 & G1 & B0 & B1 \end{array} \\ \begin{pmatrix} 0 & q_1 & p_1\beta & p_1(1-\beta) \\ 0 & q_1 & p_1\beta & p_1(1-\beta) \\ 0 & p_2 & q_2\beta & q_2(1-\beta) \\ 0 & p_2 & q_2\beta & q_2(1-\beta) \end{pmatrix} \end{array}$$

For the result of connecting a source of the type described in the previous section to a channel as described in this section, reference 6 should be consulted.

CHAPTER VII

CONCLUSIONS AND REMARKS

A probabilistic approach to the working phase was shown to lead in many cases to better results than a deterministic one.

Although no rigorous proof was given, the computer program provided in Appendix III will supply the needed answer, i.e., for a given set of statistics will compute the P.I.'s . Notice that with a small change in the program the computer can decide by itself which P.I. is higher, pointing out to the designer, man or machine, the right approach.

For the case when the two constituent densities are identical it was proven that both approaches lead to the same result, constant and equal P.I.'s . Another case, $\epsilon + \delta = 1$, was mentioned when constant P.I. is obtained in the fixed threshold approach.

The designer may prefer to build a probabilistic controller even if its P.I. is slightly less than for the other approach, and this because of the desired constant performance, given that there is no minimum P.I. requirement preventing him from doing so. Notice that the P.I., as a function of time, when $\epsilon + \delta > 1$ is oscillatory, as can be seen from (4.3.3) . This, in many cases, might be highly undesirable.

The computer results show that for some values of the statistics, the working phase P.I. is even higher than that obtained at the end of the training phase. This is a direct result from the fact that there is a change in P.I. during the transfer from the training to the working phase.

Another important fact is that redistribution of the states at the end of the training phase[4] is not necessary in the proposed model.

The controller will not be probabilistic in the true sense of the word but pseudo-random. A possible approach is to run the switching circuit at a much higher rate than the process itself such that sampling will produce the desired "randomness". Following some suggestions[23] the controller might be implemented with the aid of a random noise generator. In any case, the type of probabilistic finite state machine[24] will be very simple because of the proposed model. Observe that only twelve different transition probabilities are involved in the sixth order matrix. This reduction becomes more evident as the number of states increases since, while the number of the matrix elements increases geometrically, the number of relevant ones increases arithmetically only.

We have tried to find a measure, by looking at how fast a regular chain tends to achieve its steady state conditions, to tell us which of the two approaches is better. Although this idea might be useful in other circumstances, no satisfactory results were obtained for our model.

Finally, we want to point out the fact that the performance of our model is independent of time. This has been proven by using the z - transform method. Moreover, this performance is achieved because of the judicious use of the information provided by the theory of the first-order Markov processes.

System realization \ Analysis	Deterministic	Probabilistic
Deterministic	1	2
Probabilistic	3	4

Fig. 1

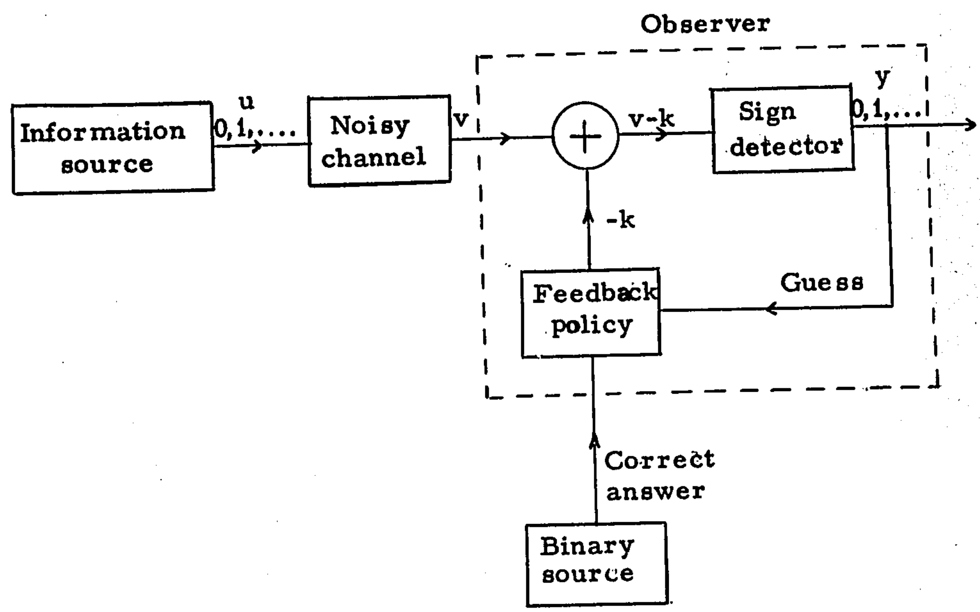

Fig. 2: The T. L. P. Model.

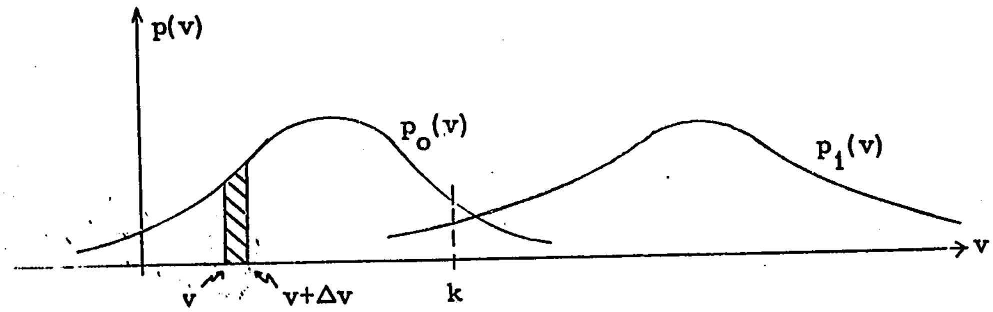

Fig. 3: Conditional probability density functions.

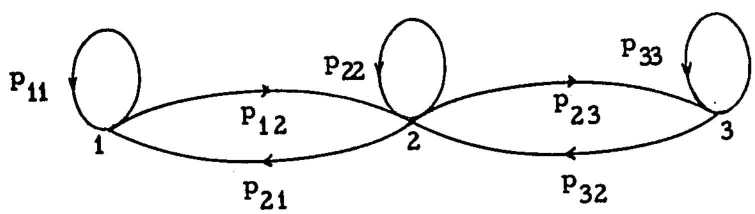

Fig. 4: State transition graph for one-mode T. L. P.

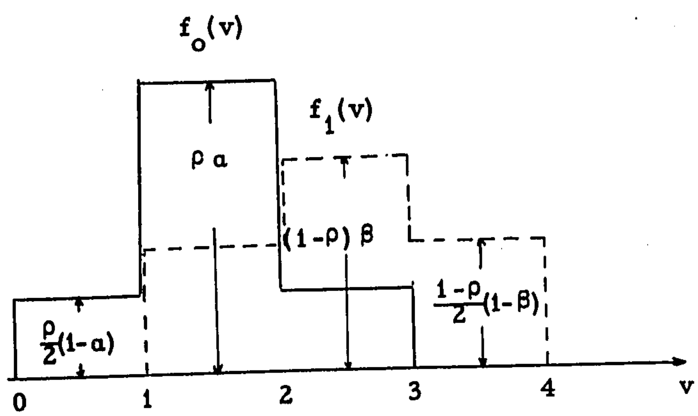

Fig. 5: Quantal probability density functions.

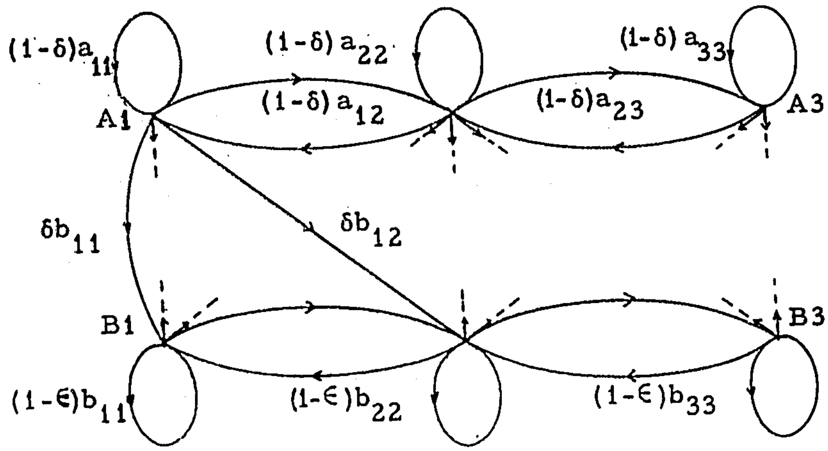

Fig. 6: State graph of a two-mode process.

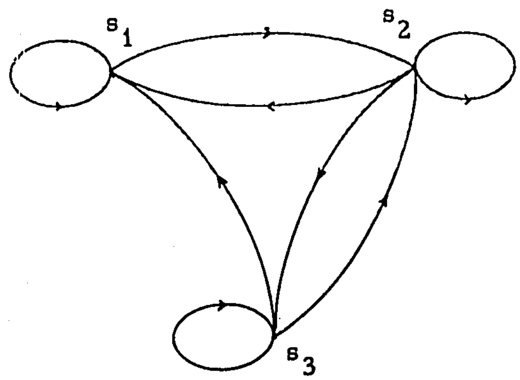

Fig. 7: Unifilar Marcov source.

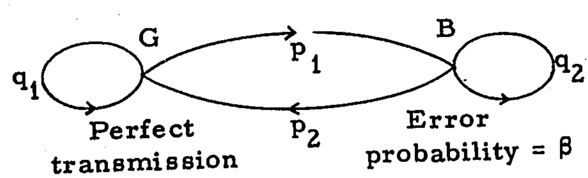

Fig. 8: Burst noise channel.

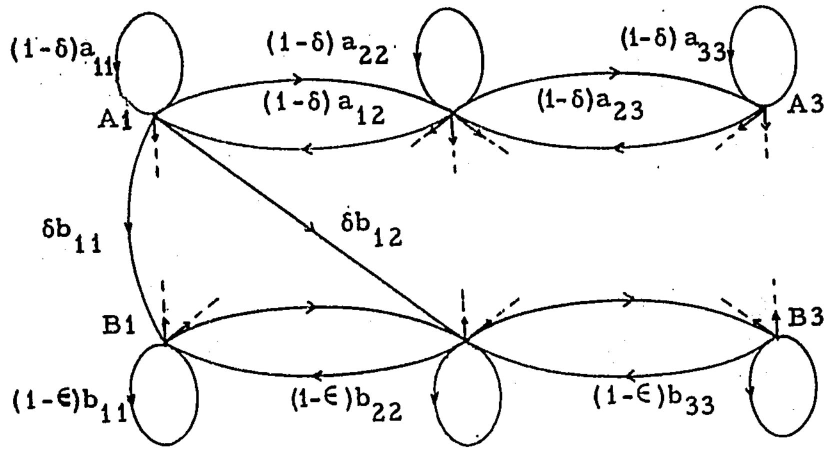

Fig. 6: State graph of a two-mode process.

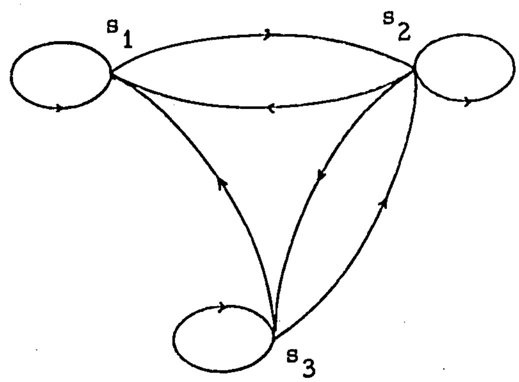

Fig. 7: Unifilar Marcov source.

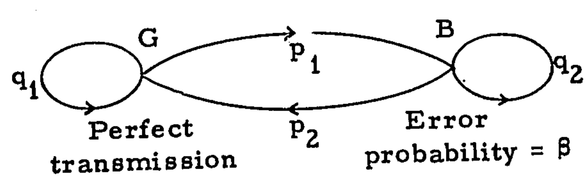

Fig. 8: Burst noise channel.

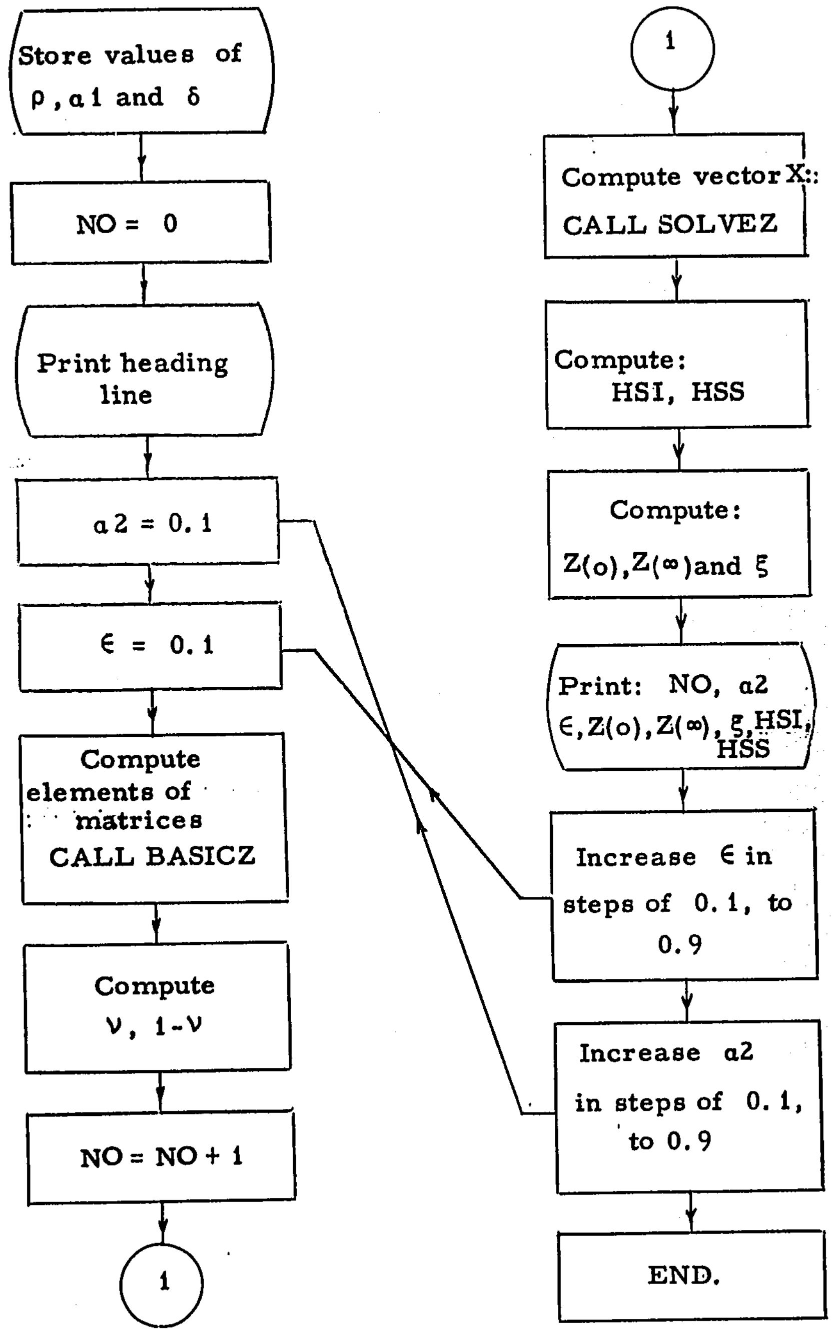

Fig. 9: Block diagram of the main program (Program No. 1).

APPENDIX I

Z - Transform Pairs

No.	Time Function	Z- Transform
1	$f(n)$	$f(z)$
2	$f_1(n) + f_2(n)$	$f_1(z) + f_2(z)$
3	$k^* f(n)$	$k f(z)$
4	$f(n-1)$	$z f(z)$
5	$f(n+1)$	$z^{-1} [f(z) - f(o)]$
6	a^n	$1/(1 - a z)$
7	1 (unit step)	$1/(1 - z)$

*k is a constant

APPENDIX II

Stochastic matrix of the proposed model.

$$
\bar{P} = \begin{pmatrix}
(1-\delta)a_1 & (1-\delta)a_2 & (1-\delta)a_3 & \delta b_1 & \delta b_2 & \delta b_3 \\
(1-\delta)a_1 & (1-\delta)a_2 & (1-\delta)a_3 & \delta b_1 & \delta b_2 & \delta b_3 \\
(1-\delta)a1 & (1-\delta)a_2 & (1-\delta)a_3 & \delta b_1 & \delta b_2 & \delta b_3 \\
\epsilon a_1 & \epsilon a_2 & \epsilon a_3 & (1-\epsilon) b_1 & (1-\epsilon) b_2 & (1-\epsilon) b_3 \\
\epsilon a_1 & \epsilon a_2 & \epsilon a_3 & (1-\epsilon)b_1 & (1-\epsilon)b_2 & (1-\epsilon)b_3 \\
\epsilon a_1 & \epsilon a_2 & \epsilon a_3 & (1-\epsilon)b_1 & (1-\epsilon)b_2 & (1-\epsilon)b_3
\end{pmatrix}
$$

```
C     ELEMENT COMPUTATION FOR MATRICES M.INGHEL
      SUBROUTINE BASICZ (R,A1,A2,D,E,C,Q,S,T,A,B)
      DIMENSION S(3),T(3),Q(6),A(3,3),B(3,3),C(6,7)
      R1=R/2.
      R2=(1.-R)/2.
      A3=1.+A1
      A4=1.-A1
      A5=1.+A2
      A6=1.-A2
      F=1.-D
      G=1.-E
      DELTS=A4*R2**2+R1*(R2*A3+A4*R1)
      DELTT=A6*R2**2+R1*(R2*A5+A6*R1)
      S(1)=(A4*R2**2)/DELTS
      S(2)=(R1*R2*A3)/DELTS
      S(3)=(A4*R1**2)/DELTS
      T(1)=(A6*R2**2)/DELTT
      T(2)=(A5*R1*R2)/DELTT
      T(3)=(A6*R1**2)/DELTT
      A(1,1)=1.-R1*A3
      A(1,2)=R1*A3
      A(1,3)=0.
      A(2,1)=R2*A4
      A(2,2)=A3/2.
      A(2,3)=R1*A4
      A(3,1)=0.
      A(3,2)=R2*A3
      A(3,3)=1.-R2*A3
      B(1,1)=1.-R1*A5
      B(1,2)=R1*A5
      B(1,3)=0.
      B(2,1)=R2*A6
      B(2,2)=A5/2.
      B(2,3)=R1*A6
      B(3,1)=0.
      B(3,2)=R2*A5
      B(3,3)=1.-R2*A5
      DO 17 I=1,3
      DO 17 J=1,3
      C(I,J)=F*A(J,I)
      C(I,J+3)=E*A(J,I)
      C(I+3,J)=D*B(J,I)
```

-49-

```
 17 C(I+3,J+3)=G*B(J,I)
    DO 21 K=1,5
    C(K,K)=C(K,K)-1.
 21 C(K,7)=0.
    DO 25 M=1,7
 25 C(6,M)=1.
    DO 29 L=1,3
    Q(L)=F*A(L,L)+D*B(L,L)
 29 Q(L+3)=E*A(L,L)+G*B(L,L)
    RETURN
    END
```

```
C     GAUSS-JORDAN SOLUTION OF SIMULTANEOUS EQTN.
C     PROGRAMMED BY D.C.BAXTER N.R.C. SEP. 1963
C     ADAPTED FOR I.B.M. BY M.INGHEL
      SU ROUTINE SOLVEZ (A)
      DIMENSION A(6,7)
      DO 205 I=1,6
      IF(A(I,I))200,206,200
  200 P=1./A(I,I)
      DO 201 J=I,7
  201 A(I,J)=P*A(I,J)
      DO 202 K=1,6
      IF(I-K)203,202,203
  203 Q=A(K,I)
      DO 204 J=I,7
  204 A(K,J)=A(K,J)-Q*A(I,J)
  202 CONTINUE
  205 CONTINUE
      RETURN
  206 IF(6-I)300,304,300
  300 N=I+1
      DO 303 L=N,6
      IF(A(L,I))301,303,301
  301 DO 302 J=I,7
      HOLD=A(I,J)
      A(I,J)=A(L,J)
  302 A(L,J)=HOLD
      GO TO 200
  303 CONTINUE
  304 DO 305 M=1,6
  305 A(M,7)=0.
      RETURN
      END
```

```
C     PERFORMANCE COMPUTATION M.INGHEL
C     PROGRAM NO.1
      DIMENSION S(3),T(3),Z(4),Q(6),C(6,7),AA(3),AB(3),BB(3),BA(3),
     1A(3,3),B(3,3),CC(6,6),EE(4),FF(4),GG(4),HH(4),QQ(4),DD(6)
      A1=0.5
      D=0.1
      R=0.1
      NO=0
      PRINT 116
      DO 114 IA2=1,9
      XA2=IA2
      A2=XA2/10.
      DO 114 IE=1,9
      XE=IE
      E=XE/10.
      CALL BASICZ (R,A1,A2,D,E,C,Q,S,T,A,B)
      G1=E/(E+D)
      G2=1.-G1
      NO=NO+1
      CALL SO VEZ(C)
      Y=1./LOGF(2.)
      DO 211 I=1,3
      AA(I)=(1.-D)*S(I)
      AB(I)=D*T(I)
      BB(I)=E*S(I)
  211 BA(I)=(1.-E)*T(I)
      DO 212 K=1,4
  212 QQ(K)=0.
      DO 213 L=1,3
      QQ(1)=QQ(1)-AA(L)*Y*LOGF(AA(L))
      QQ(2)=QQ(2)-AB(L)*Y*LOGF(AB(L))
      QQ(3)=QQ(3)-BB(L)*Y*LOGF(BB(L))
  213 QQ(4)=QQ(4)-BA(L)*Y*LOGF(BA(L))
      HSI=G1*(QQ(1)+QQ(2))+G2*(QQ(3)+QQ(4))
      DO 224 I=1,3
      DO 224 J=1,3
      CC(I,J)=(1.-D)*A(I,J)
      CC(I+3,J)=E*A(I,J)
      CC(I,J+3)=D*B(I,J)
  224 CC(I+3,J+3)=(1.-E)*B(I,J)
      DO 225 K=1,6
```

```
  225 DD(K)=0.
      DO 226 L=1,2
      GG(L)=CC(4,L)*Y*LOGF(CC(4,L))
      GG(L+2)=CC(4,L+3)*Y*LOGF(CC(4,L+3))
      EE(L)=CC(1,L)*Y*LOGF(CC(1,L))
  226 EE(L+2)=CC(1,L+3)*Y*LOGF(CC(1,L+3))
      DO 227 M=1,4
      DD(4)=DD(4)+GG(M)
  227 DD(1)=DD(1)+EE(M)
      DO 228 N=1,6
      DD(5)=DD(5)+CC(5,N)*Y*LOGF(CC(5,N))
  228 DD(2)=DD(2)+CC(2,N)*Y*LOGF(CC(2,N))
      DO 229 I=2,3
      HH(I-1)=CC(6,I)*Y*LOGF(CC(6,I))
      HH(I+1)=CC(6,I+3)*Y*LOGF(CC(6,I+3))
      FF(I-1)=CC(3,I)*Y*LOGF(CC(3,I))
  229 FF(I+1)=CC(3,I+3)*Y*LOGF(CC(3,I+3))
      DO 230 J=1,4
      DD(6)=DD(6)+HH(J)
  230 DD(3)=DD(3)+FF(J)
      HSS=0.
      DO 236 M=1,6
  236 HSS=HSS-C(M,7)*DD(M)
      DO 111 L=1,4
  111 Z(L)=0.
      DO 112 M=1,3
      Z(1)=Z(1)+C(M,7)*Q(M)
      Z(2)=Z(2)+C(M+3,7)*Q(M)
      Z(3)=Z(3)+C(M,7)*Q(M+3)
  112 Z(4)=Z(4)+C(M+3,7)*Q(M+3)
      ZO=Z(1)+Z(4)
      DET=G1*(Z(1)+Z(2))+G2*(Z(3)+Z(4))
      PROB=0.
      DO 113 L=1,3
  113 PROB=PROB+G1*S(L)*Q(L)+G2*T(L)*Q(L+3)
  114 PRINT 115,NO,A2,E,ZO,DET,PROB,HSS,HSI
      CALL EXIT
  115 FORMAT(/I15,2F5.1,3F9.5,2F10.5)
  116 FORMAT(//3X,2HNO,3X,2HA2,4X,1HE,5X,2HZO,7X,3HDET,5X,4HPROB,6X,
     13HHSS,7X,3HHSI)
      END
```

NO	A2	E	ZO	DET	PROB	HSS	HSI
1	.1	.1	.88351	.87908	.88341	.94217	1.21263
2	.1	.2	.88030	.87707	.88003	1.04381	1.34209
3	.1	.3	.87878	.87658	.87834	1.07117	1.38344
4	.1	.4	.87792	.87643	.87732	1.07361	1.39430
5	.1	.5	.87737	.87639	.87665	1.06511	1.39144
6	.1	.6	.87701	.87639	.87617	1.05072	1.38110
7	.1	.7	.87675	.87640	.87580	1.03234	1.36576
8	.1	.8	.87656	.87641	.87552	1.01030	1.34609
9	.1	.9	.87642	.87642	.87530	.98330	1.32100
10	.2	.1	.88063	.87795	.88064	.95766	1.24323
11	.2	.2	.87865	.87672	.87855	1.05410	1.36250
12	.2	.3	.87772	.87641	.87751	1.07887	1.39874
13	.2	.4	.87719	.87632	.87688	1.07976	1.40654
14	.2	.5	.87687	.87630	.87646	1.07023	1.40164
15	.2	.6	.87665	.87629	.87616	1.05510	1.38984
16	.2	.7	.87650	.87630	.87594	1.03618	1.37342
17	.2	.8	.87639	.87630	.87576	1.01370	1.35289
18	.2	.9	.87631	.87631	.87562	.98636	1.32712
19	.3	.1	.87833	.87704	.87837	.97175	1.27608
20	.3	.2	.87735	.87643	.87734	1.06337	1.38440

21	.3	.3	.87689	.87628	.87682	1.08575	1.41516
22	.3	.4	.87664	.87624	.87651	1.08521	1.41968
23	.3	.5	.87648	.87622	.87631	1.07474	1.41259
24	.3	.6	.87638	.87622	.87616	1.05895	1.39923
25	.3	.7	.87631	.87622	.87605	1.03952	1.38163
26	.3	.8	.87626	.87622	.87596	1.01666	1.36019
27	.3	.9	.87622	.87622	.87589	.98902	1.33369
28	.4	.1	.87676	.87640	.87678	.98383	1.31118
29	.4	.2	.87648	.87623	.87649	1.07126	1.40779
30	.4	.3	.87635	.87619	.87634	1.09157	1.43271
31	.4	.4	.87628	.87617	.87625	1.08981	1.43372
32	.4	.5	.87624	.87617	.87619	1.07853	1.42429
33	.4	.6	.87621	.87617	.87615	1.06216	1.40925
34	.4	.7	.87619	.87617	.87612	1.04231	1.39040
35	.4	.8	.87618	.87617	.87609	1.01913	1.36799
36	.4	.9	.87617	.87617	.87608	.99122	1.34071
37	.5	.1	.87614	.87614	.87614	.99293	1.34809
38	.5	.2	.87614	.87614	.87614	1.07724	1.43240
39	.5	.3	.87614	.87614	.87614	1.09600	1.45117
40	.5	.4	.87614	.87614	.87614	1.09332	1.44848
41	.5	.5	.87614	.87614	.87614	1.08143	1.43659

42	.5	.6	.87614	.87614	.87614	1.06464	1.41980
43	.5	.7	.87614	.87614	.87614	1.04446	1.39963
44	.5	.8	.87614	.87614	.87614	1.02103	1.37620
45	.5	.9	.87614	.87614	.87614	.99293	1.34809
46	.6	.1	.87682	.87638	.87689	.99748	1.38527
47	.6	.2	.87648	.87619	.87654	1.08047	1.45719
48	.6	.3	.87634	.87615	.87636	1.09853	1.46975
49	.6	.4	.87626	.87615	.87625	1.09540	1.46335
50	.6	.5	.87622	.87614	.87618	1.08321	1.44899
51	.6	.6	.87619	.87615	.87613	1.06619	1.43042
52	.6	.7	.87617	.87615	.87609	1.04584	1.40892
53	.6	.8	.87616	.87615	.87606	1.02227	1.38446
54	.6	.9	.87615	.87615	.87604	.99406	1.35553
55	.7	.1	.87929	.87732	.87980	.99480	1.41806
56	.7	.2	.87771	.87643	.87808	1.07963	1.47905
57	.7	.3	.87704	.87624	.87722	1.09838	1.48615
58	.7	.4	.87669	.87619	.87670	1.09556	1.47647
59	.7	.5	.87649	.87618	.87636	1.08351	1.45992
60	.7	.6	.87637	.87618	.87611	1.06657	1.43979
61	.7	.7	.87629	.87619	.87593	1.04626	1.41712
62	.7	.8	.87623	.87619	.87578	1.02270	1.39175

63	.7	.9	.87619	.87619	.87567	.99449	1.36209
64	.8	.1	.88439	.87928	.88630	.97986	1.43243
65	.8	.2	.88010	.87691	.88153	1.07240	1.48863
66	.8	.3	.87835	.87642	.87915	1.09426	1.49334
67	.8	.4	.87748	.87629	.87772	1.09298	1.48222
68	.8	.5	.87699	.87626	.87676	1.08179	1.46471
69	.8	.6	.87669	.87626	.87608	1.06537	1.44390
70	.8	.7	.87650	.87626	.87557	1.04540	1.42071
71	.8	.8	.87636	.87627	.87517	1.02208	1.39494
72	.8	.9	.87627	.87627	.87485	.99403	1.36496
73	.9	.1	.89350	.88282	.89952	.94204	1.37834
74	.9	.2	.88409	.87774	.88856	1.05424	1.45257
75	.9	.3	[illegible]	.87672	.88308	1.08368	1.46629
76	.9	.4	.87871	.87645	.87979	1.08608	1.46058
77	.9	.5	.87775	.87638	.87760	1.07696	1.44668
78	.9	.6	.87717	.87637	.87603	1.06180	1.42844
79	.9	.7	.87680	.87637	.87486	1.04267	1.40719
80	.9	.8	.87656	.87638	.87395	1.01992	1.38292
81	.9	.9	.87639	.87639	.87321	.99228	1.35414

```
C      PERFORMANCE COMPUTATION M.INGHEL
C      PROGRAM NO.2
       DIMENSION S(3),T(3),Z(4),Q(6),C(6,7),AA(3),AB(3),BB(3),BA(3),
      1A(3,3),B(3,3),CC(6,6),EE(4),FF(4),GG(4),HH(4),QQ(4),DD(6)
       A1=0.5
       D=0.1
       E=0.1
       NO=0
       PRINT 116
       DO 114 IA2=1,9
       XA2=IA2
       A2=XA2/10.
       DO 114 IR=1,9
       XR=IR
       R=XR/10.
       CALL BASICZ (R,A1,A2,D,E,C,Q,S,T,A,B)
       G1=E/(E+D)
       G2=1.-G1
       NO=NO+1
       CALL SOLVEZ(C)
       Y=1./LOGF(2.)
       DO 211 I=1,3
       AA(I)=(1.-D)*S(I)
       AB(I)=D*T(I)
       BB(I)=E*S(I)
  211  BA(I)=(1.-E)*T(I)
       DO 212 K=1,4
  212  QQ(K)=0.
       DO 213 L=1,3
       QQ(1)=QQ(1)-AA(L)*Y*LOGF(AA(L))
       QQ(2)=QQ(2)-AB(L)*Y*LOGF(AB(L))
       QQ(3)=QQ(3)-BB(L)*Y*LOGF(BB(L))
  213  QQ(4)=QQ(4)-BA(L)*Y*LOGF(BA(L))
       HSI=G1*(QQ(1)+QQ(2))+G2*(QQ(3)+QQ(4))
       DO 224 I=1,3
       DO 224 J=1,3
       CC(I,J)=(1.-D)*A(I,J)
       CC(I+3,J)=E*A(I,J)
       CC(I,J+3)=D*B(I,J)
  224  CC(I+3,J+3)=(1.-E)*B(I,J)
       DO 225 K=1,6
```

```
  225 DD(K)=0.
      DO 226 L=1,2
      GG(L)=CC(4,L)*Y*LOGF(CC(4,L))
      GG(L+2)=CC(4,L+3)*Y*LOGF(CC(4,L+3))
      EE(L)=CC(1,L)*Y*LOGF(CC(1,L))
  226 EE(L+2)=CC(1,L+3)*Y*LOGF(CC(1,L+3))
      DO 227 M=1,4
      DD(4)=DD(4)+GG(M)
  227 DD(1)=DD(1)+EE(M)
      DO 228 N=1,6
      DD(5)=DD(5)+CC(5,N)*Y*LOGF(CC(5,N))
  228 DD(2)=DD(2)+CC(2,N)*Y*LOGF(CC(2,N))
      DO 229 I=2,3
      HH(I-1)=CC(6,I)*Y*LOGF(CC(6,I))
      HH(I+1)=CC(6,I+3)*Y*LOGF(CC(6,I+3))
      FF(I-1)=CC(3,I)*Y*LOGF(CC(3,I))
  229 FF(I+1)=CC(3,I+3)*Y*LOGF(CC(3,I+3))
      DO 230 J=1,4
      DD(6)=DD(6)+HH(J)
  230 DD(3)=DD(3)+FF(J)
      HSS=0.
      DO 236 M=1,6
  236 HSS=HSS-C(M,7)*DD(M)
      DO 111 L=1,4
  111 Z(L)=0.
      DO 112 M=1,3
      Z(1)=Z(1)+C(M,7)*Q(M)
      Z(2)=Z(2)+C(M+3,7)*Q(M)
      Z(3)=Z(3)+C(M,7)*Q(M+3)
  112 Z(4)=Z(4)+C(M+3,7)*Q(M+3)
      ZO=Z(1)+Z(4)
      DET=G1*(Z(1)+Z(2))+G2*(Z(3)+Z(4))
      PROB=0.
      DO 113 L=1,3
  113 PROB=PROB+G1*S(L)*Q(L)+G2*T(L)*Q(L+3)
  114 PRINT115,NO,A2,R,ZO,DET,PROB,HSS,HSI
      CALL EXIT
  115 FORMAT(/I15,2F5.1,3F9.5,2F10.5)
  116 FORMAT(//3X,2HNO,3X,2HA2,4X,1HR,5X,2HZO,7X,3HDET,5X,4HPROB,6X,
     13HHSS,7X,3HHSI)
      END
```

NO	A2	R	ZU	DET	PROB	HSS	HSI
1	.1	.1	.88351	.87908	.88341	.94217	1.21263
2	.1	.2	.79288	.78510	.79369	1.21631	1.56127
3	.1	.3	.72763	.71746	.72887	1.39098	1.77812
4	.1	.4	.68811	.67653	.68932	1.48970	1.90256
5	.1	.5	.67487	.66281	.67599	1.52175	1.94359
6	.1	.6	.68811	.67653	.68932	1.48970	1.90256
7	.1	.7	.72763	.71746	.72887	1.39098	1.77812
8	.1	.8	.79288	.78510	.79369	1.21631	1.56127
9	.1	.9	.88351	.87908	.88341	.94217	1.21263
10	.2	.1	.88063	.87795	.88064	.95766	1.24323
11	.2	.2	.79018	.78555	.79078	1.23254	1.58612
12	.2	.3	.72648	.72051	.72732	1.40309	1.78725
13	.2	.4	.68852	.68178	.68932	1.49759	1.89739
14	.2	.5	.67590	.66891	.67664	1.52794	1.93279
15	.2	.6	.68852	.68178	.68932	1.49759	1.89739
16	.2	.7	.72648	.72051	.72732	1.40309	1.78725
17	.2	.8	.79018	.78555	.79078	1.23254	1.58612
18	.2	.9	.88063	.87795	.88064	.95766	1.24323
19	.3	.1	.87833	.87704	.87837	.97175	1.27608
20	.3	.2	.78909	.78690	.78944	1.24436	1.60824

21	.3	.3	.72785	.72508	.72830	1.40832	1.78982
22	.3	.4	.69201	.68891	.69242	1.49711	1.88389
23	.3	.5	.68021	.67700	.68059	1.52530	1.91323
24	.3	.6	.69201	.68891	.69242	1.49711	1.88389
25	.3	.7	.72785	.72508	.72830	1.40832	1.78982
26	.3	.8	.78909	.78690	.78944	1.24436	1.60824
27	.3	.9	.87833	.87704	.87837	.97175	1.27608
28	.4	.1	.87676	.87640	.87678	.98383	1.31118
29	.4	.2	.78992	.78934	.79003	1.25054	1.62598
30	.4	.3	.73210	.73137	.73223	1.40521	1.78380
31	.4	.4	.69893	.69813	.69905	1.48686	1.86017
32	.4	.5	.68812	.68729	.68823	1.51243	1.88310
33	.4	.6	.69893	.69813	.69905	1.48686	1.86017
34	.4	.7	.73210	.73137	.73223	1.40521	1.78380
35	.4	.8	.78992	.78934	.79003	1.25054	1.62598
36	.4	.9	.87676	.87640	.87678	.98383	1.31118
37	.5	.1	.87614	.87614	.87614	.99293	1.34809
38	.5	.2	.79310	.79310	.79310	1.24932	1.63664
39	.5	.3	.73966	.73966	.73966	1.39184	1.76621
40	.5	.4	.70967	.70967	.70967	1.46496	1.82357
41	.5	.5	.69999	.69999	.70000	1.48753	1.83994

42	.5	.6	.70967	.70967	.70967	1.46496	1.82357
43	.5	.7	.73966	.73966	.73966	1.39184	1.76621
44	.5	.8	.79310	.79310	.79310	1.24932	1.63664
45	.5	.9	.87614	.87614	.87614	.99293	1.34809
46	.6	.1	.87682	.87638	.87689	.99748	1.38527
47	.6	.2	.79917	.79849	.79936	1.23811	1.63562
48	.6	.3	.75108	.75028	.75128	1.36552	1.73260
49	.6	.4	.72469	.72383	.72486	1.42888	1.77035
50	.6	.5	.71627	.71538	.71641	1.44814	1.78028
51	.6	.6	.72469	.72383	.72486	1.42888	1.77035
52	.6	.7	.75108	.75028	.75128	1.36552	1.73260
53	.6	.8	.79917	.79849	.79936	1.23811	1.63562
54	.6	.9	.87682	.87638	.87689	.99748	1.38527
55	.7	.1	.87929	.87732	.87980	.99480	1.41806
56	.7	.2	.80886	.80594	.80986	1.21302	1.61459
57	.7	.3	.76700	.76361	.76795	1.32240	1.67589
58	.7	.4	.74453	.74091	.74528	1.37505	1.69482
59	.7	.5	.73742	.73374	.73808	1.39082	1.69893
60	.7	.6	.74453	.74091	.74528	1.37505	1.69482
61	.7	.7	.76700	.76361	.76795	1.32240	1.67589
62	.7	.8	.80886	.80594	.80986	1.21302	1.61459

63	.7	.9	.87929	.87732	.87980	.99480	1.41806
64	.8	.1	.88439	.87928	.88630	.97986	1.43243
65	.8	.2	.82317	.81601	.82616	1.16767	1.55718
66	.8	.3	.78823	.78015	.79081	1.25651	1.58386
67	.8	.4	.76980	.76132	.77177	1.29801	1.58759
68	.8	.5	.76401	.75541	.76572	1.31028	1.58739
69	.8	.6	.76980	.76132	.77177	1.29801	1.58759
70	.8	.7	.78823	.78015	.79081	1.25651	1.58386
71	.8	.8	.82317	.81601	.82616	1.16767	1.55718
72	.8	.9	.88439	.87928	.88630	.97986	1.43243
73	.9	.1	.89350	.88282	.89952	.94204	1.37834
74	.9	.2	.84351	.82949	.85070	1.09029	1.42539
75	.9	.3	.81579	.80051	.82138	1.15714	1.43157
76	.9	.4	.80125	.78548	.80532	1.18789	1.43008
77	.9	.5	.79668	.78079	.80019	1.19693	1.42894
78	.9	.6	.80125	.78548	.80532	1.18789	1.43008
79	.9	.7	.81579	.80051	.82138	1.15714	1.43157
80	.9	.8	.84351	.82949	.85070	1.09029	1.42539
81	.9	.9	.89350	.88282	.89952	.94204	1.37834

REFERENCES

(1) J. Sklansky, "Learning Systems for Automatic Control" I.E.E.E Transactions on Automatic Control, AC-11, No.1, January, 1966, pp. 16 - 19.

(2) Ya. Z. Tsypkin, "Adaptation, Learning and Self-Learning in Automatic Systems", Avtomatika i Telemekhanika, No. 1, January, 1966, pp. 23 - 61.

(3) Z.J. Nikolic, K.S. Fu, "A Mathematical Model of Learning in an Unknown Random Environment" Proceedings of the National Electronics Conference 1966, Vol. 22, pp. 607 - 612.

(4) J. Sklansky, "Threshold Training of the Two-Mode Signal Detection", Preprint Vol. of 1964 Joint Automatic Control Conference, I.E.E.E., pp. 6 - 12.

(5) D.N. Chorafas, "Systems and Simulation", Academic Press, New York, 1965.

(6) R. Ash, "Information Theory", Interscience Publishers, New York, 1966.

(7) C.W. Helstrom, "Statistical Theory of Signal Detection", Pergamon Press, New York, 1960.

(8) W.B. Davenport Jr. and W.L. Root, "An Introduction to the Theory of Random Signals and Noise", Lincoln Laboratory, M.I.T., McGraw-Hill Book Company, Inc., New York, 1958.

(9) J. Neyman and E.S. Pearson, "On the Problem of the Most Efficient Tests of Statistical Hypotheses", Phil. Trans. Roy. Soc., A 231, 289, 1933.

(10) A.A. Pervozvanskii, "Random Processes in Nonlinear Control Systems,", Academic Press, New York, 1965

(11) E.F. Beckenbach, Editor, "Modern Mathematics for the Engineer", McGraw-Hill Book Company, Inc., New York, 1956.

(12) E.N. Gilbert, "Capacity of a Burst-Noise Channel", Bell Systems Technical Journal, Vol. 39, September, 1960, pp. 1253 - 1266.

(13) E.J. Baghdady, Editor, "Lectures on Communication System Theory", McGraw-Hill Book Company, Inc., New York, 1961.

(14) J.G. Kemeny and J.L. Snell, "Finite Markov Chains", D. Van Nostrand Company, Princeton, N.J., 1960.

(15) J.G. Kemeny, H. Mirkil, J.L. Snell, G.L. Thompson, "Finite Mathematical Structures", Prentice-Hall, Inc., Englewood Cliffs, N.J., 1963.

(16) W.H. Huggins, "Signal Flow Graphs and Random Signals", Proc. I.R.E., Vol. 45, January, 1957, pp. 74 - 86.

(17) R.W. Sittler, "Systems Analysis of Discrete Markov Processes", I.R.E. Trans. on Circuit Theory, CT-3, December, 1956, pp. 257 - 266.

(18) E.J. Jury, "Sampled-Data Control Systems", John Wiley and Sons, New York, 1958.

(19) R.A. Howard, "Dynamic Programming and Markov Processes", The Technology Press of the M.I.T. and John Wiley and Sons, Inc., New York, 1960.

(20) Operations Research Center, M.I.T., "Notes on Operations Research," Technology Press, Cambridge, 1959.

(21) W. Feller, "An Introduction to Probability Theory and its Applications", Vol. 1, 2nd. Edition, John Wiley and Sons, New York, 1957.

(22) D.D. McCracken and W.S. Dorn, "Numerical Methods and Fortran Programming", John Wiley and Sons, New York, 1964.

(23) G.N. Tsertsvadze, "Certain Properties of Stochastic Automata and Certain Methods for Synthesizing Them", Avtomatika i Telemekhanika, Vol. 24, pp. 341 - 352, March, 1963. (English Translation : Automat. Remote Control, pp. 316 - 326).

(24) J.N. Warfield, "Principles of Logic Design," Ginn and Company, Boston, 1963.

VITA

Name :	Mondy Inghel
Date of Birth :	December 6, 1932
Place of Birth :	Romania
Education :	
University :	"Technion", Israel Institute of Technology
Place	Haifa, Israel
Period	1957 - 1961
Degree	B.Sc. E.E.
Experience :	
1961 - 1964	Development, maintenance and design of telecommunication stations. Radio and Telecommunication Division, Department of Posts, Israel.
1964 - 1965	Development of communication equipment. Measurement Engineering Limited, Arnprior, Ontario.