



National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Services des thèses canadiennes

Ottawa, Canada
K1A 0N4

CANADIAN THESES

THÈSES CANADIENNES

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

**THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED**

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

**LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE**

**Analysis and Testing of a New Hybrid Parallel Search
Scheme for Spread Spectrum Codes Acquisition**

Ho Tuan Minh

A Thesis

in

The Department

of

Electrical Engineering

**Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering at
Concordia University
Montréal, Québec, Canada**

July 1987

© Ho Tuan Minh, 1987

Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-37105-6

ABSTRACT

Analysis and Testing of a New Hybrid Parallel Search
Scheme for Spread Spectrum Codes Acquisition

Ho Tuan Minh

A parallel technique for SS code acquisition will be presented. The optimal parameter design problem involved in this technique is investigated. A prototype model of this acquisition scheme is built and tested. Analysis models are described for the different cases and the results of the testing is presented. Also an improved version of this scheme is presented and analyzed. The results of the performance are discussed and a comparison with other multi-dwell [3] and parallel techniques [4] is considered in terms of the mean acquisition time and standard deviation involved.

ACKNOWLEDGEMENT

The author wishes to thank his supervisor, Dr. A.K. Elhakeem, for his continuous encouragement, guidance, and advice during the course of this thesis.

Appreciation is, as well, due to Ma G. Minh for his helpful discussion about editing.

Thanks also to Miss Suzette Paradis for her effort in the typing of the thesis.

TABLE OF CONTENTS

	PAGE
Abstract	iii
Acknowledgements	iv
Table of contents	v
List of figures	vii
List of tables	ix
List of Symbols and Abbreviations	x
CHAPTER 1 : Introduction and Review of Some of Previous Works	
1.1 General	1
1.2 Acquisition technique for SS signal	3
1.3 Review of Previous Works	5
1.3.1 DiCarlo's Scheme	5
1.3.2 Milstein's Scheme	6
1.4 Thesis Contribution	8
1.5 Outline of the Thesis	10
CHAPTER 2 : Basic Serial Search Technique for SS code acquisition	
2.1 Single Dwell Serial Search Method	12
2.2 Multiple Dwell Technique	17
2.3 Acquisition Using Matched Filter Technique	27
CHAPTER 3 : Optimal Sequence Detection Circuit Approach to Spread Spectrum Code Acquisition	
3.1 Introduction	29
3.2 The Parallel Sequence Detection Circuits Code Acquisition Method	30
3.3 Analysis	36
3.4 Results and Conclusions	44
CHAPTER 4 : Implementation and Testing of the Sequence Detection Circuit Acquisition Technique	
4.1 The TAD Circuits	79
4.2 The SDCs	81
4.3 The Detailed Algorithm of the Prototype	86
4.4 Testing Performance of the System	88
4.4.1 Test Result Report 1	92
4.4.2 Test Result Report 2	93
4.4.3 Test Result Set 3	95
4.4.4 Conclusion	96
CHAPTER 5 : A Hybrid Sequential/Parallel Scheme for Combined Acquisition/Search Lock of Spread Spectrum Codes	
5.1 Introduction	97
5.2 The New Acquisition/Search Lock Scheme	100
5.3 Analysis	108
5.4 Results and Conclusions	139

	PAGE
CHAPTER 6 : Comparison with Dicarolo's and Milstein's Schemes	
6.1 Comparison with Dicarolo's Scheme	162
6.2 Comparison with Milstein's Scheme	165
CHAPTER 7 : Summary, Conclusions and Suggestions	172
REFERENCES	176

LIST OF FIGURES

- Fig. 1.1 : A DS/SS Communication System
- Fig. 1.2 : A FH/SS Communication System
- Fig. 2.1 : A Single Dwell Search PN Acquisition System
- Fig. 2.2 : Flow Diagram of the Single Dwell System
- Fig. 2.3 : A N-Dwell Serial Search PN Acquisition System
- Fig. 2.4 : Flow Diagram of the N-Dwell System
- Fig. 2.5 : Flow Diagram to Compute ϕ_D , λ_D , ϕ_{FA} , λ_{FA}
- Fig. 3.1 : Configuration of NN-Sequence Detection Circuits
- Fig. 3.2 : Flow Chart of the Acquisition Technique
- Fig. 3.3 : Flow Diagram Representation of the Acquisition Scheme
- Fig. 3.4 : Representation of the μ_d Branch
- Fig. 3.5 : Representation of the λ_m Branch
- Fig. 3.6 : Representation of the μ_f Branch
- Fig. 3.7 : Representation of the λ_r Branch
- Fig. 3.8 a) to d) : Mean Acquisition Time Versus Number of SDCs for Varying P_b
- Fig. 3.9 a) to d) : Mean Acquisition Time Versus Number of SDCs for Varying P_b
- Fig. 3.10 : Mean Acquisition Time Versus Threshold
- Fig. 3.11 : Mean Acquisition Time Versus Probability P_b
- Fig. 3.12 : Mean Acquisition Time Versus Probability P_b
- Fig. 3.13 a) : Mean Acquisition Time Versus Number of Bits Per TAD M
- Fig. 3.13 b) : Mean Acquisition Time Versus Number of TADs N
- Fig. 3.14 : Mean Acquisition Time Versus P_b for Varying P_b
- Fig. 3.15 : Mean Acquisition Time Versus P_b for Varying P_b
- Fig. 4.1 : Block Diagram of the Detection System
- Fig. 4.2 : Detailed Structure Diagram of an SDC

- Fig. 4.3 : Timing Diagram of the Prototype System
- Fig. 4.4 : Testing Equipment
- Fig. 4.5 : TAD Circuit Board
- Fig. 4.6 : SDCs Board
- Fig. 4.7 : Circuit Board Connection and Layout
- Fig. 5.1 : DS Receiver Block Diagram
- Fig. 5.2 : Details of the Combiner Logic and (T.D.C.)^S Actions
- Fig. 5.3 : Ideal Case of Errorless Reception of the SS Code
- Fig. 5.4 : Typical Example for the Action of the (T.D.C.)^S and Combiner Logic
- Fig. 5.5 : Probability Tree for the Transitions Count of the (T.D.C.)
- Fig. 5.6 : Markovian State Diagram for the Acquisition Time
- Fig. 5.7 a) to c) : Mean Acquisition Time Versus P_b'
- Fig. 5.8 a) to c) : Mean Acquisition Time Versus P_b
- Fig. 5.9 a) and b) : Mean Acquisition Time Versus Number of Tolerable Transitions MM
- Fig. 5.10 a) and b) : Mean Acquisition Time Versus P_b and P_b' ,
Respectively for $L = 256$ and 128
- Fig. 5.11 a) and b) : Mean Acquisition Time Versus Number of (T.D.C.)^S
 NN
- Fig. 5.12 a) and b) : Mean Acquisition Time Versus TAD Threshold TH

LIST OF TABLES

Table 3.1, A to I : Best TH, best NN, and corresponding optimum mean and standard deviation. L 256

Table 3.2, A to H : Best TH, best NN, and corresponding optimum mean and standard deviation. L 128

Table 5.1, A and B : α , β , M_1 , M_2 , M_3 , and corresponding mean and standard deviation. Pb' 0.5

Table 5.2, A and B : α , β , M_1 , M_2 , M_3 , and corresponding mean and standard deviation. Pb 0.3

Table 6.1 : Comparison With Dicarolo's Scheme

Table 6.2 : Comparison With Milstein's Scheme

LIST OF SYMBOLS AND ABBREVIATIONS

SS	Spread Spectrum
DS	Direct Sequence
FH	Frequency Hopping
PN	Pseudonoise
SDC	Sequence Detection Circuits
λ_m	Probability of Miss
μ_d	Probability of Detection
λ_r	Probability of Recognizing Misalignment
μ_f	Probability of False Alarm
z	Delay Associated with False Acquisition
P_{d_i}	Probability that the Correctly Peaked i^{th} Tapped Analog Delay is Detected
P_{f_i}	Probability that the i^{th} Tapped Analog Delay will Falsely Peak
α_i	Probability that the Correctly Peaked i^{th} Tapped Analog Delay is Detected
ϵ_i	Probability that the Correctly Peaked i^{th} Tapped Analog Delay is not Detected
β_i	Probability that the i^{th} Tapped Analog Delay will Falsely Peak
γ_i	Probability that the i^{th} Tapped Analog Delay will not Falsely Peak
τ	Integration Time for a Serial Search Acquisition Scheme
TADs	Tapped Analog Delays
L	Codelength for the Direct Sequence Acquisition Code
N	Number of Tapped Analog Delays
M	Number of Taps per Tapped Analog-Delay
P_{align}	Probability of Alignment

CHAPTER 1

INTRODUCTION AND REVIEW OF SOME OF PREVIOUS WORKS

1.1 General

Since the number of available frequency bands decreases and existing bands become more scarce, the Spread Spectrum (SS) Communication which has originally been devised for military application is now a growing interest for use in commercial applications especially in the mobile environs. By spreading the spectrum a lot of benefits can be obtained [13].

Two main techniques of SS communication are in use: the direct sequence and the frequency hopping spread spectrum techniques.

In the DS method a carrier is modulated by a binary code sequence (PN sequence) and this mixed with the digitized base band information signal to form the signal $S_T(t)$ in a mechanism called "spreading". After transmission and at the receiver the received SS signal is despread and then demodulated to obtain the original signal (Fig. 1.1).

One of the most important benefits of DS/SS communication is the anti-jamming properties due to the fact that the transmitting band is spreaded.

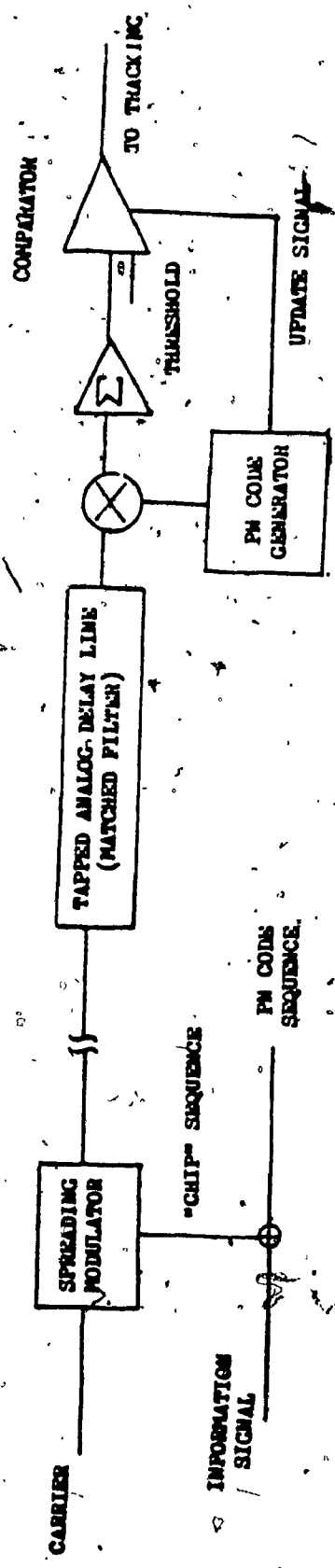


FIG. 1.1 : BLOCK DIAGRAM OF A IS/SS COMMUNICATION SYSTEM

In frequency hopping (FH) spread spectrum systems the FH transmitter mainly consists of a frequency synthesizer and a code generator used to select the PN frequency pattern.

At the receiver, a replica of the frequency pattern is produced by a locally generated PN code sequence which should run synchronously with that at the transmitter.

The received signal is then recovered using this PN replica and the resulting wave-form then appropriately demodulated to receive the information signal.

A typical block diagram of a FH system is shown in Fig. 1.2

1.2 Acquisition techniques for SS signal

To recover the information the original code of the received SS signal must be synchronized with the locally generated code.

The synchronization process of the code of the received signal and the local PN signal consists of two stages. An alignment of the two PN signals to bring the two codes to within one code chip of each other alignment is referred to as PN acquisition. The second stage is fine synchronization to maintain the two codes in the fine alignment and is called PN tracking.

In the thesis we will focus our attention on the PN acquisition that is the coarse alignment of the codes.

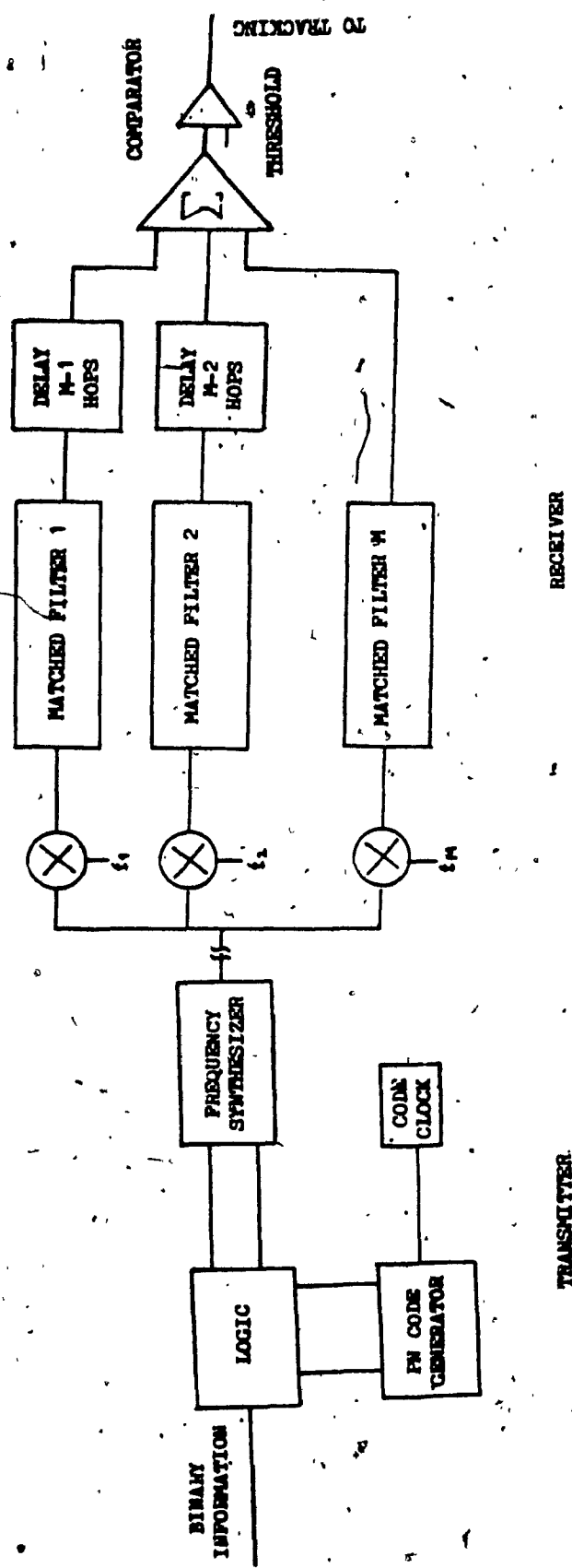


FIG. 1.2 : BLOCK DIAGRAM OF A FH/SS COMMUNICATION SYSTEM

There are several techniques available for PN acquisition. In the following chapter we will describe some of the basic ones which belong to the serial search subclass of the DS/SS acquisition schemes. The simplest one is the single dwell system, the others are multi-dwell serial search and serial search with matched filters. (Chapter 2).

1.3 Review of previous works

We will review some of the previous works having the same objective to reduce the mean acquisition time. Two works we pay attention to are those of Dicarlo and Milstein, which we will describe in the following subsections.

1.3.1 Dicarlo's Scheme

Dicarlo's scheme is a multiple dwell serial search technique [1]. It is the generalized scheme of the basic single dwell serial search. In other words the single dwell serial search technique is just a specific case of the N-dwell search where $N = 1$.

The general form of the N dwell search acquisition scheme described in Dicarlo's paper is illustrated in Fig. 1 of [3]. There are N detectors each with a dwell time τ_i , $i = 1, 2, 3, \dots, N$ where $\tau_{i-1} \leq \tau_i$. The operation of the multi-dwell scheme will be described in Chapter II

Dicarlo described in details the procedure of the derivation of the generating function using the flow graph technique. Dicarlo also has derived the statistical properties of the N dwell system (the mean acquisition time and standard deviation). He described the derivation of the false alarm and correct detection probabilities P_f , P_d and has

given the relation between P_f and P_d .

He examined the performance of the system by calculating the $\mu(N)$ and $\sigma(N)$ [3] which corresponds to a value of \hat{P}_d and \hat{P}_f by computing the minimum value of $F(N)$ for a given N , the corresponding combinations of the dwell time were recorded.

He ended with four Tables [3], two of which (Tables 1,2) will be used to compare with our scheme.

The Tables showed the resulting values of $\mu(N)$ and $\sigma(N)$ corresponding to various values of \hat{P}_d , \hat{P}_f and N for $\text{SNR} = -20\text{dB}$ and $T_p = 10^5\Delta$, $10^5\Delta$, $10^7\Delta$, $10^9\Delta$ respectively [3]. From these results he ended up in the conclusion that the multiple dwell technique yields better performance than a fixed dwell scheme. The performance is enhanced where the penalty time is increased.

According to Dicarlo the most significant result is the reduction in the average acquisition time obtained by increasing the dwell number (N) from 1 to 2. Beyond that increasement, only nominal improvement is gained.

1.3.2 Milstein's scheme

Milstein's scheme is a technique using parallel detectors. The acquisition scheme is displayed in [4]. The system can be described as follows

An array of N convolvers (using surface acoustic wave (SAW) devices) are used to detect the correct code phase position of the

received signal, each having as a reference input one of the subsections of the code length L .

The received code is fed simultaneously into the N convolvers. Since each convolver is "programmed" to match to a subsection of the code length L this guarantees that the correct phase position will be detected by one of the N convolvers somewhere in the integration interval of the convolvers.

Once the correct phase position has been detected, the system enters into the lock mode in which the code is locked to that detected phase position and verification is proceeded.

For verification purposes the code is fed into the N convolvers using the delay system. The convolver outputs are non-coherently summed and compared with a threshold.

If the verification is confirmed (i.e. the summer output exceeds the threshold) then synchronization is declared. If not the system goes back to the search mode.

To enter the lock mode, after the phase position has been decided upon, two more successive "hits" at that phase position are required in order to enter into the lock. Similarly two successive "misses" in the lock mode will be necessary in order for that phase position to be rejected and the search mode reinitiated. A flow diagram of the system is shown in [4].

Milstein has derived expression for the probability that the

correct phase position has been found (or the probability in the search mode), the probabilities of false alarm and correct detection in the lock mode $P_{fa}(\ell)$, $P_d(\ell)$ respectively which are shown in [1] and [4] using the canonical form of the transition matrix derived from this model. He derived the expression for the probability of entering lock, the mean dwell time in an incorrect phase position and the expected time to loss of lock.

Finally, the general form for the mean time to acquire for both the parallel convolver scheme and serial search scheme was derived.

By assuming $T_{fpe}(s) \ll \bar{T}$ [4], the mean acquisition time of the scheme can be compared with the serial search scheme

$$T_{acq}(\text{parallel convolver}) \simeq \frac{2}{K} T_{acq}(\text{serial search})$$

which is equation (27) in Milstein paper [4].

1.4 Thesis contribution

The acquisition or coarse alignment is the first stage of the synchronization in which the received code is aligned with the local PN code to within a fraction of a chip. The objective of the present research is to minimize the mean acquisition time as much as possible. As presented above there have been several works which have the same objective. The most recent and relevant is the schemes of Milstein and D'Carlo.

Our thesis can be divided into 3 main parts. We propose two hybrid parallel scheme to reduce the mean acquisition time. For simplicity we call them the static and the dynamic scheme. The static scheme in which we use the sequence detection circuit to search for the existence of the code has been studied and examined in a previous paper by Elhakeem and Shawn. Our aim here to optimize the scheme.

The dynamic scheme in which while searching for the presence of the code we allow some tolerable number of subcodes to be missed but the synchronization is still declared.

Comparing with Dicarlo's and Milstein's schemes, while Dicarlo uses the multi dwell serial search technique and Milstein uses the parallel search scheme, in our dynamic scheme we use a hybrid parallel/sequential combined search lock method.

Like Milstein's scheme our dynamic scheme first has a band of correlators to search for the code phase and after that some mechanism to check the correction of the code phase. However past decisions are not thrown away as in Milstein's and new acquisition decisions are based also on the past decisions and the inherent code sequence. Forgetting past/subcode decisions (like Milstein) is like saying that uncoded communication systems perform better than systems using coding techniques.

Milstein calls the first stage the search mode and the second stage the lock mode where the system is locked to the code phase chosen in the search mode and the verification is proceeded. While Milstein only uses an array of the convolvers to sum in one codelength time, we

use a band of parallel transition detectors. We also allow the system to tolerate the missing detection of several subcodes which can be caused by pulsed jamming and interference or noise. So our scheme has an advantage in the environments with heavy jamming and noise because it has a memory and past decisions are not thrown away. Also while in Milstein's scheme the independence of the subcodes is assumed, in our system the inherent coding is utilized.

The third main part of the thesis describes the implementation of a prototype model for the static scheme. It includes the testing of the system to confirm the results described in the theoretical part.

We also do an analytical comparison of our scheme with those of Milstein and Dicarlo.

1.5 Outline of the Thesis

In this thesis we start in chapter 2 by surveying several basic search techniques of the classic SS code acquisition.

In chapter 3 we optimize the mean acquisition time with respect to all parameters involved in the recently proposed parallel search of Elhakeem and Shawn [2].

We use chapter 4 to discuss the implementation and practical testing results of the detection circuit described in chapter 3. A physical system has been actually built and tested.

Chapter 5 is the proposition and analysis of the hybrid parallel/sequential scheme for combined acquisition/search lock. It is

the dynamic version of the system in chapter 3 and 4.

In chapter 6 we compare the scheme of chapter 5 with those of Milstein [3] and DiCarlo [4]. And finally, in chapter 7 we outline the conclusion from this thesis, and suggested topics of future research.

CHAPTER II

A BASIC SERIAL SEARCH TECHNIQUE FOR SS CODE ACQUISITION

2.1 Single dwell serial search method

Here each of the possible code phases of the uncertainty region will be examined for a fixed period of time (dwell time, integration time) in serial fashion until the correct one has been located and verified by successful tracking following acquisition.

The model of a single dwell search PN acquisition system is illustrated in Fig. 2.1. The received signal plus noise will be actively correlated with the local replica of the PN code. After passing through a band-pass (predetection) filter and square law envelope detector the resulting waveform will be integrated for a fixed time duration τ_d (the "dwell time") and in an integrate - and - dump circuit (post-detection) and then compared to a preset threshold.

For analysis purpose the system is modeled as a Markov chain process.

If the (I & D) output exceeds the preset threshold, then a "hit" is declared. If this hit represents a true hit (i.e. the correct code phase was really examined) then the system has officially acquired the code and the search comes to an end. If the hit is a

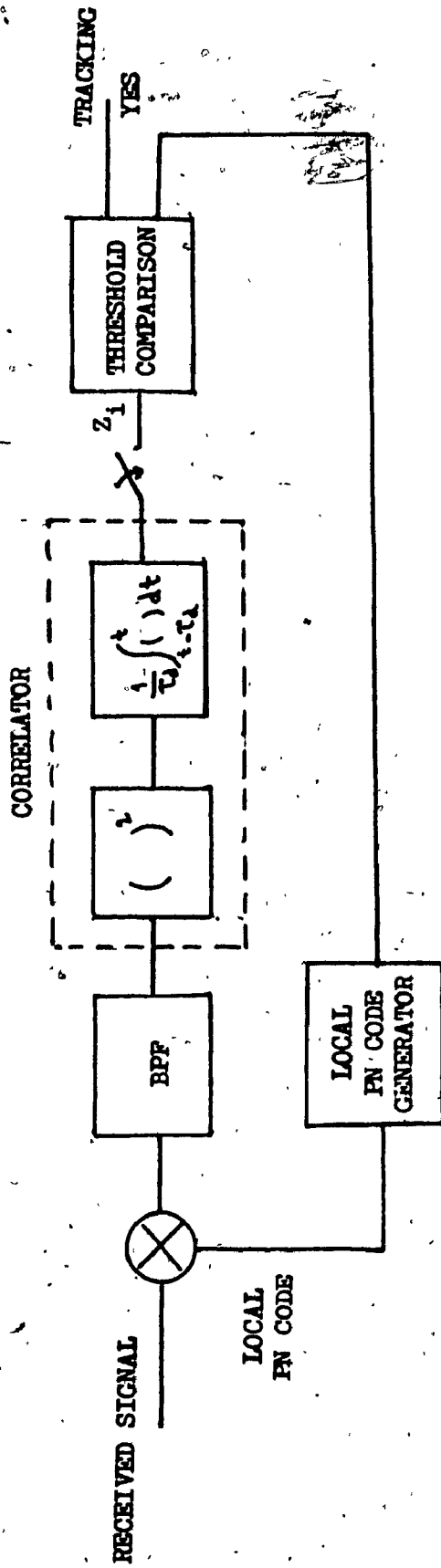


FIG. 2.1 MODEL OF A SINGLE DWELL SEARCH PN ACQUISITION SYSTEM

false alarm then code tracking will fail and the search must continue. However the time $K\tau_c$ paid for the time spent in unsuccessful code tracking must be considered in the analysis.

If the I & D output does not exceed the preset threshold then local PN code generator is shifted into its next position and the search proceeds.

We define P_D and P_{FA} as the probability of correct detection and probability of false alarm, respectively of the system.

Assuming that the received and local PN code signals are to be aligned to within one half a code chip period ($\tau_c/2$) before relinquishing (passing) the control to the fine synchronization (tracking) system. If $T_u = N_u\tau_c$ is the time uncertainty to be resolved then $q = 2N_u$ (τ_c is the chip duration) would be the number of possible code alignments and is referred to as cells.

In the absence of a priori knowledge concerning the relative code phase positions of the received and locally generated codes, the local PN generator is assumed to start the search at any code phase position with equal probability.

The a priori probability of alignment P_k is assumed as

$$P_K = \frac{1}{q - 1 - K} \quad (2.1)$$

assuming the signal is not present in the first $K-1$ cells.

A generating function flow graph for the q-state Markov chain which characterizes the acquisition process of the SDS is illustrated in Fig. 2.2. Each branch in the flow graph is labelled with the product of the transition probability associated with going for the mode at the originated end of the branch to the mode at its terminating end and is an integer (including zero) power of a parameter denoted here by Z. The parameter Z is used to mark time as one proceeds through the graph and its power represents the number of time units (dwell times) spent in traversing that branch. Furthermore, note that the sum of the branch probabilities (letting $Z = 1$) emerging from each mode equals unity.

Using standard signal flow graph reduction techniques [10] - [12], one can show that the generating function for the flow graph in Fig. 2.2 is given by [1].

$$U(z) = \frac{(1 - \beta) z}{1 - \beta z H^q(z)} \left[\frac{1}{q} \sum_{\ell=0}^{q-1} H^\ell(z) \right] \quad (2.2)$$

where

$$\beta = 1 - P_D \quad (2.3)$$

$$H(z) = P_{FA} z^{k+1} + (1 - P_{FA}) z \quad (2.4)$$

The mean acquisition time TACQ is obtained by differentiating $U(Z^q)$ with respect to Z and evaluating the result at $Z = 1$. After some algebra we have

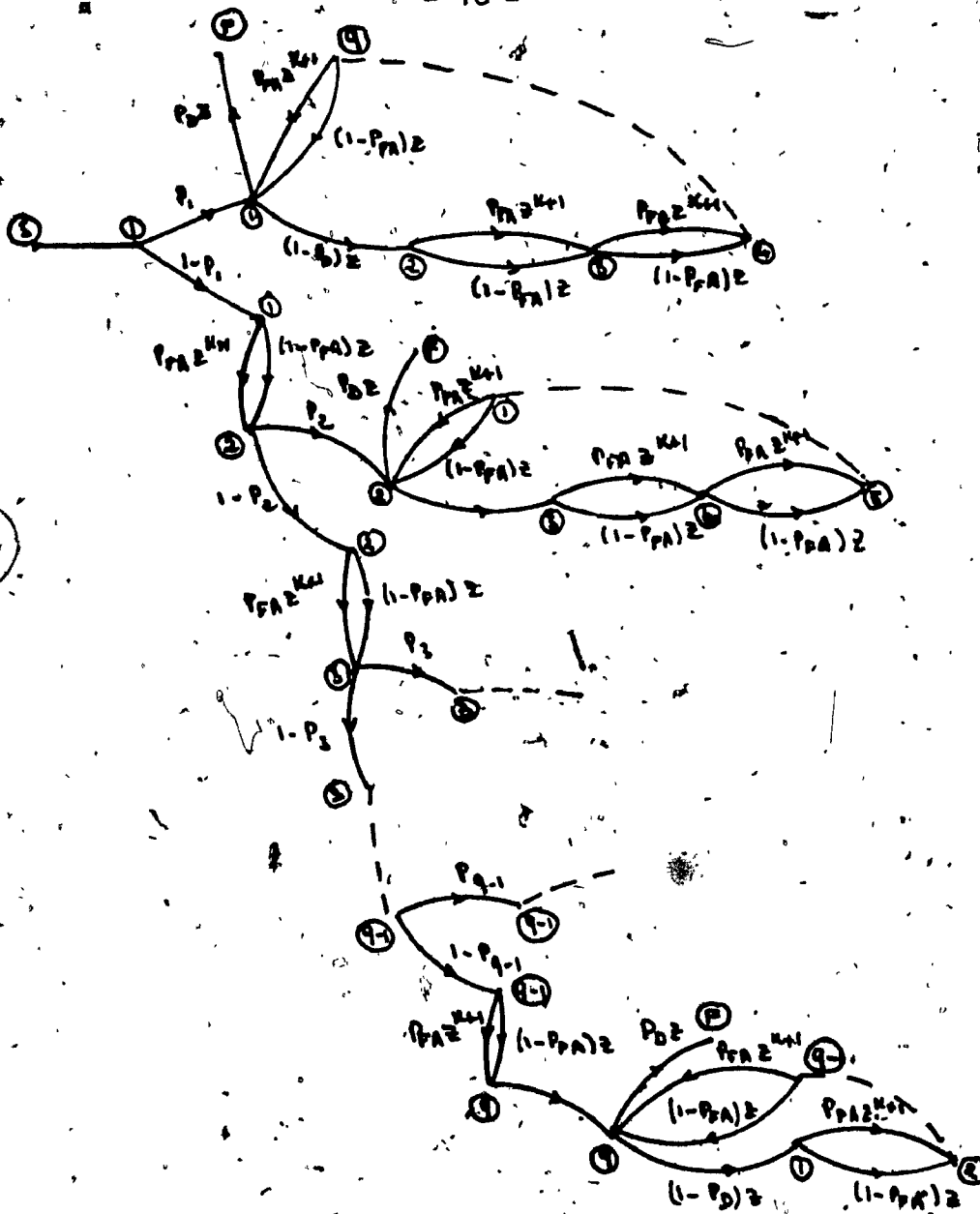


FIG. 2.2 : GENERATING FUNCTION FLOW DIAGRAM OF THE SINGLE DWELL SYSTEM

$$\bar{T}_{ACQ} = \left. \frac{d \ln U(z^{\tau_d})}{dz} \right|_{z=1} \quad (2.5)$$

$$= \frac{2 + (2-P_D)(q-1)(1-KP_{FA})}{2P_D} \tau_d \quad (2.6)$$

For $q \gg 1$

$$\bar{T}_{ACQ} = \frac{(2-P_D)(1-KP_{FA})}{2P_D} (q\tau_d) \quad (2.7)$$

The variance of the acquisition time is

$$\sigma_{ACQ}^2 = \left[\frac{d^2 U(z^{\tau_d})}{dz^2} + \frac{d U(z^{\tau_d})}{dz} - \left(\frac{d U(z^{\tau_d})}{dz} \right)^2 \right] \Big|_{z=1} \quad (2.8)$$

Since $U(1) = 1$, we have

$$\sigma_{ACQ}^2 = \left[\frac{d^2 \ln U(z^{\tau_d})}{dz^2} - \frac{d \ln U(z^{\tau_d})}{dz} \right] \Big|_{z=1} \quad (2.9)$$

For $q \gg 1$ and $K \ll q$

$$\sigma_{ACQ}^2 = \tau_d^2 (1-KP_{FA})^2 q^2 \left(\frac{1}{12} - \frac{1}{P_D} - \frac{1}{P_D} \right) \quad (2.10)$$

2.2 Multiple dwell technique

In the following description of multi dwell search, we will only concentrate on the expressions for the acquisition time and standard deviation being the most relevant performance criteria.

The multidwell technique is the generalization of the single dwell serial search technique. In this scheme the examination interval per cell does not have to be a constant but consists of a series of fixed short dwell periods (each larger than its predecessor) with a decision being made after each dwell. This permits dismissal of an incorrect alignment earlier than would be possible in a single dwell system. This happens because the integration time in a given cell examination interval is started with a smaller value than the maximum and is allowed to increase toward the maximum in discrete steps as per the above. Since most of the cells searched indeed correspond to incorrect alignments, this ability to quickly discard them produces a considerable reduction in acquisition time, particularly for long codes.

A block diagram of a typical N-dwell serial synchronization system is illustrated in Fig. 2.3. There are N detectors each is characterized by a detection probability P_{D_i} , a false alarm probability P_{FA_i} and a dwell time τ_{d_i} ... The dwell time for each non-coherent detector is assumed to be ordered such that

$$\tau_{d_1} \leq \tau_{d_2} \leq \tau_{d_3} \leq \dots \leq \tau_{d_n}$$

The decision to continue or stop the search at the present cell is made by sequentially examining the N detection outputs (starting with the first) and follows the following algorithm.

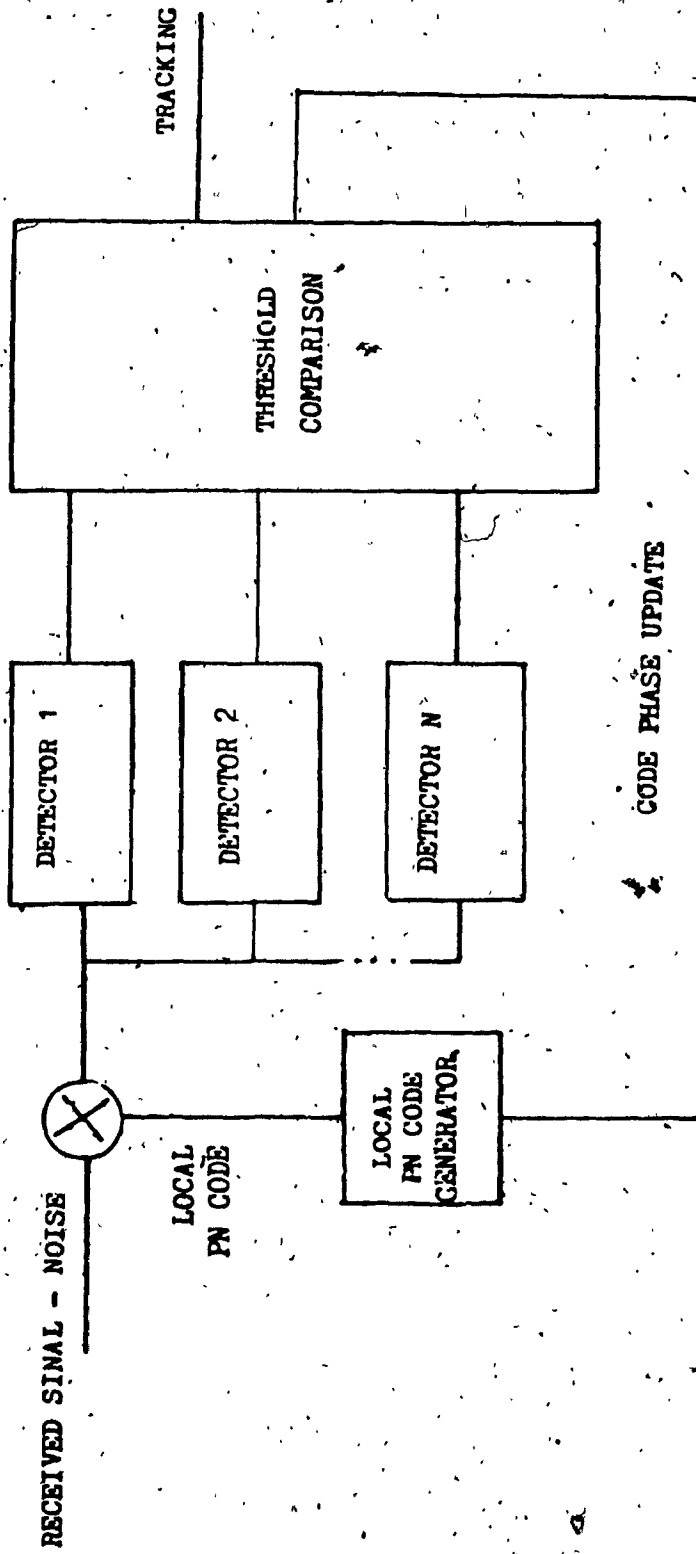


FIG. 2.3 : MODEL OF A N-DWELL SERIAL SEARCH PN ACQUISITION SYSTEM

- If all the N detectors indicate that the present cell is correct then the decision is made to stop the search.

- If any one detection fails to indicate that the present cell is correct i.e. it does not exceed the threshold, then the decision is made to continue the search and the time delay $\bar{\tau}$ of the local PN generator is retarded by the chosen phase update increment. Thus as soon as one detection indicates a misalignment the search may move on without waiting for the decision of the other remaining detection.

Thus the maximum search time for a given cell is τ_{dn} , while the minimum is τ_d . Most of the cells can be dismissed after an integration (dwell) time τ_{dk} ; $k \ll N$, whereas in the single dwell system it is the maximum τ_{dn} .

The system uses N integrate and dump circuits initiating their integration at the same instant in time, each one dumping, however at a later and later time instant in view of above the probability that Z_i crossed its threshold depends on the probability that Z_k , $k = 1, 2, \dots, i-1$ crossed their respective threshold. In accordance with the search update algorithm the output of the integrate and dump is sampled and compared to a threshold only if all of the previous $(i-1)$ integrate and dump outputs have previously exceeded their respective thresholds otherwise, the first integrate

and dump output does not exceed its threshold causes the local code to update its phase and search next cell - thereby resetting all of the I & D circuits.

As in the single dwell case a penalty T_p is specified for a false alarm. This occurs when all N detector outputs exceed their respective thresholds for a cell which does not correspond to the correct code alignment. It is convenient to model T_p as an integer multiple of the additional time required by the N th dwell, i.e.

$$T_p = K_N (\tau_{dN} - \tau_{d,N-1}) \quad (2.11)$$

The flow graph representation of the N -dwell system is illustrated in Fig. 2.4.

Analogous to Fig. 2.2 each branch is labelled with the product of the transition probability associated with going from the mode at the originating end of the branch to the mode at its terminating end and an integer (including zero) power of a parameter Z_i , $i = 1, 2, \dots, N$.

Z_i , $i = 1, 2, \dots, N$ represents the additional dwell time one must wait before testing the i th threshold after the $(i-1)$ -st threshold has been tested.

The detection transition probability $P_{D_i/i-1}$; $i = 1, 2, \dots, N$ is expressed as

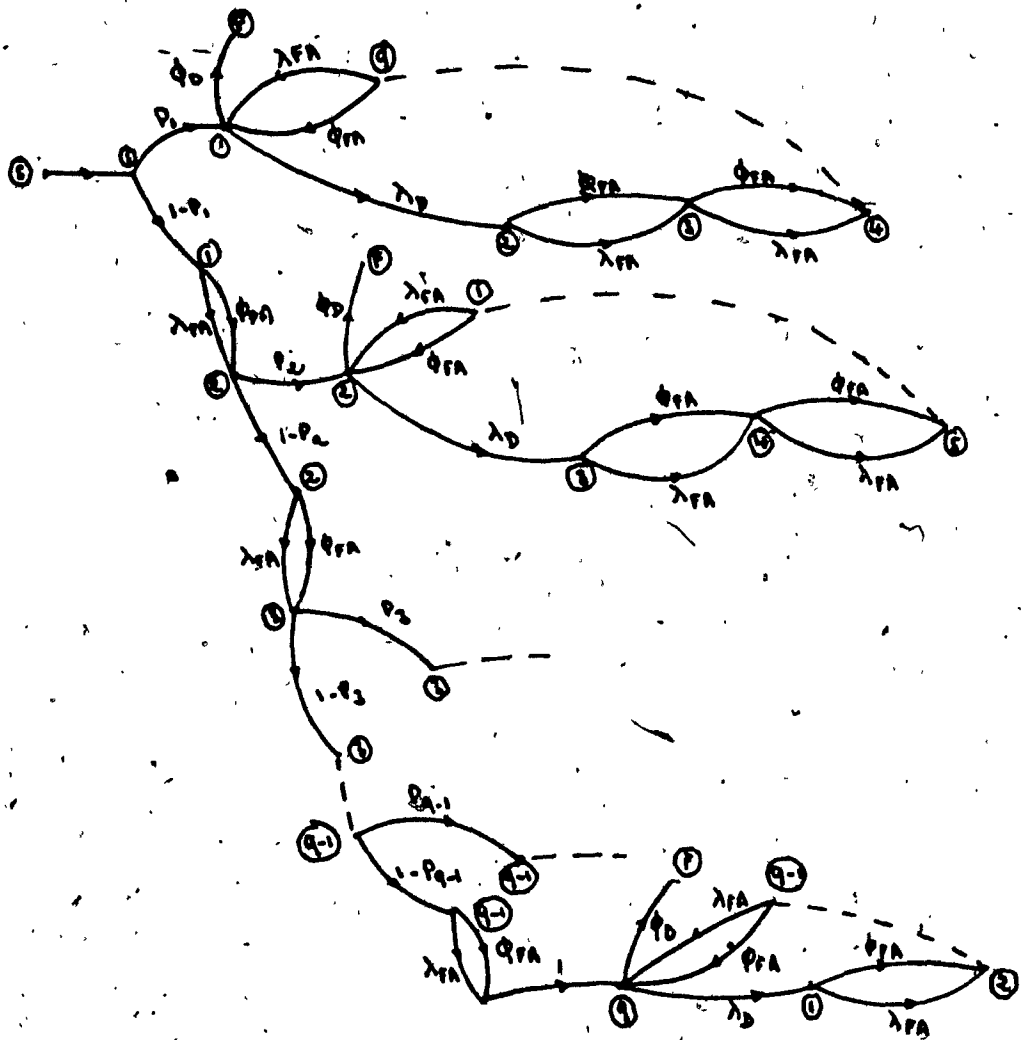


FIG. 2.4 : GENERATING FUNCTION FLOW DIAGRAM OF THE N-DWELL SYSTEM

$$P_{Di|i-1} = P_r \left\{ z_i > \eta_i \mid z_1 > \eta_1, z_2 > \eta_2, \dots, z_{i-1} > \eta_{i-1} \right\} \quad (2.12)$$

where $i; i = 1, 2, \dots, N$ denotes the i th threshold for cell containing noise only the false alarm transition probability $P_{FAi|i-1}$ is identically defined.

Referring to Fig. 2.5a) the probability P_D associated with traversing the entire branch is the probability that all the N -integrate-and-dump outputs exceed their thresholds or

$$\begin{aligned} P_D &= \prod_{i=1}^N P_{Di|i-1} \\ &= \Pr (z_1 > \eta_1, z_2 > \eta_2, \dots, z_N > \eta_N) \end{aligned} \quad (2.13)$$

From Fig. 2.5b) the probability corresponds to the event of causing a code phase update will be the probability of any of the $N(I \& D)$ outputs fails to exceed its threshold as one progress through the N -dwell system.

$$\begin{aligned} P_{\lambda_L} &= \sum_{i=1}^N \left(\prod_{k=1}^{i-1} P_{Dk|k-1} \right) (1 - P_{Di|i-1}) \\ &= \Pr (z_1 < \eta_1 \text{ or } z_2 < \eta_2 \text{ or } \dots \text{ or } z_N < \eta_N) \end{aligned} \quad (2.14)$$

Similarly Fig. 2.5c) and 2.5d) give the system probability of false alarm.

$P_{\lambda_{FA}}$ and probability of no false alarm

$P_{\lambda_{\bar{FA}}}$ is clearly the system false alarm

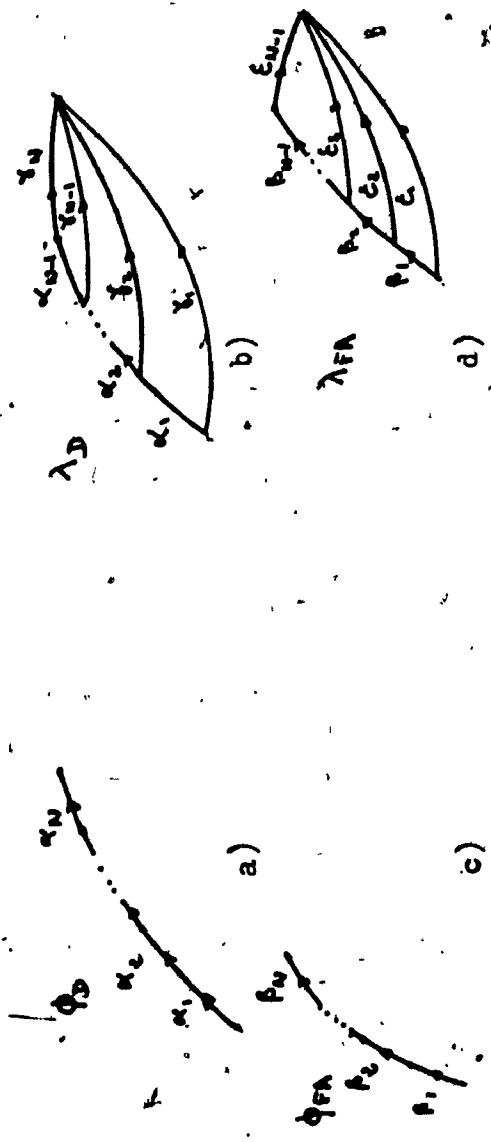


FIG. 2.5 : FLOW DIAGRAM TO COMPUTE ϕ_D , λ_D , ϕ_{FA} , λ_{FA}

$$P_{FA} = \prod_{i=1}^N P_{FAi|i-1} = \Pr (Z_1 > \eta_1, Z_2 > \eta_2, \dots, Z_N > \eta_N) \quad (2.15)$$

whereas

$$P_{FA} = \sum_{i=1}^N \left(\prod_{k=1}^{i-1} P_{FAk|k-1} \right) (1 - P_{FAi|i-1}) \Pr (Z_1 < \eta_1 \text{ or } Z_2 < \eta_2 \text{ or } \dots \text{ or } Z_N < \eta_N) \quad (2.16)$$

Using the same standard flow graph reduction techniques [10] - [12] the N-dimensional generating function for the N-dwell system is [1]

$$U(z) = C(z) \left[\frac{1}{q} \sum_{\ell=0}^{q-1} H^\ell(z) \right] \quad (2.17)$$

The mean acquisition time is given by,

$$\bar{T}_{ACQ} = (2 - P_D) q \sum_{j=1}^N \left[\prod_{i=1}^{j-1} P_{FAi|i-1} (\tau_{dj} - \tau_{d,j-1})^{K - P_{FA}} \delta_{jN} (\tau_{dN} - \tau_{d,N-1}) \right] \quad (2.18)$$

where δ_{ij} is the Kronecker delta function

$$\delta_{ij} = \begin{cases} 1; & i = j \\ 0; & i \neq j \end{cases} \quad (2.19)$$

The single dwell system is a specific case of the multi dwell system is a specific case of the multi dwell technique when $N = 1$ and $K_1 = K$, (2.18) reduces to (2.8) as it should.

An approximate expression for the acquisition time variance σ_{ACQ}^2 of the N-dwell system is derived as

$$\sigma_{ACQ}^2 = \sum_{i=1}^N \sum_{l=1}^N \frac{\partial^2 U(z^* \tau_d)}{\partial z_l \partial z_l^*} \Big|_{z=1} + \bar{T}_{ACQ} (1 - \bar{T}_{ACQ}) \quad (2.20)$$

For large q

$$\sigma_{ACQ}^2 = q^2 \left\{ \sum_{j=1}^N \left[\prod_{i=1}^{j-1} P_{FAi|i-1} \right] (\tau_{dj} - \tau_{d,j-1}) \right. \\ \left. + K_N P_{FA} \delta_{jN} (\tau_{dN} - \tau_{d,N-1}) \right\} \left(\frac{1}{12} - \frac{1}{P_D} - \frac{1}{P_D} \right) \quad (2.21)$$

Again for $N = 1$ and $K_1 = K$, (2.21) reduces to (2.10) for $q \gg 1$.

Comparing we see that if

$$\sum_{j=1}^N \left(\prod_{i=1}^{j-1} P_{FAi|i-1} \right) (\tau_{dj} - \tau_{d,j-1}) < \tau_d \quad (2.22)$$

is satisfied, the N dwell search yields a smaller acquisition time variance than the single dwell system.

In fact for large q , both T_{acq} and the standard deviation σ_{ACQ} are directly proportional to the same function $F(N)$ of false alarm probabilities and dwell times.

$$F(N) = \sum_{j=1}^N \left[\left(\prod_{i=1}^{j-1} P_{FAi|i-1} \right) (\tau_{dj} - \tau_{d,j-1}) + K_N P_{FA} \delta_{jN} (\tau_{dN} - \tau_{d,N-1}) \right] \quad (2.23)$$

2.3 Acquisition using matched filter techniques

In these techniques a passive correlator device such as a matched filter (MF) is used. This device can be implemented either as a continuous time or discrete time.

In the continuous time case which we focus in, the received PN waveform plus noise is convolved with a fixed finite segment of the PN wave-form (M chips) and the continuous output is compared with a threshold to determine if acquisition has occurred. In this configuration, the input continuously slides past the stationary (not running in time) stored PN replica until the two are in synchronism, at which point the threshold would be exceeded and the local PN generator enabled.

A matched filter for a finite length PN waveform can be seen in the form of a tapped delay line (TDL) designed to match a single PN chip wave-form with a passive filter which follows.

Using the diagram of Fig. 1.1, where the matched filter acquisition system is a special case.

The mean acquisition time is for the worse case in this sense derived as

$$\bar{T}_{ACQ} = P_{ACQ} \left\{ (K+1) P_D + \bar{T}_M P_{ACQ} + (V-1) \bar{T}_O \frac{P_D + P_{ACQ}(1-P_D)}{1-P_{FA}} \right\} \tau_d \quad (2.24)$$

Where N is the number of state

P_{acq} is the probability of acquiring after any number of dwells

$$P_{ACQ} \triangleq A_{ACQ}(\sigma) = U(1) = \sum_{i=0}^{\infty} P_i \quad (2.25)$$

$$\bar{T}_0 = (1 - P_{FA0}) + (K+1)P_{FA0}(1 - P_{FA1}) \quad (2.26)$$

$$\bar{T}_M = (1 - P_{D0}) + (K+1)P_{D0}(1 - P_{D1}) \quad (2.27)$$

One can model the quadrature total noise components at the matched filter output as independent gaussian random variable with variance.

$$\sigma_i^2 = \frac{N_0}{2} \left[1 + \frac{E_c}{N_0} G_i(\xi) \right], \quad i=0, 1 \quad (2.28)$$

where $G_i(E)$ is the normalized variance of the partial correlation.

CHAPTER III

OPTIMAL SEQUENCE DETECTION CIRCUIT APPROACH

TO SPREAD SPECTRUM CODE ACQUISITION

3.1 Introduction

There has been a growing interest lately in using multi-dwell [3] and parallel acquisition techniques [4] for minimizing the acquisition time of spread spectrum code. This was recently complemented by a parallel processing technique [2] for direct sequence (DS) and (FH) codes acquisition. The mean acquisition times were shown to be less than those of multi dwell techniques. However, it was assumed among other assumptions that the number of necessary tapped analog delays (N) equals the number of synchronization detection circuits (NN). In this thesis we relax this assumption and free all parameters with the final objective of minimizing the mean acquisition time.

The total code length L will be fixed and at certain N (thus fixing the number of bits in each TAD (M)) the number NN and the dismissal threshold (TH) will be varied.

Of relevance to computing the mean and variance of the acquisition time is the bit probabilities P_b, P_b' . These are defined as the intrinsic DS bit error probability as reflected by the channel conditions in the two cases of code presence or absence respectively.

For all values of (M), the analysis algorithm will substitute the same values for P_b, P_b' thus guaranteeing fair comparisons among all cases of differing M .

3.2 The parallel sequence detection circuits code acquisition method

The technique presented in [2] is based on the use of tapped analog delay lines (TADs) as matched filters for acquisition of a direct sequence (DS) SS signal. The total DS code (L bits) is divided into N segments of equal length (M bits). Each segment will be detected by one of the (N) TADs. The number of chips (bits) per TAD (M), is related to the total code length L and the number of TADs N by,

$$M = \frac{L}{N} \text{ (in units of chips)}$$

Variations of M will affect the mean acquisition time as we will see later in this paper. Using N TADs (see Fig. 3.1), each matched to one of N segments of the code length, the entire code length will be detected once the right sequence of TADs output peaks. Each TAD (which can be replaced by programmable SAW devices for higher frequencies) is composed of M taps to be matched with M bits of a segment of the code length. The taps of each TAD are weighted (to either one or zero) to match the sequence of M chips of a segment of the code length. Every time there is an alignment of the received segment with the taps in the TAD, the TAD (with few logic extras) will give a peak pulse output.

The TADs are numbered TAD₁, TAD₂, TAD₃, ..., TAD_N according to the number of the segments in the code it is supposed to detect. The sequence detection circuits (SDCs) system is used to detect the in-sequence peakings of the TADs.

If the code received is the right code, each of its N segments will be detected by the proper TAD. Thus, the peaking of the TADs will be in sequence (i.e.; $TAD_1 - TAD_2 - TAD_3 - \dots - TAD_N$). The SDC (first SDC) will detect this sequence of pulses coming from the N TADs and will signal an output pulse thus declaring synchronization. However, a band of SDCs is used. The first (SDC_1) will be programmed to detect the unique sequence

1 → 2 → 3 → 4 → ... → N

The second SDC (SDC_2) will be programmed to detect

2 → 3 → 4 → ... → N

The third SDC (SDC_3) will be programmed to detect

3 → 4 → ... → N

and so forth. The N th SDC (SDC_N) will detect only the peaking of TAD_N .

1, 2, 3, 4, ..., N above refers to $TAD_1, TAD_2, TAD_3, TAD_4, \dots, TAD_N$ correctly matching the received input signal segments. It is easily seen that NN , the number of SDCs cannot be larger than N (the number of TADs) but could be less. If one or more SDCs successfully detect their respective sequences, acquisition will be declared and the system will move to the tracking phase.

For illustration purposes (see the flow chart of Fig. 3.2), we take a code-length composed of 8 divisions and assume that a certain portion of the received signal in noise and jamming have resulted in the following sequence of TADs peaking: 3 → 1 → 2 → 4 → 5 → 1 → 7 → 8. In this case, SDC_1 starts counting at the 2nd received

division, goes to the waiting state at the 3rd, starts again at the 6th and goes to wait state at the 7th. SDC₂ starts at the 3rd and goes to wait state at the 4th (having received 4 not 3). SDC₃ starts at the first and goes to wait at the 2nd. SDC₄ starts at the 4th and goes to wait at the 6th. SDC₅ starts at the 5th and goes to wait at the 6th. SDC₆ will not respond to this sequence and will remain in wait state (no 6 received). SDC₇ and SDC₈ both receive their anticipated sequences and both declare synchronization at the right time. It is seen from this typical example that within 8 division (i.e. one codelength) the code was acquired in spite of the mess that the jamming and noise have caused. The key issue here is bridging the gap between parallelism and sequential detection. Going back to the example, we see that SDC₁ waited for 1 to peak, he got it in the 2nd division, lost count, regained again at the 6th and finally lost and went to the wait state at the 7th. Similar situations happened to SDC₂, SDC₃, SDC₄, SDC₅, but finally because of parallelism of action, SDC₇, SDC₈ succeeded in declaring correct synchronization. For this specific example other multi dwell techniques [3] will skip the specific search cell corresponding to the above received sequence (having one synchronization circuit rather than 8 as in our system). Our technique effectively reduces the uncertainty region of the received code phase. It is similar to [5] only in that sense but different in the way of assumptions and implementation. While [5] assumes that synchronization has just been lost to reduce the code uncertainty

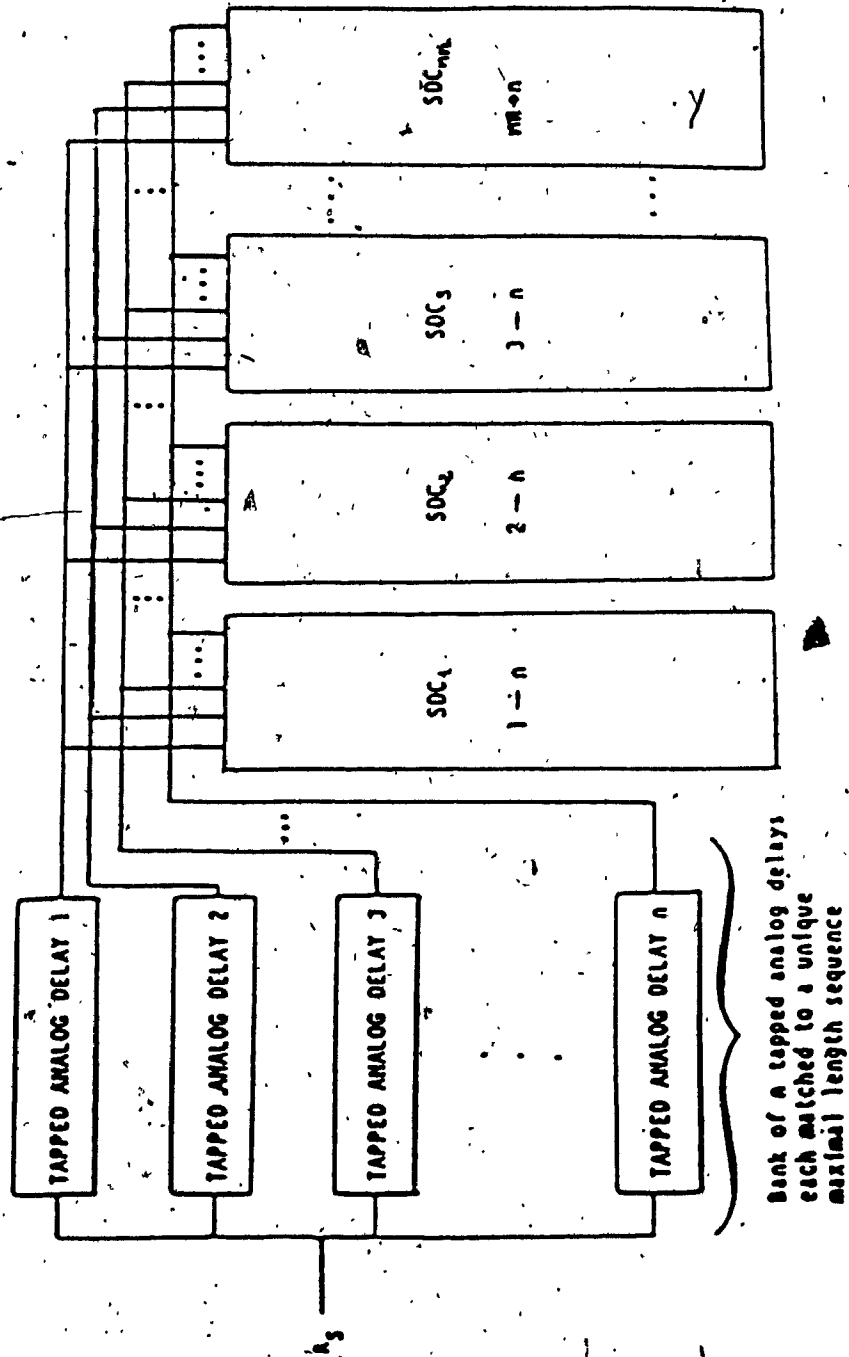


FIG. 3.1 : CONFIGURATION OF n-TAPPED ANALOG DELAYS AND n-M-SEQUENCE DETECTION CIRCUITS

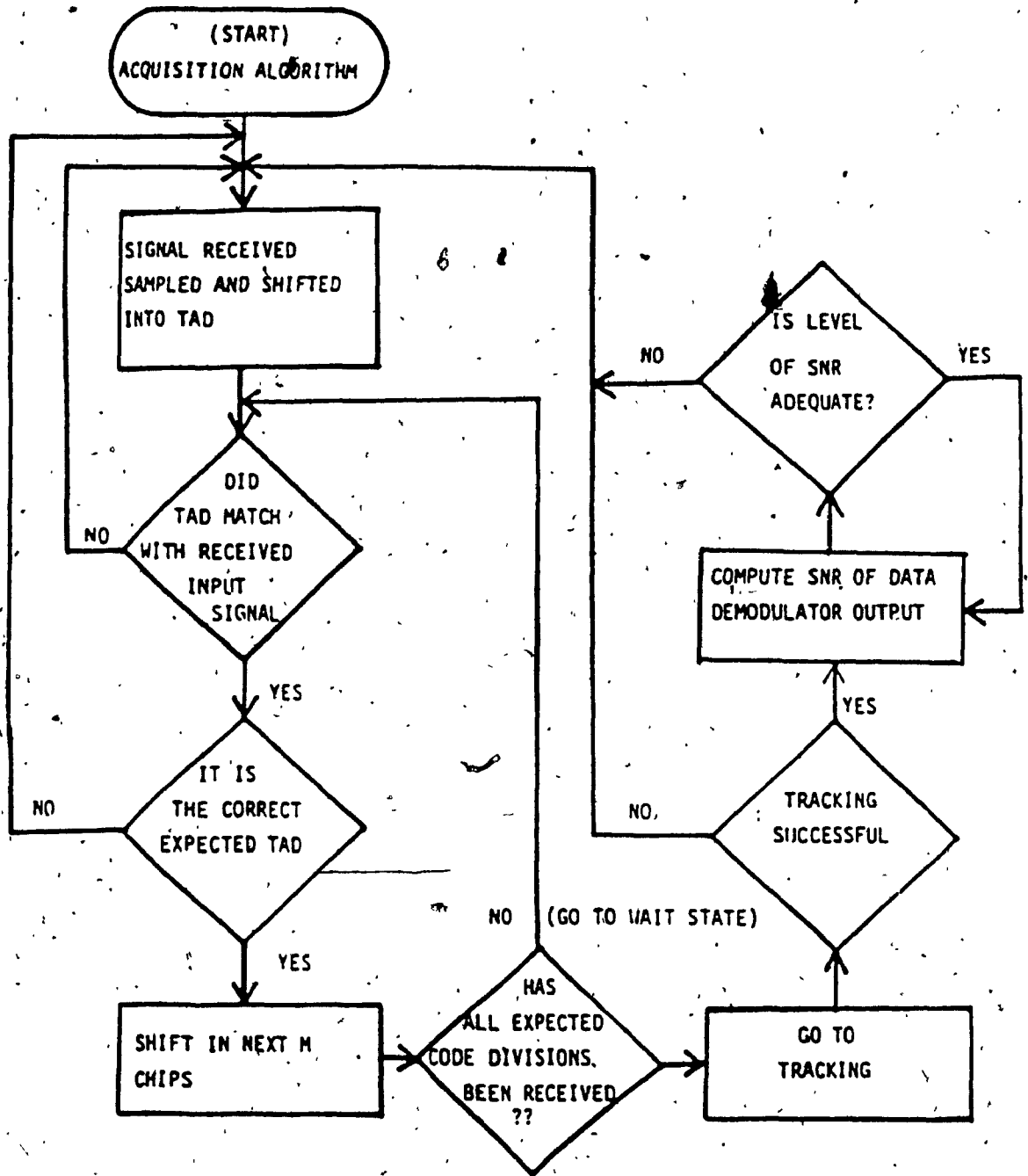


FIG. 3.2 : FLOW CHART OF THE ACQUISITION TECHNIQUE (ACTION TAKEN BY ONE OF THE SOC'S AND OTHER TRACKING AND DEMODULATOR CIRCUITS).

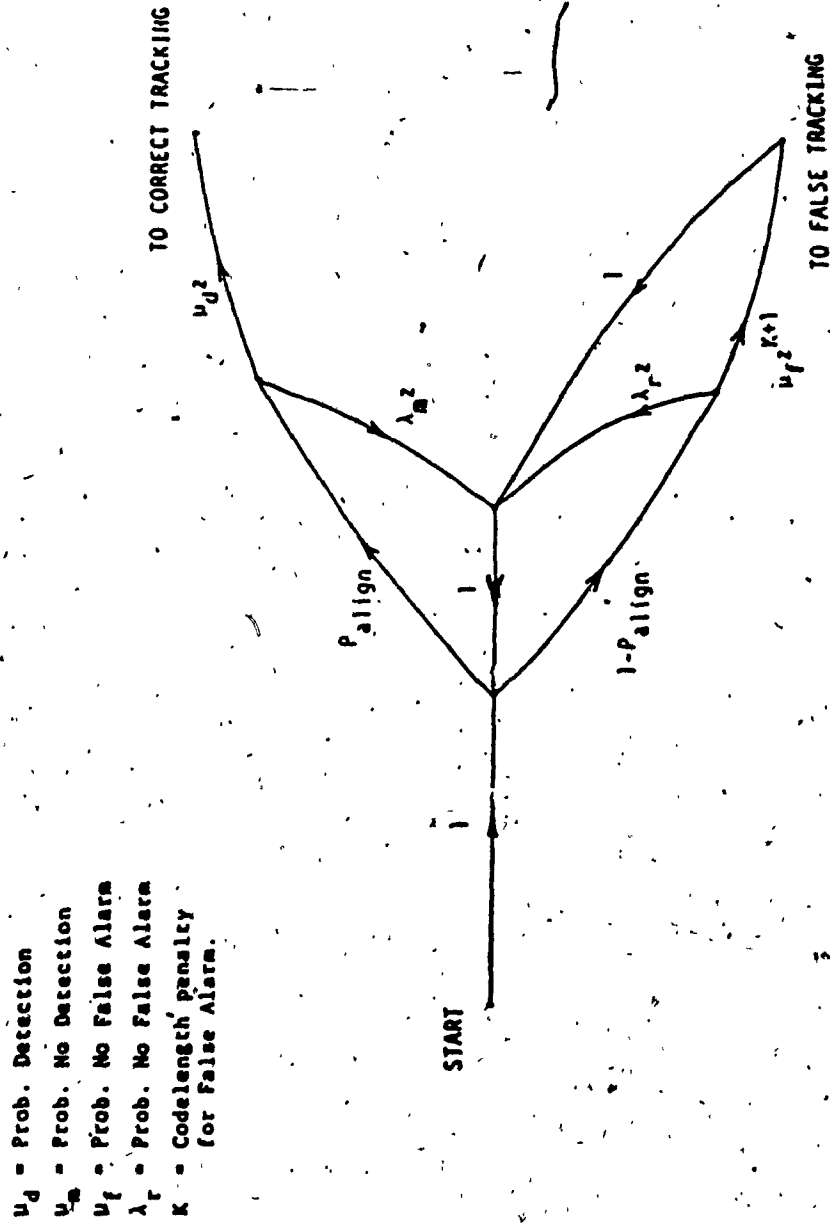


FIG. 3-3: FLOW DIAGRAM REPRESENTATION OF THE ACQUISITION SCHEME

region is reduced by having so many SDCs each sensitive to a specific code phase (start).

3.3 Analysis

Similar to [2] the flow diagram approach (Fig. 3.3) will be employed to obtain the equation for the mean and standard deviation of the acquisition time. The probability of alignment of a code length L is given by,

$$P_{\text{align}} = \frac{N}{L} = \frac{1}{M} \quad (3.1)$$

If we define the total system probability of detection, miss, false alarm and no false alarm as μ_d , λ_m , μ_f and λ_r respectively, together with the following definitions pertaining to TAD number i .

α_i = the probability that the i th TAD will correctly peak and be detected. (3.2)

ξ_i = the probability that the i th TAD will be aligned but will not peak (due to noise, jamming, etc.). (3.3)

β_i = the probability that the i th TAD will falsely peak (again due to noise, jamming, etc.). (3.4)

γ_i = the probability that the i th TAD will not falsely peak (3.5)
($i = 1, 2, 3, \dots, N$)

γ_i = the probability that the i th TAD will not falsely peak (3.5)

($i = 1, 2, 3, \dots, N$)

If d_j is the probability of detection of the SDC $_j$ ($j = 1, 2, 3, \dots, NN$), we have

$$\mu_d = \mu_{d1} + (1-\mu_{d1})\mu_{d2} + (1-\mu_{d1})(1-\mu_{d2})\mu_{d3} + \dots + (1-\mu_{d1})(1-\mu_{d2})(1-\mu_{d3}) \dots (1-\mu_{d_{NN-1}})\mu_{d_{NN}} \quad (3.6)$$

where the different μ_{d_j} 's are calculated as in [2] i.e.,

$$d_j = \prod_{i=j}^N \alpha_i \quad (3.7)$$

From Fig. (3.4) it follows that

$$\begin{aligned} \mu_d &= \sum_{j=1}^{NN} \mu_{d_j} \sum_{A=1}^{j-1} (1-\mu_{dA}) \\ &= \sum_{j=1}^{NN} \prod_{i=j}^N \alpha_i \prod_{A=1}^{j-1} (1 - \prod_{i=A}^N \alpha_i) \end{aligned} \quad (3.8)$$

Similarly, from Figures (3.5), (3.6), (3.7) and referring to [2] we have

$$\lambda_m = \prod_{h=1}^{NN} \sum_{g=h}^N \epsilon_g \prod_{i=h}^{g-1} \alpha_i \quad (3.9)$$

$$\mu_f = \sum_{c=1}^{NN} \prod_{i=c}^N \beta_i \prod_{B=1}^{c-1} (1 - \prod_{i=B}^N \beta_i) \quad (3.10)$$

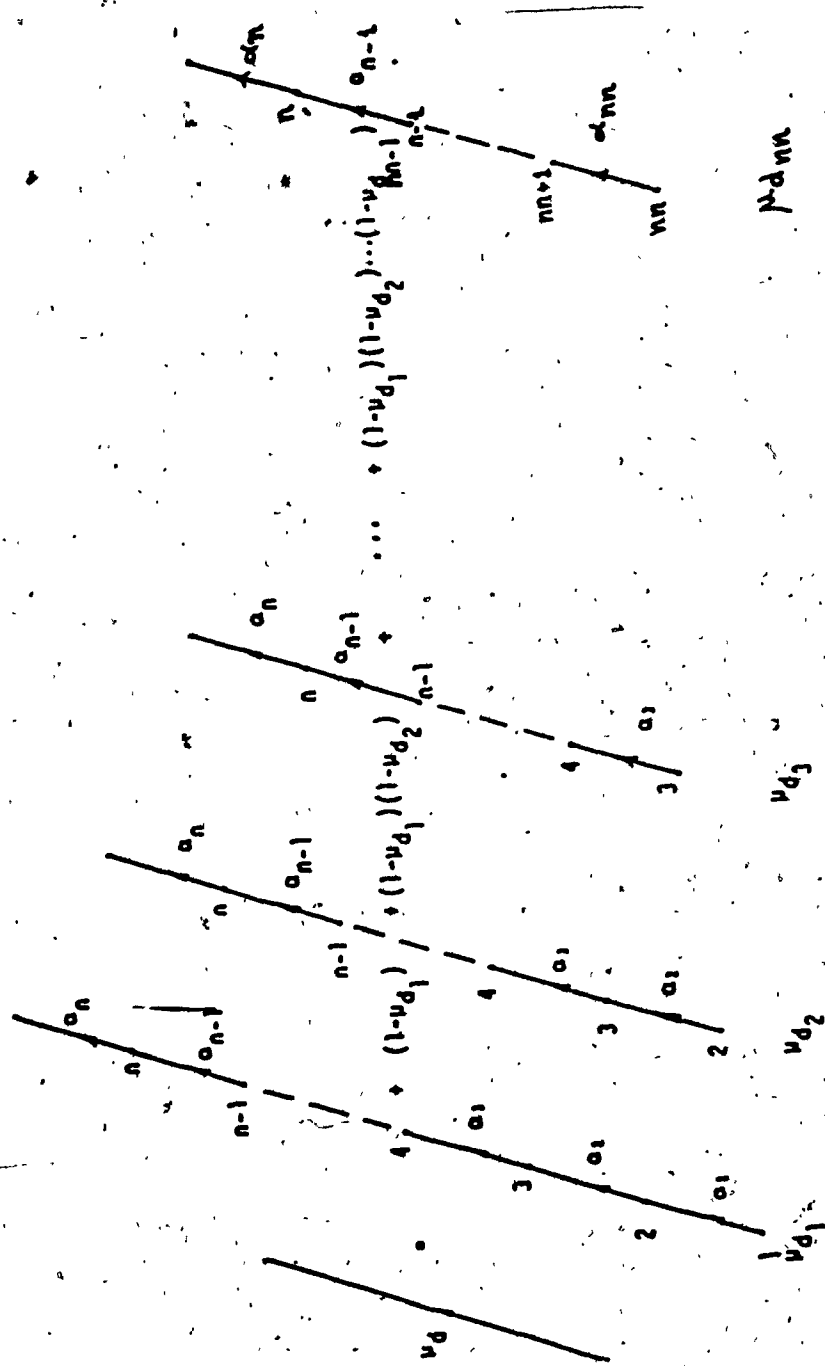


FIG. 3.4 : REPRESENTATION OF THE μ_B BRANCH ($\mu_0 \leq n$)

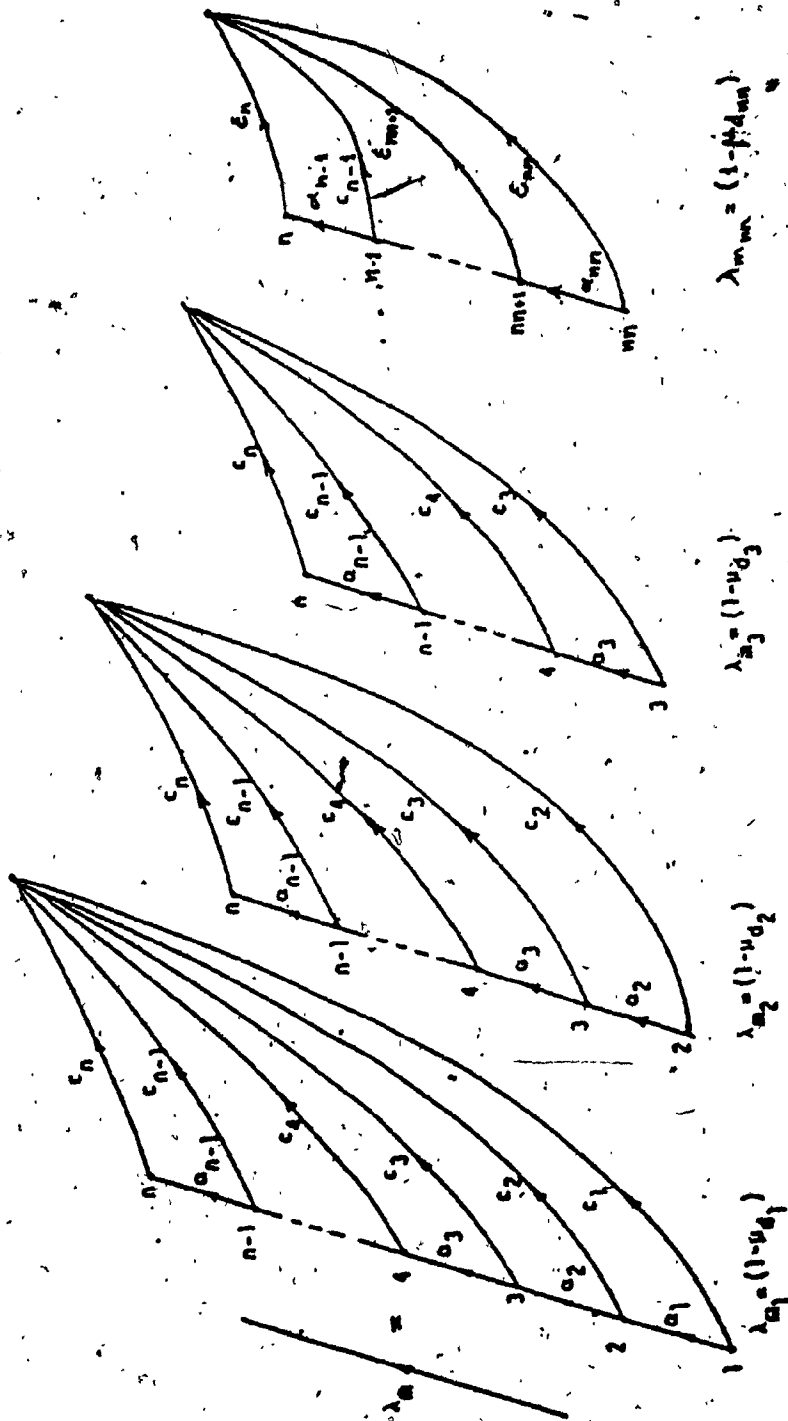


FIG. 3.5 : REPRESENTATION OF THE λ_m BRANCH ($m \leq n$)

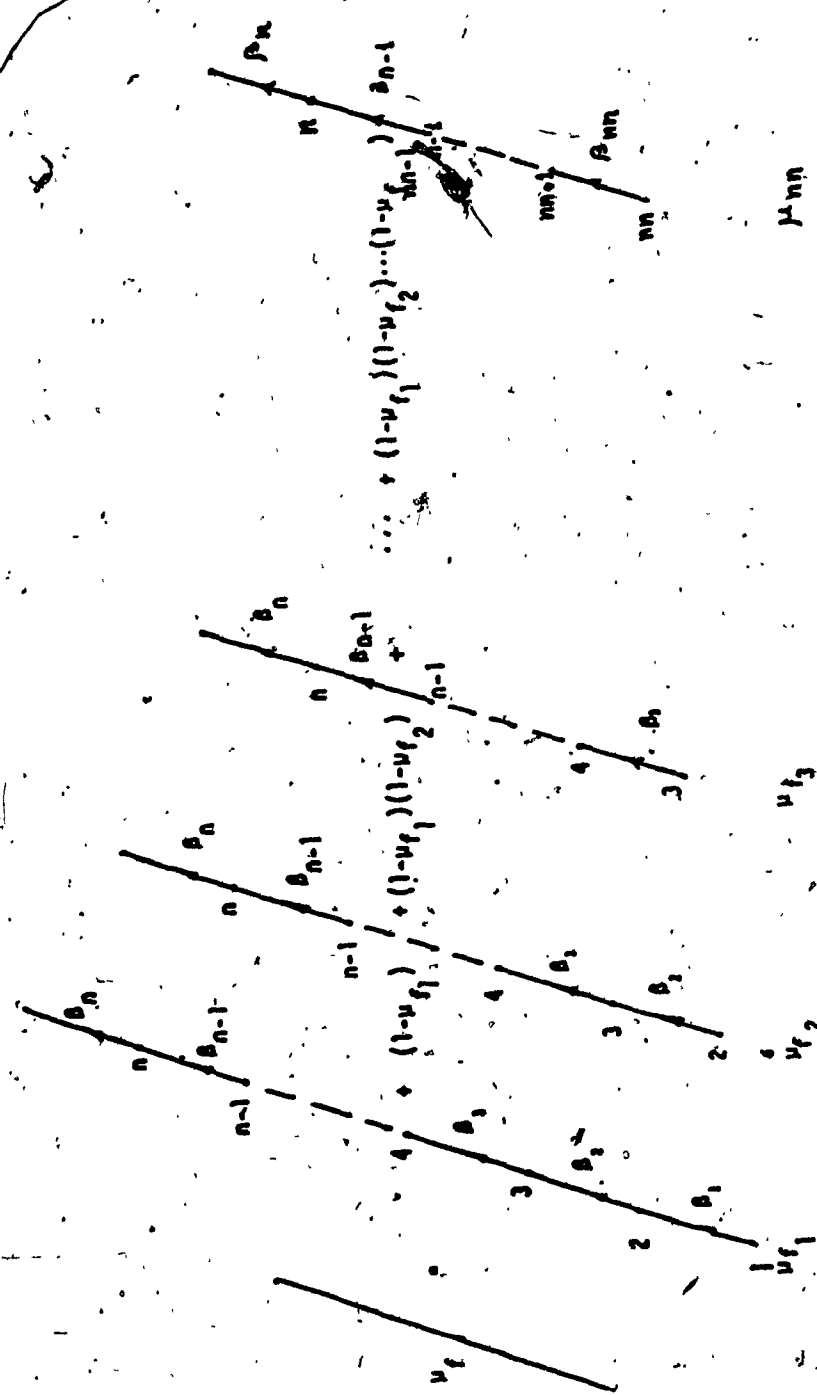


FIG. 3.6 : REPRESENTATION OF THE γ_f BRANCH ($nn \leq n$)

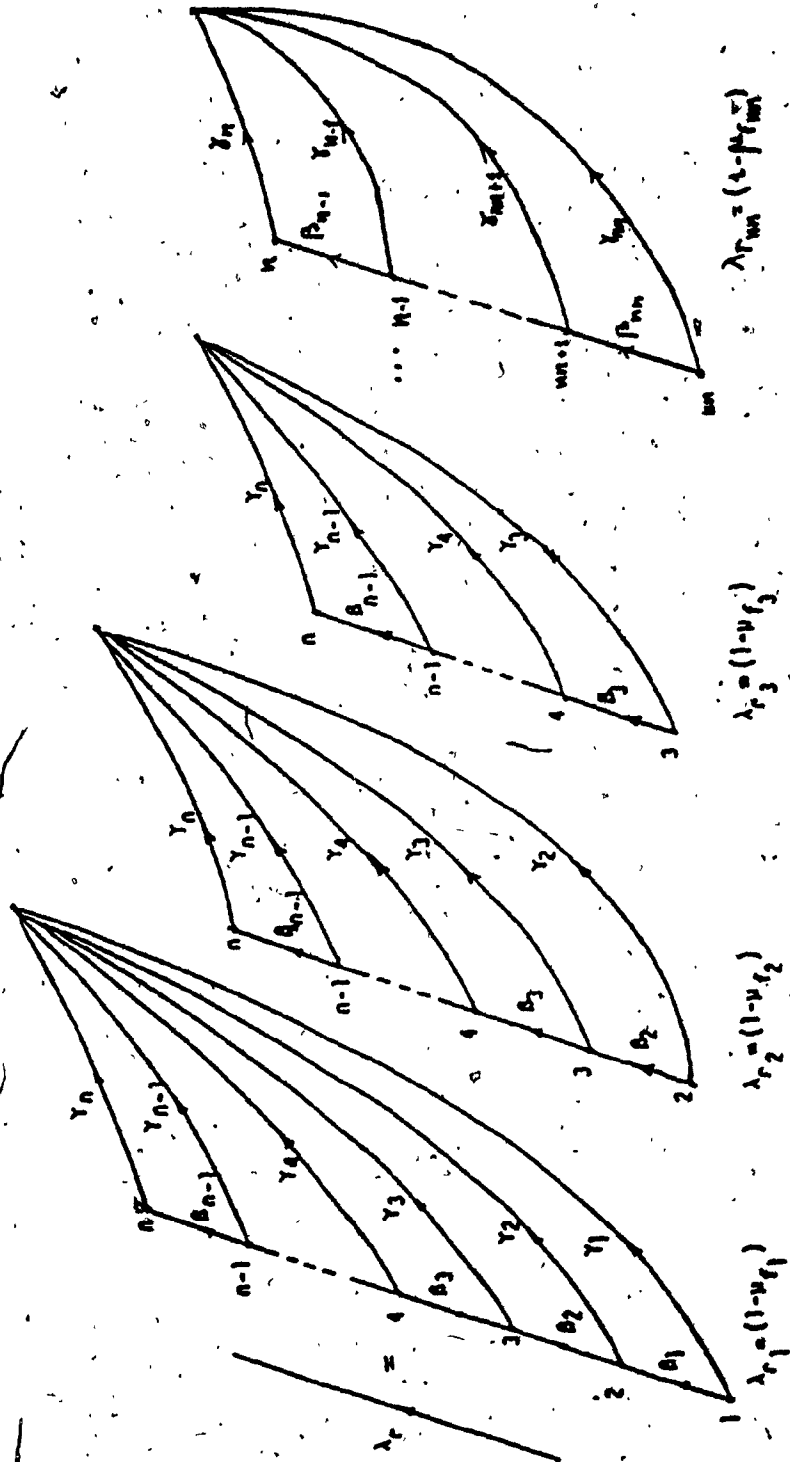


FIG. 3.7 : REPRESENTATION OF THE λ_r BRANCH ($mn \leq n$)

$$\lambda_r = \prod_{s=1}^{NN} \prod_{q=s}^N \gamma_q \prod_{i=s}^{q-1} \beta_i \quad (3.11)$$

If we assume $\alpha_i = \alpha$ and $\beta_i = \beta$ for all i , then

$$\mu_d = \sum_{j=1}^{NN} \alpha^{(N-j+1)} \prod_{A=1}^{j-1} (1 - \alpha^{(N-A+1)}) \quad (3.12)$$

$$\lambda_n = \prod_{h=1}^{NN} \sum_{g=h}^N \epsilon \alpha^{(g-h)} \quad (3.13)$$

$$\mu_p = \sum_{c=1}^{NN} \beta^{(n-c+1)} \prod_{B=1}^{c-1} (1 - \beta^{(N-B+1)}) \quad (3.14)$$

$$\lambda_r = \prod_{s=1}^{NN} \sum_{q=s}^N \gamma \beta^{(q-s)} \quad (3.15)$$

To evaluate the probabilities α and β used in (3.12) - (3.15) one needs P_b and P'_b i.e., the probability of bit errors (as reflected by the noise and interferences) in the presence and absence of the correct code respectively. Actually, the dependence on P_b and P'_b on the status of the channel, level of interferences etc., is a separate research problem and for the purposes of this paper P_b and P'_b will be assumed.

Each TAD contains M (DS) bits and we associate with each bit successful or unsuccessful detection a binary random variable taking the probabilities $(1 - P_b)$ and P_b respectively. It follows that the

M bits of each TAD will constitute a binomial distribution and if the dismissal threshold is TH we get for the probability of signal detection per TAD (α),

$$\alpha = \sum_{p=TH}^M \binom{M}{p} (1 - P_b)^p (P_b)^{M-p} \quad (3.16)$$

Similarly, in the absence of the correct code, we get the probability of false alarm per TAD as,

$$\beta = \sum_{q=TH}^M \binom{M}{q} (1 - P'_b)^{M-q} (P'_b)^q \quad (3.17)$$

Finally, by straightforward application of the signal flow graph of Fig. (3.3) and similar to [2] we get for the mean and standard deviation of the acquisition time

$$\bar{T} = \text{mean} = \frac{1 + K \mu_f (1 - P_{\text{align}})}{(P_{\text{align}} \mu_d)} \quad (3.18)$$

σ_T = standard deviation

$$= \frac{\{(1 - P_{\text{align}} \mu_d) + [K \mu_f (1 - P_{\text{align}})]^2 + K \mu_f (1 - P_{\text{align}}) (2 + K \mu_d P_{\text{align}})\}^{\frac{1}{2}}}{P_{\text{align}} \mu_d} \quad (3.19)$$

3.4 Results and conclusions

To compute the mean and standard deviation of the acquisition time from equations (3.18), (3.19) respectively, one needs the quantities M , N , NN , L , P_b , P'_b , TH . These were given different values resulting in Figures (3.8) - (3.15) and Tables 1, 2.

Figures 3.8 show the effect of NN on the mean acquisition time for a varying probability of bit errors (P_b). The existence of an optimal NN at higher values of P_b ($> .3$) (giving minimum acquisition time) is clearly depicted from Fig. 3.8a. Fig. 3.8b shows the effect of the threshold ($TH = 10 M/16$) on the shape of the acquisition time vs NN curves, the optimal NN clearly moves to the right in this case. Moreover, as TH increases (Figs. 3.8b, 3.8c, 3.8d) all curves tend to be a monotonically decreasing function of time and the optimal value of NN becomes the largest possible i.e. $NN = 8$.

Fig. 3.9a shows that the optimal NN is the lowest if P_b is fixed at .3 and P'_b takes lower values in the range (.01 - .5). The threshold TH is raised to ($10 M/16$) in Fig. 3.9b with the result that optimal values of NN now exist for those curves at which $P_b = .5, .6$. However, with $TH = 12 M/16$, $TH = 13 M/16$. (Figs. 3.9c and 3.9d) now give monotonically decreasing curves in NN for all values of P'_b . All the curves of Figs. 3.8 and 3.9 were based on the same total code length ($L = 256$ bits). Fig. 3.10 relaxes this and assumes different code lengths and different number of TADs N . It is easily seen that shorter code lengths

yield better acquisition times. Also, threshold bonds are well defined from Fig. 3.10 i.e., the acquisition times are almost constant over a certain threshold range. Fig. 3.11 shows a situation where P'_b, NN, TH are fixed and the number of TADs (N) is varied such that the total code length L is kept fixed ($L = 256$). All waves in this case are monotonically increasing functions of P_b and the optimum N is the largest ($N = 32$). However, this implies a larger system cost. However, this is not the case in Fig. 3.12 where a lower value of $N = 16$ bits yields the best results (the independent parameter is P'_b not P_b though). Fixing the code length L and also P'_b, NN, TH and varying the number of bits per TAD (M) we obtain Fig. 3.15a where an optimum value ($M = 16$) results at $P_b = 3$. We note too that the mean acquisition time monotonically increases or decreases with M at small or large values of P_b respectively. In Fig. 3.13b, P'_b, NN, TH, L are fixed, N is varied, and the existence of two optimal values for N is clearly seen. Finally, Figs. 3.14 and 3.15 show the effects of P_b, P'_b respectively on the mean acquisition time.

In all Figs. (3.8) - (3.15) one or two parameters only are allowed to change. However, if we let free all the parameters TH, NN, N at given values for L, P_b, P'_b , we obtain Tables 3.1, 3.2.

In Table 3.1a, $L = 256, N = 1, P_b$ and P'_b are given certain values and the resulting best TH , best NN and the corresponding mean and standard deviation of the acquisition time are computed and inserted together in one square of the table.

Table 3.1b is a repetition except that $N = 2$ and so are Tables 3.1c, 3.1d, 3.1e with values of $N = 4, 8, 16$ respectively. The ambitious designer will pick the results of Table 3.1f giving the minimum acquisition times. However, the minimums occur all at $N = NN = L$ which is not a very practical solution. So, the choice of the designer will be one of Tables 3.1a - 3.1i depending on his budget. Tables 3.2a - 3.2h are similar to those of Tables 3.1 except that L is now 128. Comparing all corresponding results of Tables 3.1 and 3.2 reveal the fact that shorter code lengths yield better acquisition times (note that all the results of Tables 3.1 and 3.2 are in units of bits, not code lengths and so, fair comparison between different cases prevails). The final concensus out of all these results is that the use of subdivided codes and SDC lead to reducing the acquisition time. The existence of optimal values for the parameters involved was proven and fair comparisons were guaranteed by starting at the intrinsic bit error probability P_b, P'_b .

In some cases (Tables 3.1i, 3.2h) acquisition was obtained in one code length time (when $N = NN = L$) a very logical result that proves our modeling of the acquisition problem under hand. The extension of the analysis to channels affected by code doppler (frequency uncertainties) is currently investigated.

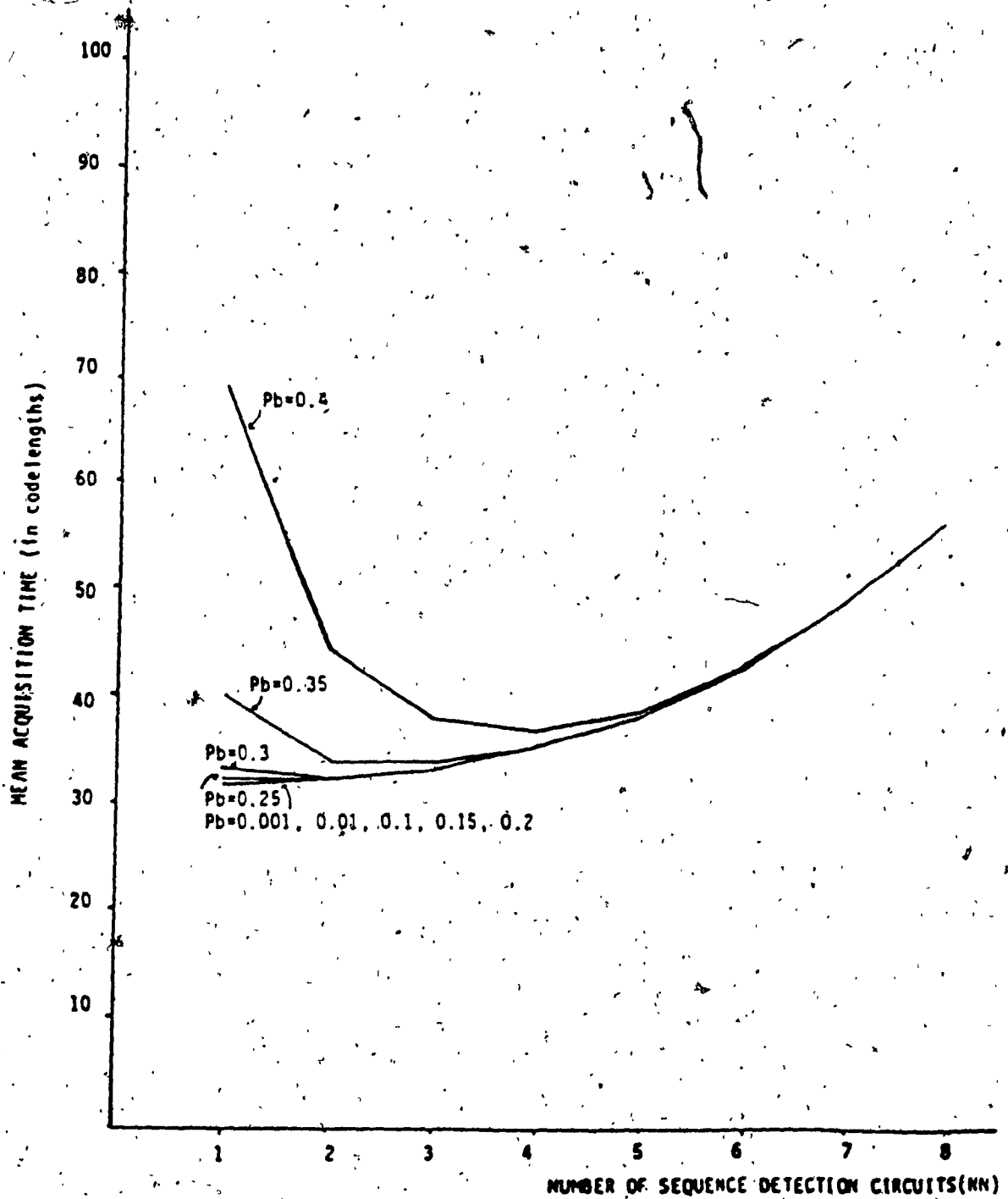


FIG. 3.8 a) : MEAN ACQUISITION TIME VERSUS NUMBER OF SEQUENCE DETECTION CIRCUITS FOR VARYING PROBABILITY OF BIT ERROR P_b
($P_b = 0.5, M = 8, M = 32, \text{THRESHOLD (TH)} = 8M/16$)
CODELENGTH = 256 chips

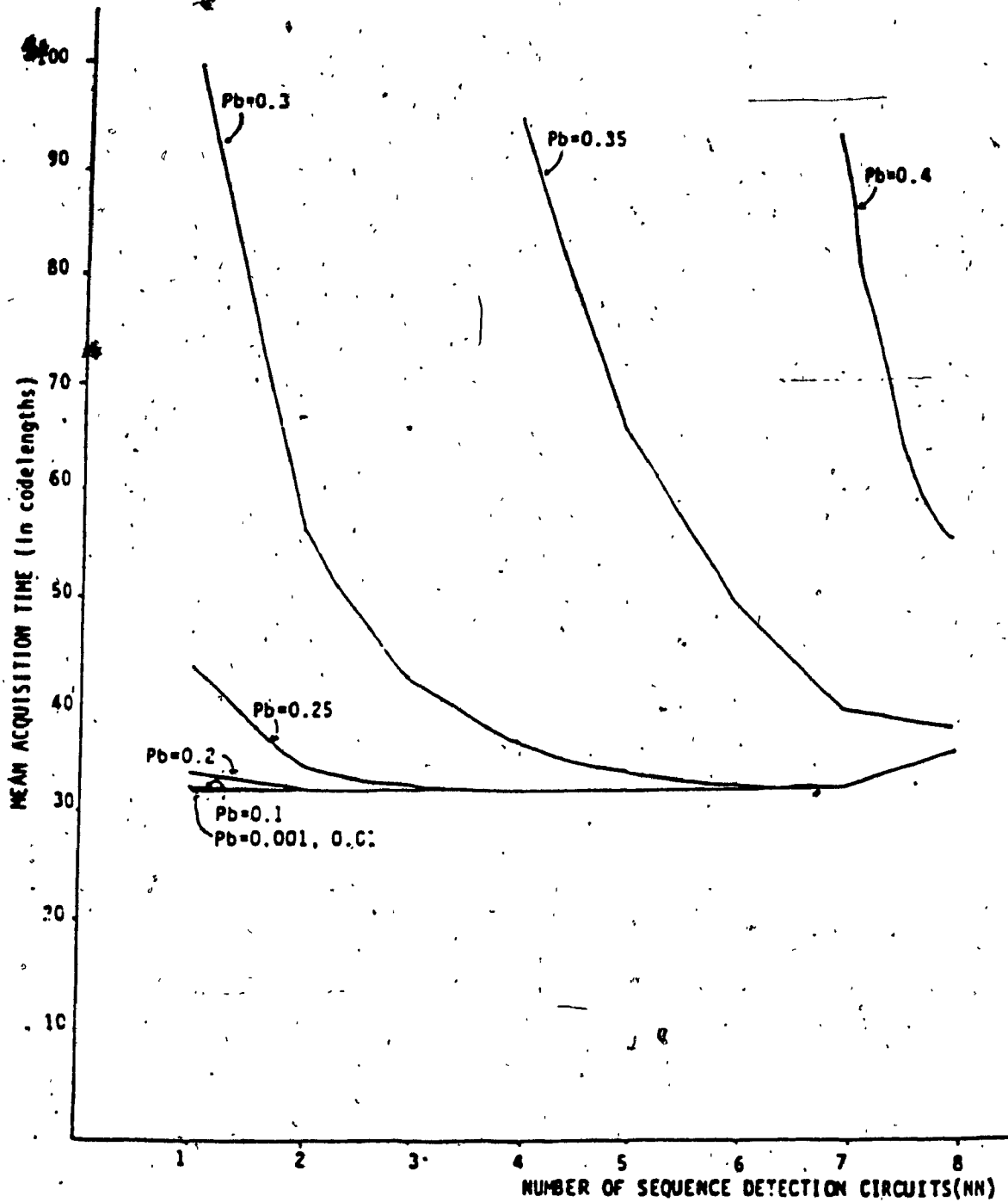


FIG. 3.8 b) : MEAN ACQUISITION TIME VERSUS NUMBER OF SEQUENCE DETECTION CIRCUITS FOR VARYING PROBABILITY OF BIT ERROR P_b ($P_b = 0.5$, $N = 8$, $M = 32$, THRESHOLD $(T_H) = 10M/16$) CODELENGTH = 256 chips

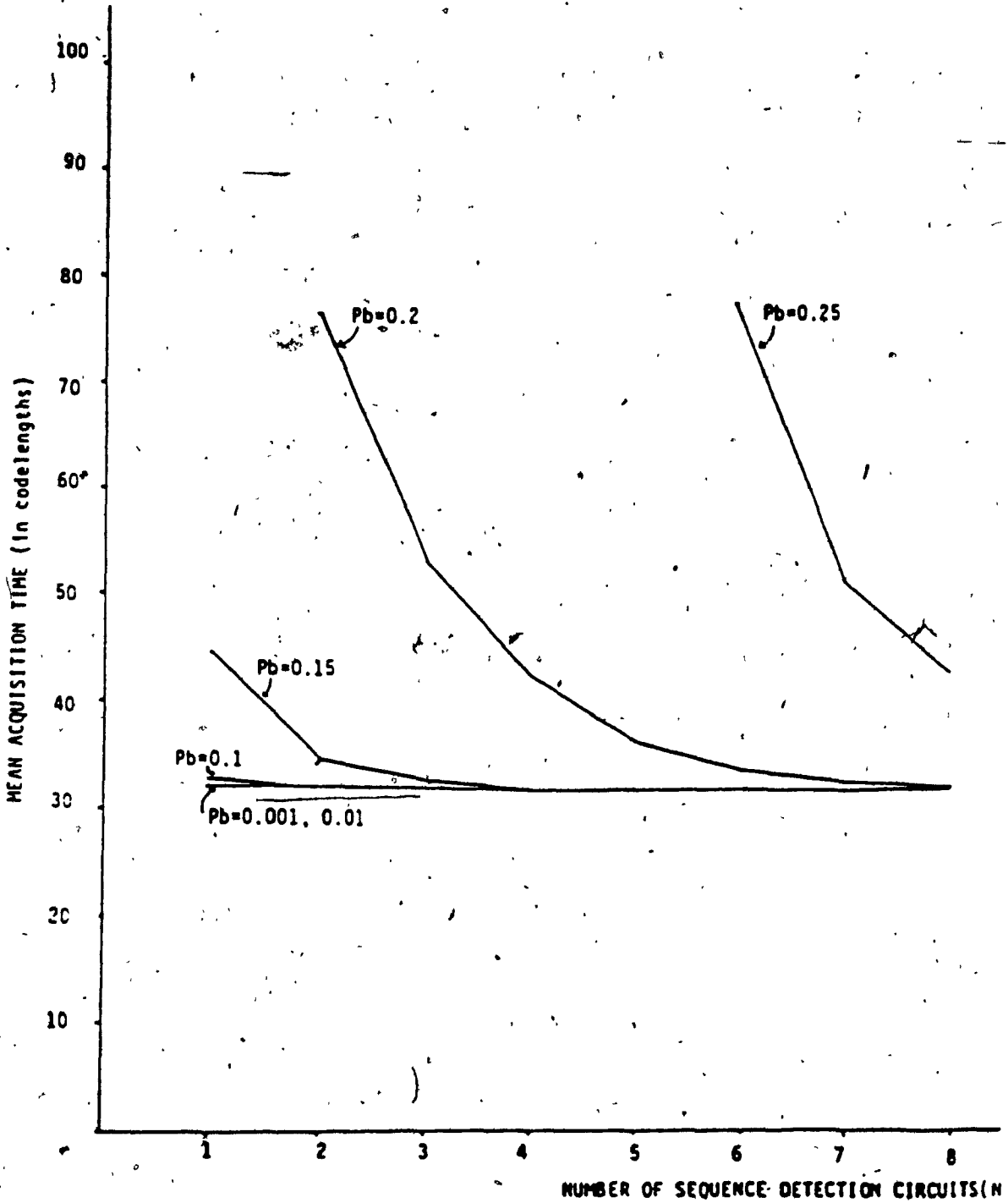


FIG. 3.8 c) : MEAN ACQUISITION TIME VERSUS NUMBER OF SEQUENCE DETECTION CIRCUITS FOR VARYING PROBABILITY OF BIT ERROR P_b ($P_b = 0.5$, $N = 8$, $M = 32$, THRESHOLD $(T_H) = 12M/16$) CODELENGTH = 256 chips

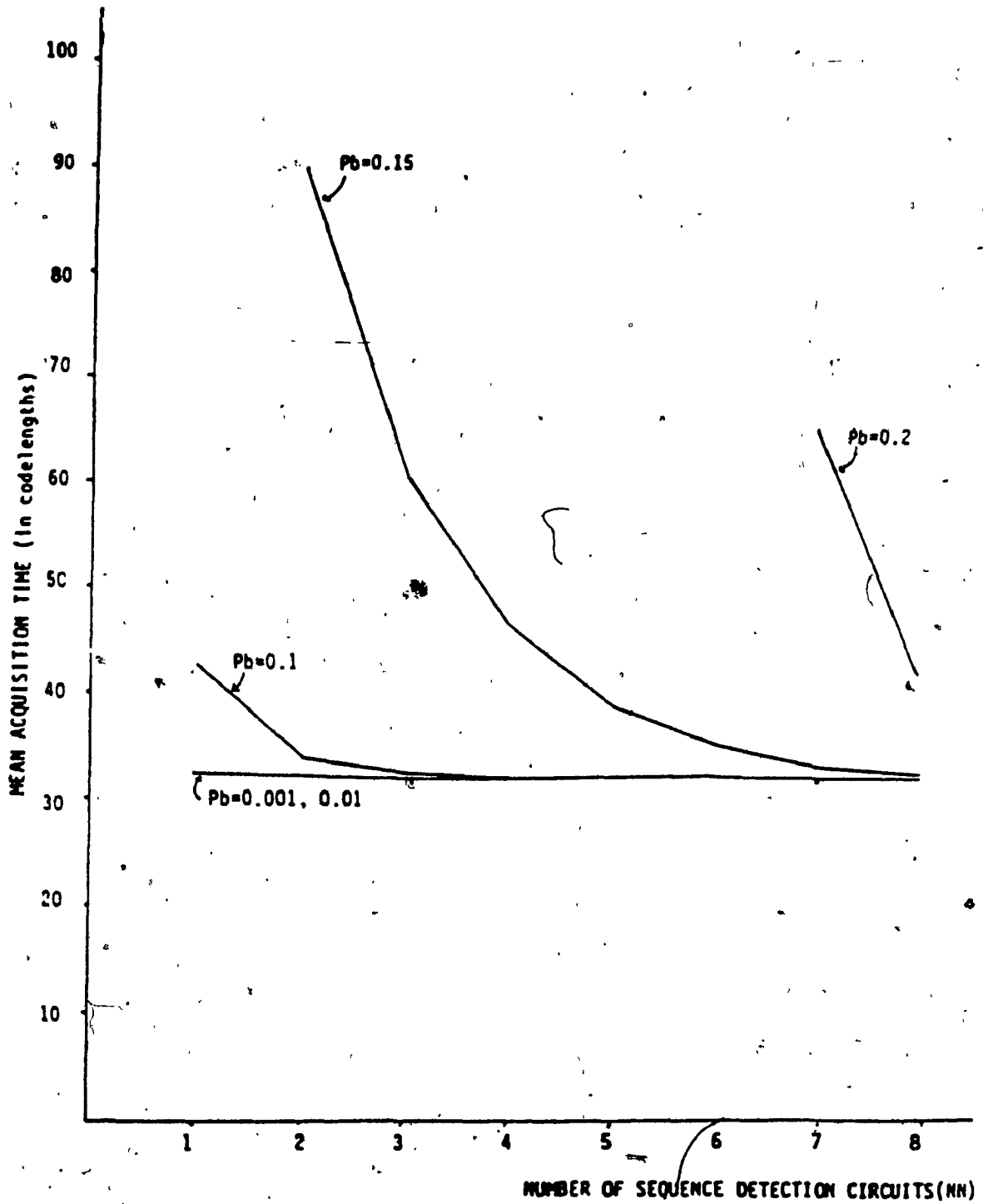


FIG. 3.8 d) : MEAN ACQUISITION TIME VERSUS NUMBER OF SEQUENCE DETECTION CIRCUITS FOR VARYING PROBABILITY OF BIT ERROR P_b
($P_b = 0.5$; $N = 8$, $M = 32$, THRESHOLD (T_H) = $13M/16$)
CODELENGTH = 256 chips

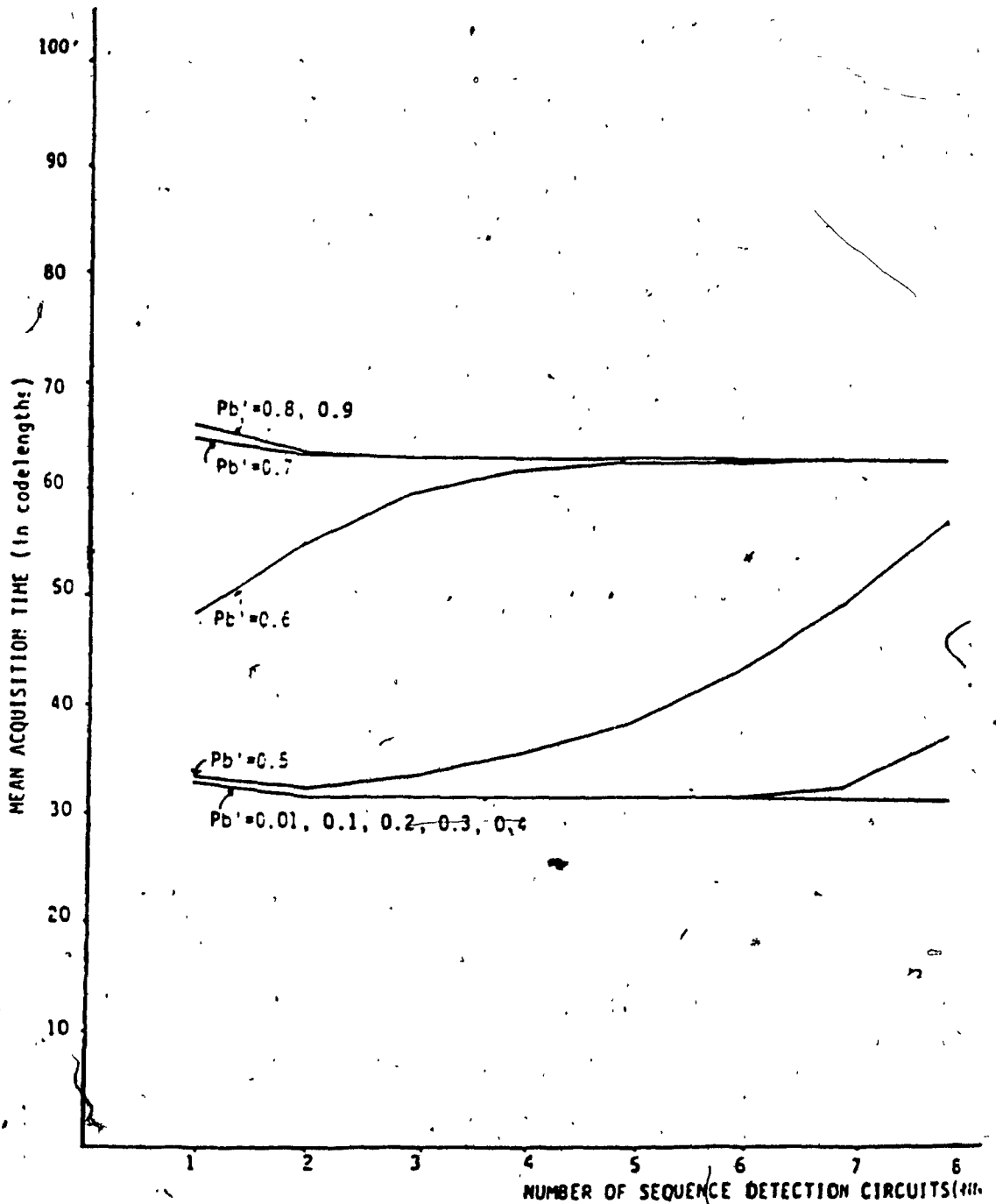


FIG. 3.9 a) : MEAN ACQUISITION TIME VERSUS NUMBER OF SEQUENCE DETECTION CIRCUITS FOR VARYING PROBABILITY OF BIT ERROR P_b' ($P_b = 0.3$, $N = 8$, $M = 32$, THRESHOLD $(T_H) = 8M/16$) CODELENGTH = 256 chips

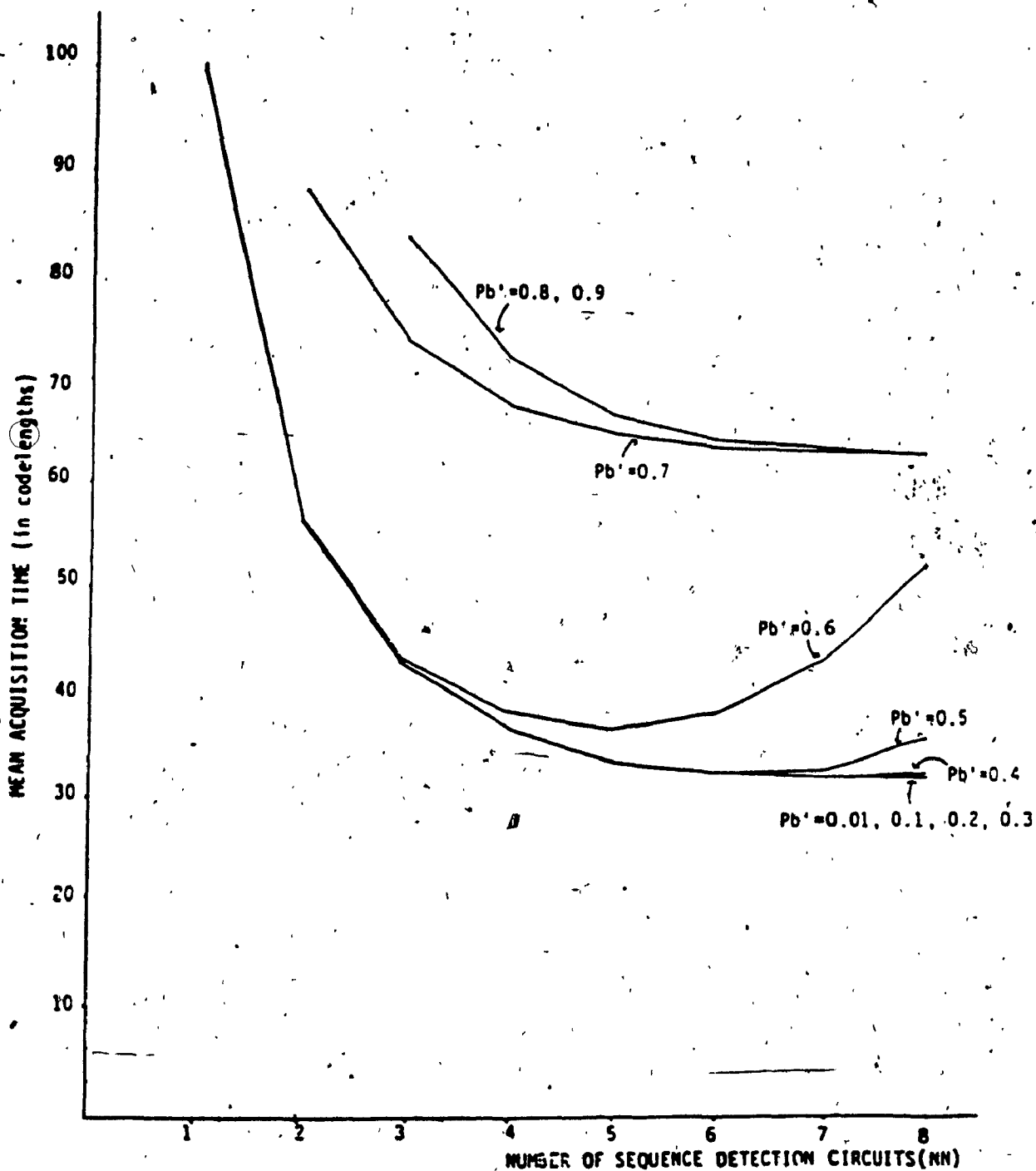


FIG. 3.9 b) : MEAN ACQUISITION TIME VERSUS NUMBER OF SEQUENCE DETECTION CIRCUITS FOR VARYING PROBABILITY OF BIT ERROR P_b' ($P_b = 0.3$, $N = 8$, $M = 32$, THRESHOLD (T_H) = $10M/16$), CODELENGTH = 256 chips

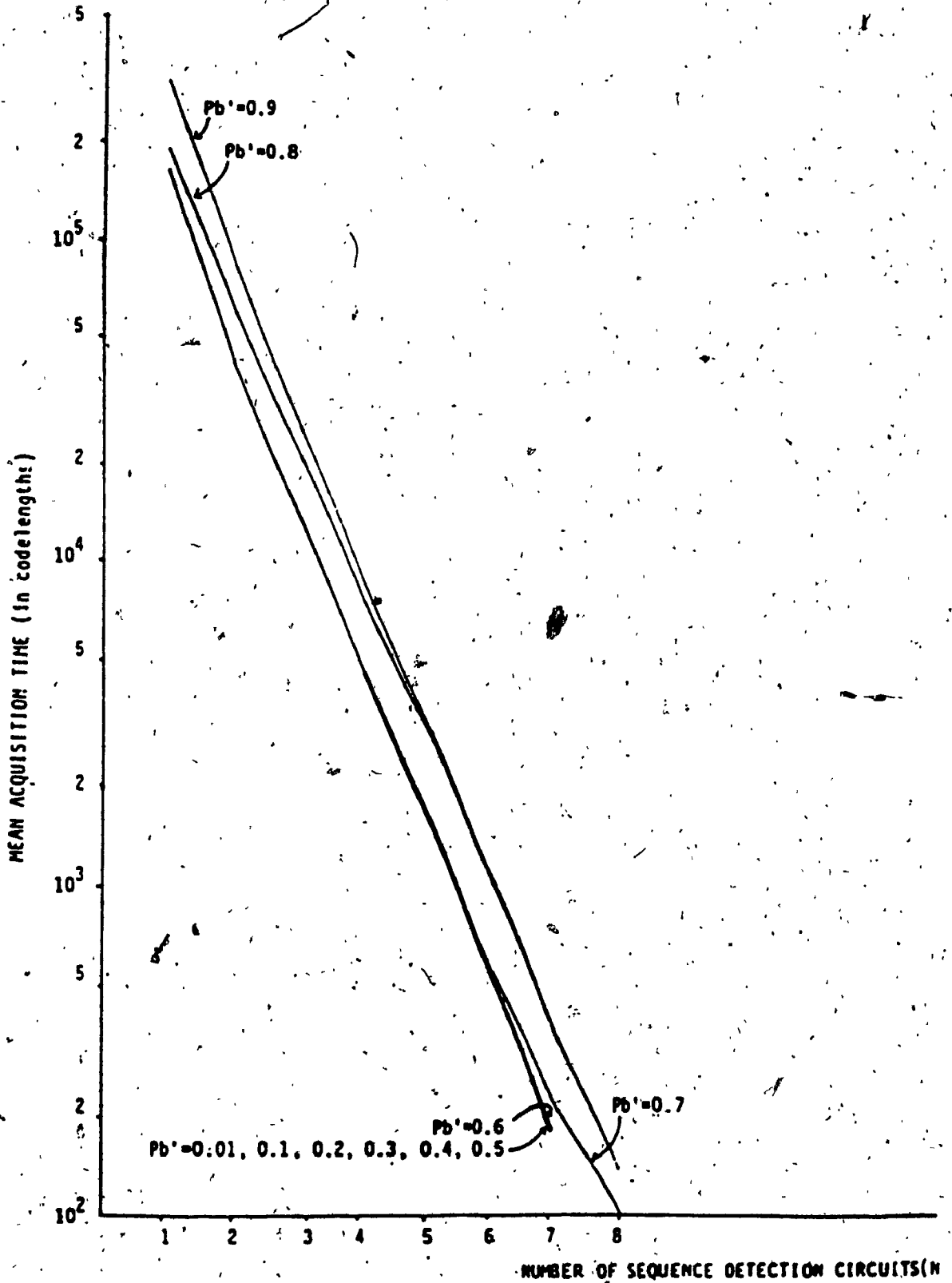


FIG. 3.9 c) : MEAN ACQUISITION TIME VERSUS NUMBER OF SEQUENCE DETECTION CIRCUITS FOR VARYING PROBABILITY OF BIT ERROR P_b' ($P_b = 0.3, N = 8, M = 32, \text{THRESHOLD (TH)} = 12M/15$)

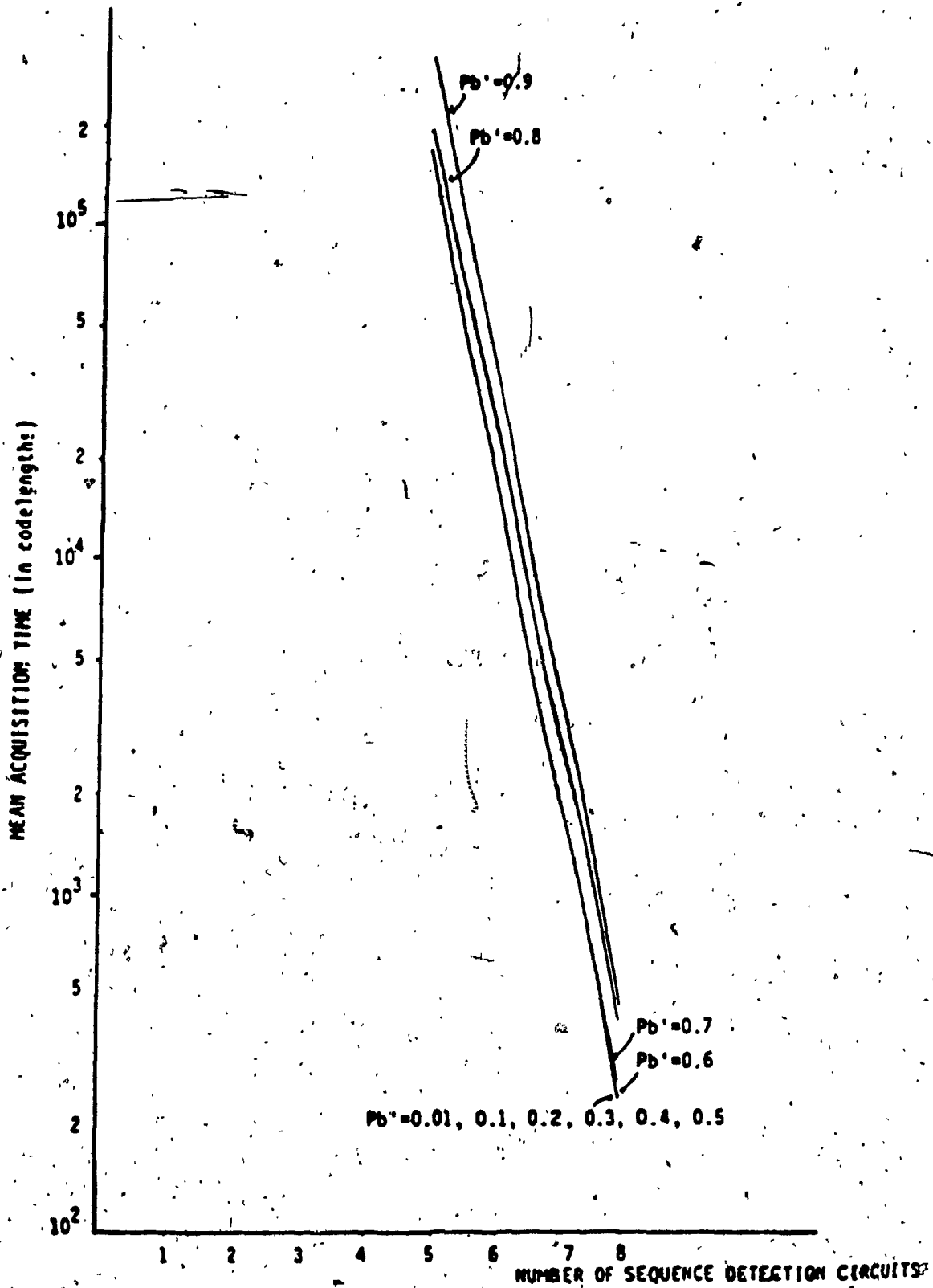


FIG. 3.9 d) : MEAN ACQUISITION TIME VERSUS NUMBER OF SEQUENCE DETECTION CIRCUIT FOR VARYING PROBABILITY OF BIT ERROR P_b' ($P_b = 0.3$, $N = 8$, $M = 32$, THRESHOLD $(T_H) = 13M/16$) CODELENGTH = 256 chips

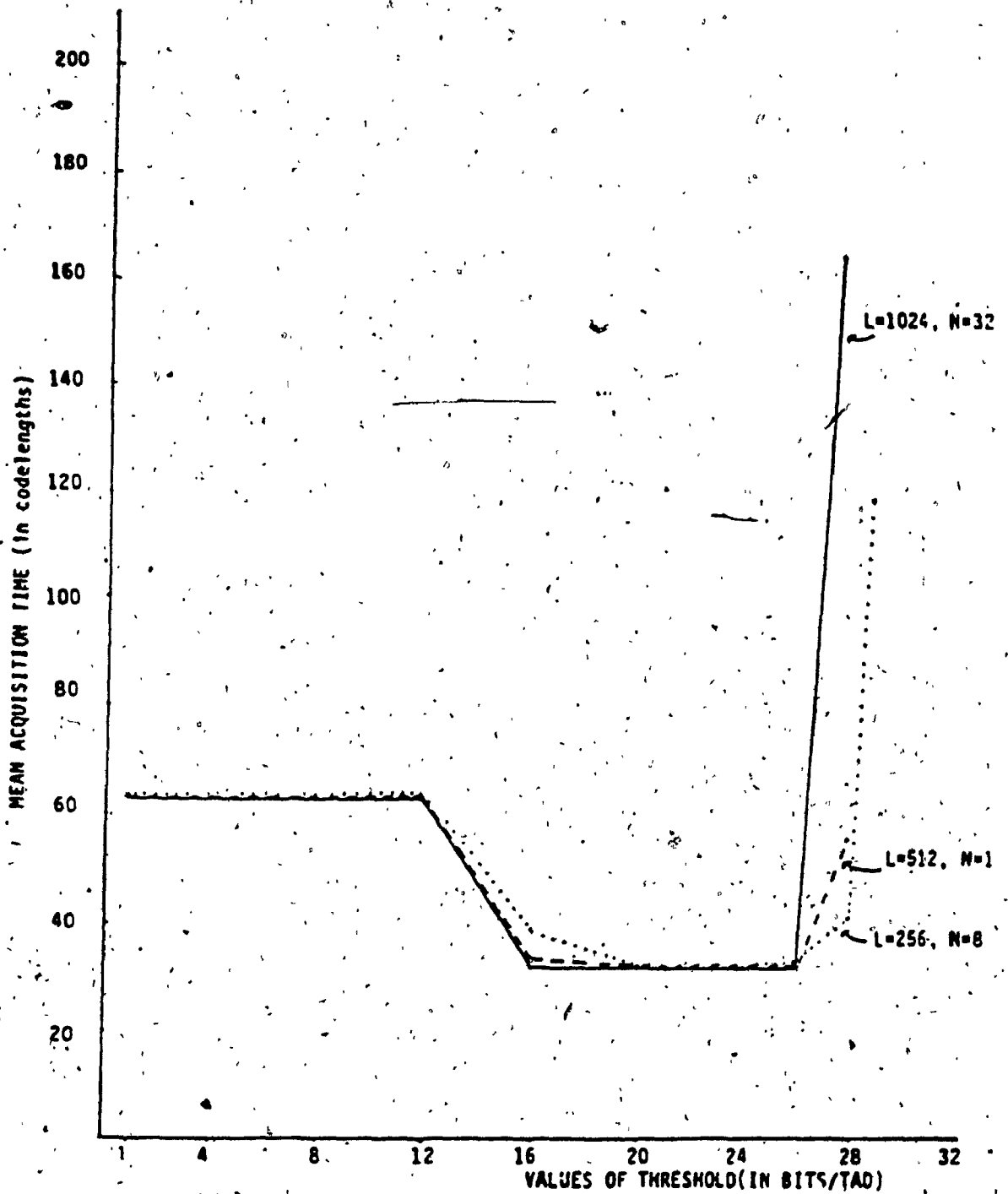


FIG. 3.10 : MEAN ACQUISITION TIME VERSUS THRESHOLD FOR VARYING VALUES OF CODELENGTH AND NUMBER OF TADS (N)
($P_b = 0.1$, $P_b' = 0.5$, $M = 32$, $NM = 5N/8$)

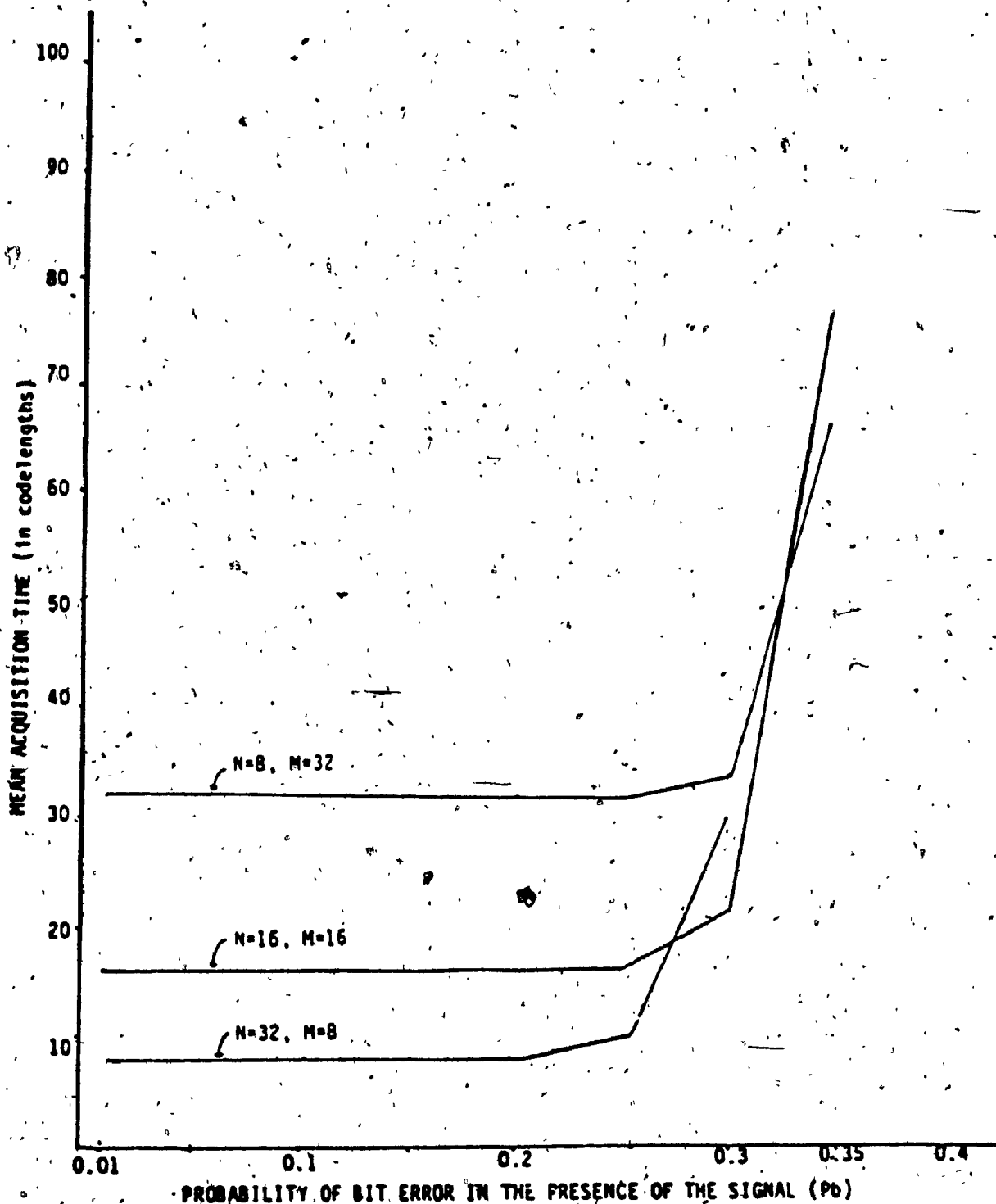


FIG. 3.11 : MEAN ACQUISITION TIME VERSUS PROBABILITY OF BIT ERROR, P_b , FOR VARYING NUMBER OF TADS N AND NUMBER OF BITS PER TAD M ($P_b = 0.5$, $NM = 5N/8$, $TH = 10M/16$)
CODELENGTH = 256 chips

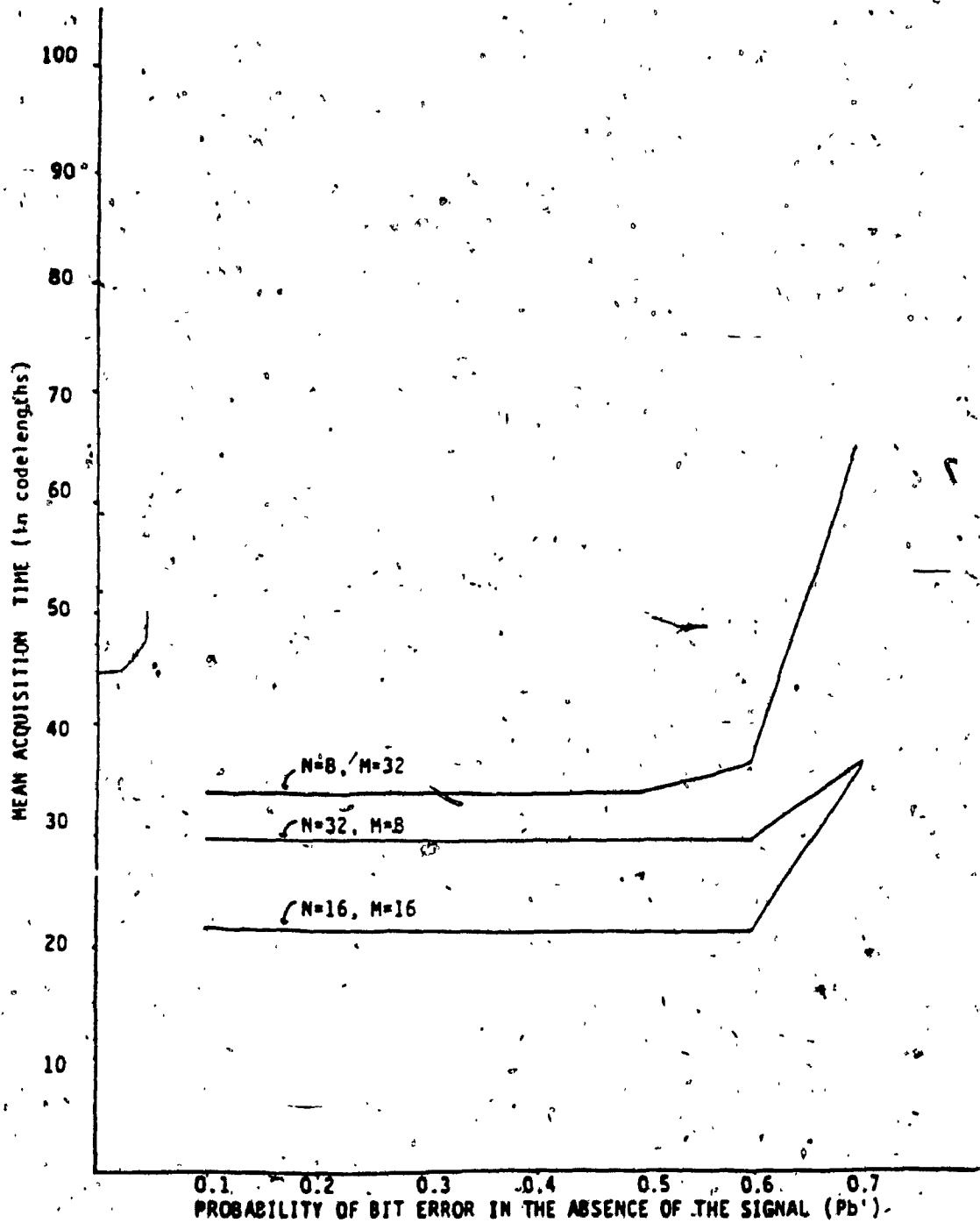


FIG. 3.12 : MEAN ACQUISITION TIME VERSUS PROBABILITY OF BIT ERROR P_b FOR VARYING NUMBER OF TADS N AND NUMBER OF BITS PER TAD M
($P_b = 0.3, \eta = 5N/8, T_H = 10M/16$)
CODELENGTH = 256 chips

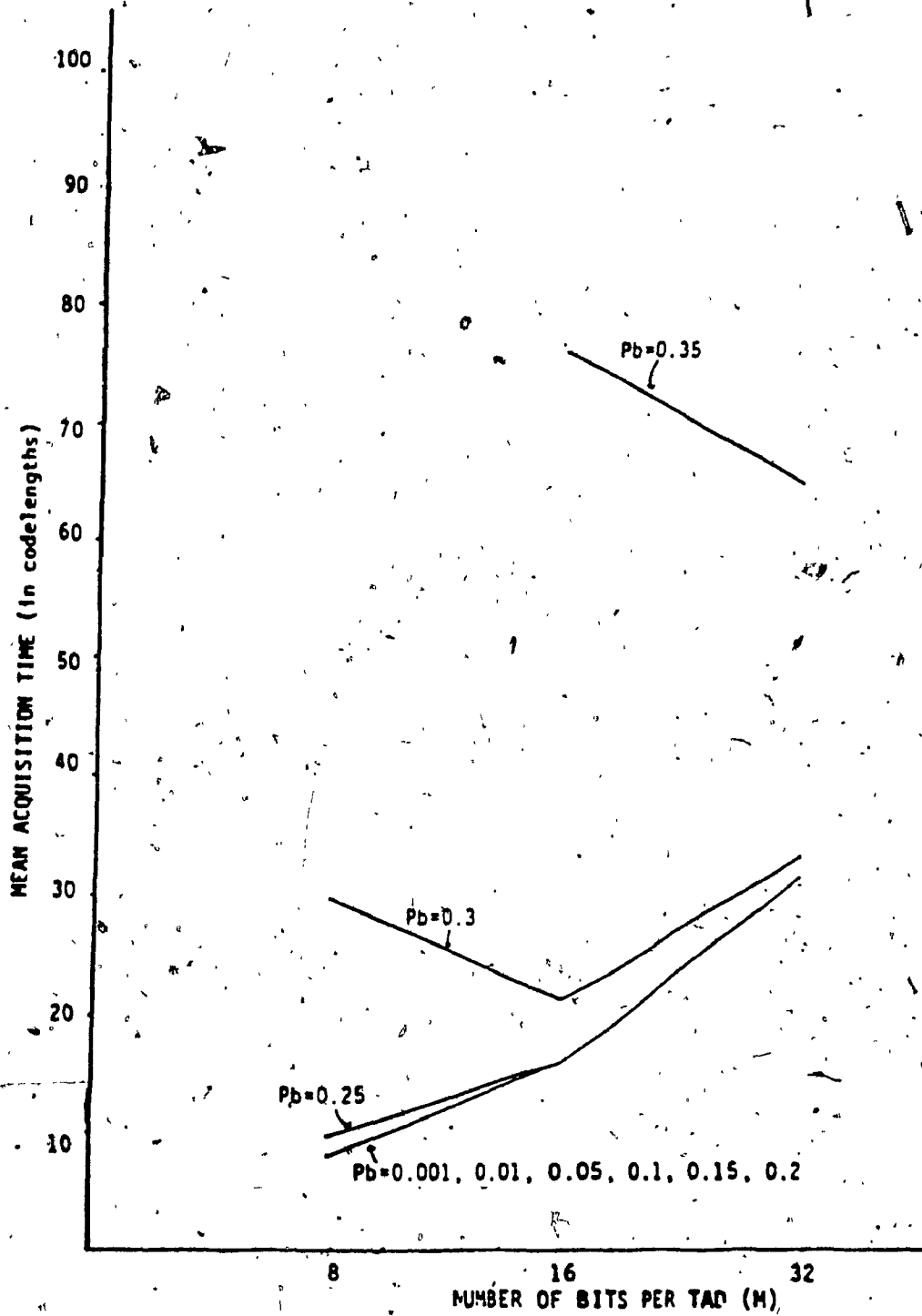


FIG. 3.13 a) : MEAN ACQUISITION TIME VERSUS NUMBER OF BITS PER TAD M FOR VARYING PROBABILITY OF BIT ERROR Pb (Pb = 0.5, CN = 5N/8, TH = 10M/16) CODELENGTH = 256 chips

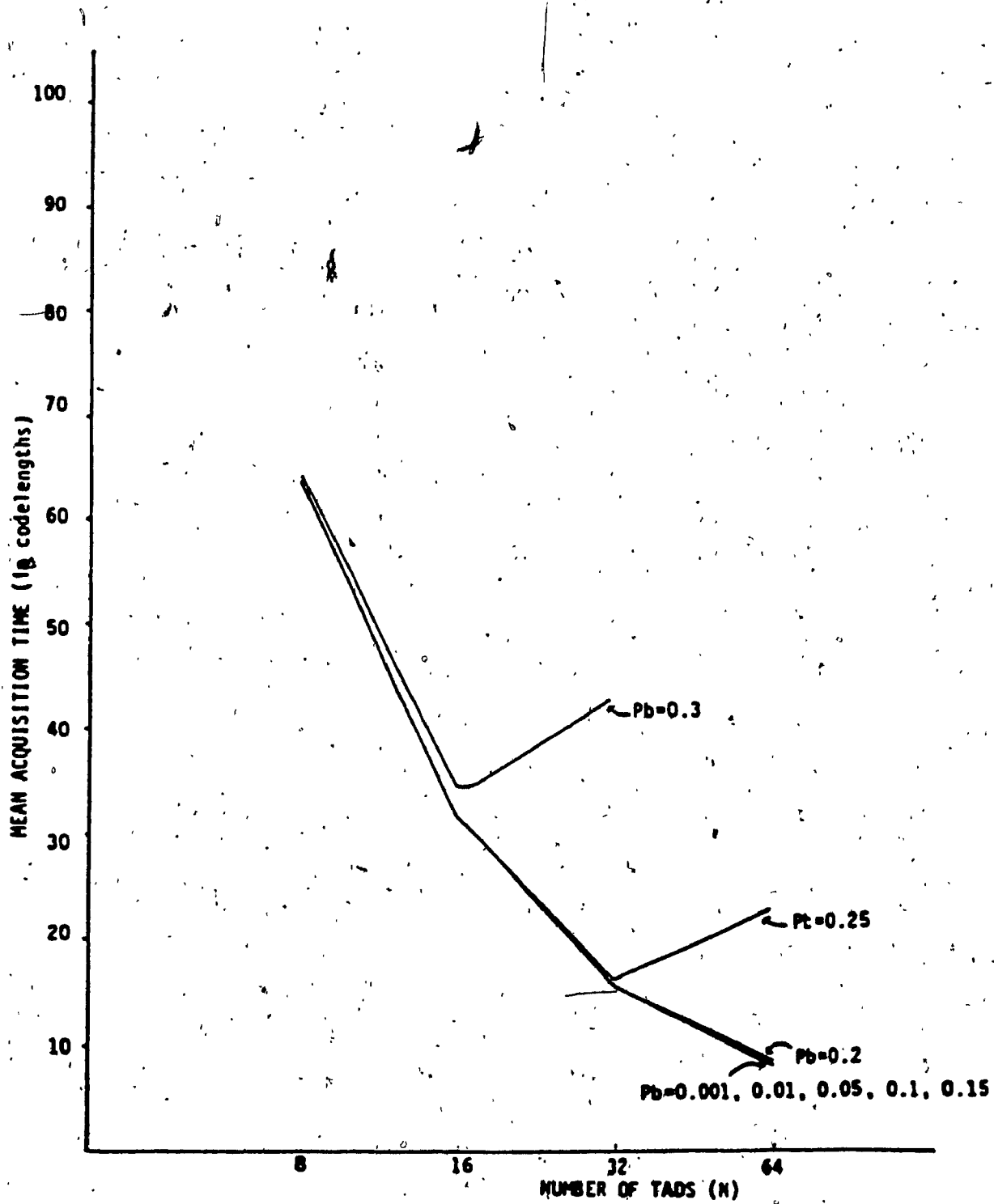


FIG. 3.13 b) : MEAN ACQUISITION TIME VERSUS NUMBER OF TADS N FOR VARYING PROBABILITY OF BIT ERROR P_b
($P_b' = 0.5$, $N_N = 5N/8$, $T_H = 10M/16$)
CODELENGTH = 512 chips

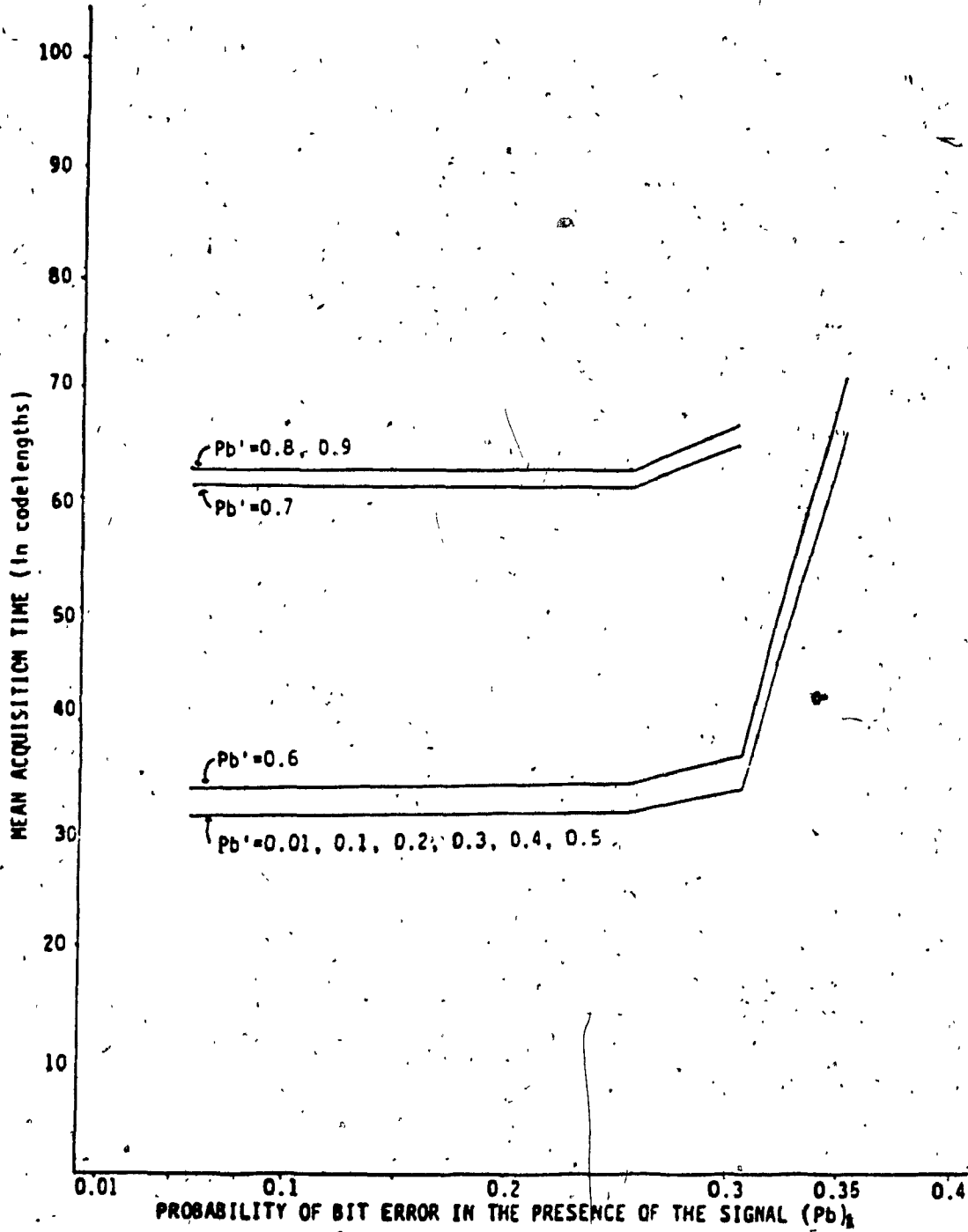


FIG. 3.14 : MEAN ACQUISITION TIME VERSUS P_b FOR VARYING VALUES OF P_b'
($N = 8, M = 32, MN = 5N/8, TH = 10M/16$)
CODELENGTH = 256 chips

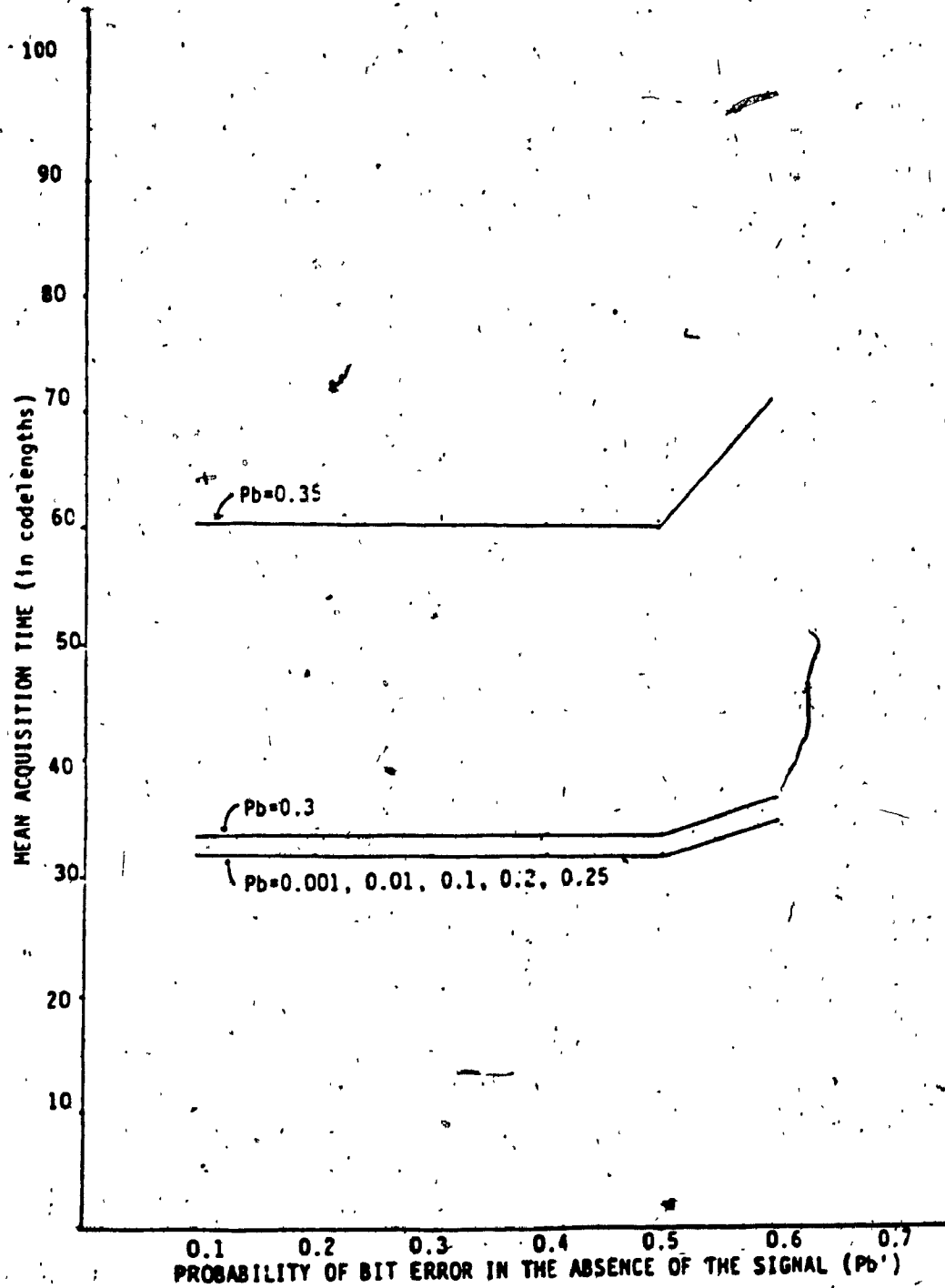


FIG. 3.15 : MEAN ACQUISITION TIME VERSUS P_b' FOR VARYING VALUES OF P_b
($N = 8, M = 32, N_N = 5N/8, T_H = 10M/16$)
CODELENGTH = 256 chips

TABLE 3.1A : BEST TH, BEST MM, AND CORRESPONDING OPTIMUM MEAN AND STANDARD DEVIATION UNDER VARIOUS BIT ERROR PROBABILITIES P_b AND P_b'

$P_b' \backslash P_b$	0.1	0.2	0.3	0.4	0.5	0.6
0.1	187 1 65536.00 65407.87	187 1 65536.00 65407.87	187 1 65536.00 65407.87	187 1 65536.00 65407.87	188 1 65536.00 65407.87	199 1 65536.00 65407.87
0.2	151 1 65536.00 65407.87	151 1 65536.00 65407.87	151 1 65536.00 65407.87	157 1 65536.00 65407.87	170 1 65536.01 65407.88	181 1 65561.00 65432.94
0.3	119 1 65536.00 65407.87	119 1 65536.00 65407.87	128 1 65536.00 65407.87	142 1 65536.05 65407.92	155 1 65597.75 65469.74	167 1 71833.70 71717.53
0.35	106 1 65536.00 65407.87	107 1 65536.00 65407.87	122 1 65536.00 65407.88	135 1 65539.03 65410.91	148 1 66490.85 66564.58	159 1 97637.77 97563.50
0.4	91 1 65536.00 65407.87	100 1 65536.00 65407.87	115 1 65536.05 65407.92	128 1 65617.10 65489.16	141 1 72884.55 72770.67	91 1 130816.00 130815.75

$M = 1$ $M = 256$ $L = 256$ chips
 $P_{align} = M/L = 0.00391$

BEST TH (units of bits/TAD)
 BEST MM
 MEAN (units of bits)
 STD DEVIATION (units of bits)

P_b : probability of bit error in the presence of the signal
 P_b' : probability of bit error in the absence of the signal
 N : number of TADs
 MM : number of SOCs
 M : number of bits per TAD
 TH : threshold
 L : codelength

TABLE 3.1B : BEST TH, BEST MN, AND CORRESPONDING OPTIMUM MEAN AND STANDARD DEVIATION UNDER VARIOUS BIT ERROR PROBABILITIES P_b AND P_{b'}

P _b \ P _{b'}	0.1	0.2	0.3	0.4	0.5	0.6
0.1	94 2 32768.00 32639.75	94 2 32768.00 32639.75	94 2 32768.00 32639.75	94 2 32768.00 32639.75	89 1 32768.00 32639.75	95 1 32768.01 32639.76
0.2	76 2 32768.00 32639.75	76 2 32768.00 32639.75	78 2 32768.00 32639.75	73 1 32768.00 32639.75	81 1 32768.33 32640.08	87 1 32846.24 32718.38
0.3	61 2 32768.00 32639.75	63 2 32768.00 32639.75	69 2 32768.00 32639.75	75 2 32768.88 32640.63	80 2 32913.97 32786.47	85 2 37426.76 37319.04
0.35	54 2 32768.00 32639.75	59 2 32768.00 32639.75	65 2 32768.03 32639.78	71 2 32784.15 32655.98	76 2 33854.53 35731.59	80 2 50533.27 50475.69
0.4	34 1 32768.00 32639.75	55 2 32768.00 32639.75	61 2 32768.87 32640.63	67 2 32938.22 32810.77	72 2 38017.78 37812.76	30 1 65280.00 65279.50

M = 2 M = 128 L = 256 chips

P_{align} = M/L = 0.00781

P_b: probability of bit error in the presence of the signal

P_{b'}: probability of bit error in the absence of the signal

M : number of TADs

MN: number of SDCs

M : number of bits per TAD

TH: threshold

L : code length

BEST TH (units of bits/TAD)
BEST MN
MEAN (units of bits)
STD DEVIATION (units of bits)

TABLE 3.10 : BEST TH, BEST NH, AND CORRESPONDING OPTIMUM MEAN AND STANDARD DEVIATION UNDER VARIOUS BIT ERROR PROBABILITIES P_b AND P_b' .

P_b \ P_b'	0.1	0.2	0.3	0.4	0.5	0.6
0.1	47 4 16384.00 16255.50	47 4 16384.00 16255.50	45 3 16384.00 16255.50	42 2 16384.00 16255.50	45 2 16384.00 16255.50	48 2 16384.00 16255.51
0.2	39 4 16384.00 16255.50	36 3 16384.00 16255.50	34 2 16384.00 16255.50	38 2 16384.00 16255.50	41 2 16384.23 16255.73	44 2 16422.30 16294.03
0.3	28 3 16384.00 16255.50	30 3 16384.00 16255.50	30 2 16384.00 16255.50	34 2 16384.62 16256.12	37 2 16459.94 16332.07	40 2 18807.96 18698.95
0.35	25 3 16384.00 16255.50	28 3 16384.00 16255.50	29 2 16384.04 16255.53	32 2 16393.63 16265.21	37 3 16936.75 16813.10	39 3 25250.44 25191.19
0.4	22 3 16384.00 16255.50	26 3 16384.00 16255.50	27 2 16384.64 16256.14	32 3 16484.72 16357.44	35 3 19024.68 18917.83	9 1 32512.00 32511.00

$N = 4$ $M = 64$ $L = 256$ chips
 $P_{align} = M/L = 0.01563$

BEST TH (units of bits/TAD)
 BEST NH
 MEAN (units of bits)
 STD DEVIATION (units of bits)

P_b : probability of bit error in the presence of the signal
 P_b' : probability of bit error in the absence of the signal
 N : number of TADs
 M : number of SDGs
 M : number of bits per TAD
 TH : threshold
 L : codeword length

TABLE 3.13 - BEST TH, BEST NM, AND CORRESPONDING OPTIMUM MEAN AND STANDARD DEVIATION UNDER VARIOUS BIT ERROR PROBABILITIES P_b AND P_b'

$P_b \backslash P_b'$	0.1	0.2	0.3	0.4	0.5	0.6
0.1	22 8 8192.00 8062.98	22 7 8192.00 8062.98	22 6 8192.00 8062.98	21 4 8192.00 8062.98	23 4 8192.00 8062.98	24 4 8192.00 8062.98
0.2	15 6 8192.00 8062.98	15 4 8192.00 8062.98	17 4 8192.00 8062.98	19 4 8192.00 8062.98	21 5 8192.14 8063.12	22 5 8213.35 8064.89
0.3	12 5 8192.00 8062.98	13 4 8192.00 8062.98	15 4 8192.00 8062.98	17 4 8192.49 8063.48	18 4 8229.57 8101.39	20 5 8939.04 9228.94
0.35	10 4 8192.00 8061.98	12 4 8192.00 8061.98	14 4 8192.02 8063.00	16 4 8198.52 8069.54	17 4 8473.19 8350.29	19 5 12688.35 12614.21
0.4	9 4 8192.00 8062.98	12 5 8192.00 8062.98	14 5 8192.37 8063.36	15 4 8245.10 8116.56	16 4 9569.54 9467.66	10 5 16128.00 16125.97

$M = 8$ $M = 32$ $L = 256$ chips

$P_{align} = M/L = 0.03125$

P_b : probability of bit error in the presence of the signal

P_b' : probability of bit error in the absence of the signal

N : number of TADs

NM : number of SDCs

M : number of bits per TAD

TH : threshold

L : code length

BEST TH (units of bits/TAD)
BEST NM
MEAN (units of bits)
STD DEVIATION (units of bits)

TABLE 3.12 : BEST TH, BEST MH, CORRESPONDING OPTIMUM MEAN AND STANDARD DEVIATION UNDER VARIOUS BIT ERROR PROBABILITIES P_b AND P_d .

$P_b \backslash P_d$	0.1	0.2	0.3	0.4	0.5	0.6
0.1	10 14 4096.00 3965.93	10 13 4096.00 3965.93	10 10 4096.00 3965.93	10 6 4096.00 3965.93	11 7 4096.00 3965.93	12 9 4096.00 3965.93
0.2	5 8 4096.00 3965.93	8 10 4096.00 3965.93	8 7 4096.00 3965.93	9 8 4096.00 3965.94	10 9 4096.07 3966.00	10 6 4106.13 3976.52
0.3	6 11 4096.00 3965.93	6 7 4096.00 3965.93	7 7 4096.00 3965.94	8 8 4096.19 3966.13	9 10 4121.95 3992.60	9 7 4688.00 4582.31
0.35	5 9 4096.00 3965.93	6 8 4096.00 3965.93	7 9 4096.01 3965.95	7 6 4099.55 3969.41	8 8 4235.01 4110.78	9 11 6309.40 6248.74
0.4	5 10 4096.00 3965.93	5 6 4096.00 3965.94	6 7 4096.22 3966.16	7 9 4122.49 3993.61	8 11 4858.51 4753.10	5 16 7936.00 7931.07

BEST TH (units of bits/TAD)
BEST MH
MEAN (units of bits)
STD DEVIATION (units of bits)

$M = 16$ $N = 16$ $L = 256$ chips

$P_{align} = M/L = 0.06250$

P_b : probability of bit error in the presence of the signal

P_d : probability of bit error in the absence of the signal

M : number of TADs

MH : number of SDCs

H : number of bits per TAD

TH : threshold

TABLE 3.1P: BEST TH, BEST NH, CORRESPONDING OPTIMUM MEAN AND STANDARD DEVIATION UNDER VARIOUS BIT ERROR PROBABILITIES P_b AND P_d .

$P_b \backslash P_d$	0.1	0.2	0.3	0.4	0.5	0.6
0.1	2 13 2048.00 1915.73	3 12 2048.00 1915.73	5 21 2048.00 1915.73	5 15 2048.00 1915.73	5 11 2048.00 1915.73	5 7 2048.00 1915.74
0.2	2 13 2048.00 1915.73	4 21 2048.00 1915.73	4 17 2048.00 1915.73	4 13 2048.00 1915.73	4 8 2048.10 1915.84	5 19 2054.72 1922.87
0.3	3 22 2048.00 1915.73	3 17 2048.00 1915.73	3 12 2048.00 1915.73	3 8 2048.48 1916.27	4 19 2061.77 1930.71	4 13 2340.77 2232.84
0.35	2 14 2048.00 1915.73	3 21 2048.00 1915.73	3 17 2048.01 1915.74	3 11 2050.26 1918.18	4 24 2190.90 2069.55	4 20 3111.47 3047.09
0.4	2 16 2048.00 1915.73	2 10 2048.00 1915.73	3 21 2048.46 1916.22	3 16 2064.29 1933.25	3 10 2440.42 2338.43	1 32 3640.00 3831.46

$N = 32$ $M = 8$ $L = 256$ chips

$P_{align} = N/L = 0.12500$

P_b : probability of bit error in the presence of the signal

P_d : probability of bit error in the absence of the signal

N : number of TADs

M : number of SOCS

H : number of bits per TAD

TH : threshold

BEST TH (units of bits/TAD)
BEST NH
MEAN (units of bits)
STD DEVIATION (units of bits)

TABLE 3.10 : BEST TH, BEST MN, CORRESPONDING OPTIMUM MEAN AND STANDARD DEVIATION UNDER VARIOUS BIT ERROR PROBABILITIES P_b AND P_b'

$P_b' \backslash P_b$	0.1	0.2	0.3	0.4	0.5	0.6
0.1	1 33 1024.00 886.81	2 45 1024.00 886.81	2 33 1024.00 886.81	2 16 1024.00 886.81	2 12 1024.00 886.81	2 8 1024.04 886.87
0.2	1 33 1024.00 886.81	1 12 1024.00 886.81	2 46 1024.00 886.81	2 40 1024.00 886.81	2 34 1024.03 886.85	2 25 1026.50 889.79
0.3	1 33 1024.00 886.81	1 22 1024.00 886.81	1 14 1024.00 886.81	1 9 1025.97 889.31	2 52 1056.94 927.34	2 47 1236.88 1123.90
0.35	1 36 1024.00 886.81	1 29 1024.00 886.81	1 21 1024.02 886.84	1 13 1028.54 892.44	1 7 1156.92 1046.36	2 55 1628.50 1574.70
0.4	1 45 1024.00 886.81	1 37 1024.00 886.81	1 29 1024.23 887.09	1 20 1038.27 904.27	1 12 1279.15 1188.11	1 64 1792.00 1773.62

$N = 64$ $M = 4$ $L = 256$ chips

$P_{align} = M/L = 0.25000$

P_b : probability of bit error in the presence of the signal

P_b' : probability of bit error in the absence of the signal

M : number of TADs

MN : number of SDCs

M : number of bits per TAD

TH: threshold

BEST TH (units of bits/TAD)
BEST MN
MEAN (units of bits)
STD DEVIATION (units of bits)

TABLE 3.1B : BEST TH, BEST MH, CORRESPONDING OPTIMUM MEAN AND STANDARD DEVIATION UNDER VARIOUS BIT ERROR PROBABILITIES P_b AND P_b'

$P_b' \backslash P_b$	0.1	0.2	0.3	0.4	0.5	0.6
0.1	1 109 512.00 362.04	1 96 512.00 362.04	1 79 512.00 362.04	1 62 512.00 362.04	1 51 512.00 362.04	1 38 512.00 362.04
0.2	1 117 512.00 362.04	1 114 512.00 362.04	1 110 512.00 362.04	1 105 512.00 362.08	1 99 512.38 362.57	1 92 516.18 368.22
0.3	1 124 512.17 362.28	1 122 512.89 363.20	1 120 515.47 366.58	1 118 523.77 378.08	1 116 551.15 418.31	1 113 641.68 541.49
0.35	1 125 513.75 364.18	1 124 517.73 369.79	1 123 528.13 385.27	1 121 554.10 418.98	1 120 618.75 511.31	1 121 750.46 694.24
0.4	1 126 519.49 371.49	1 125 532.07 387.81	1 124 557.57 421.47	1 123 609.96 491.24	1 123 708.37 629.48	1 128 768.25 724.39

$N = 128$ $M = 2$ $L = 256$ chips
 $P_{align} = N/L = 0.50000$

P_b : probability of bit error in the presence of the signal
 P_b' : probability of bit error in the absence of the signal
 N : number of TADs
 M : number of SDCs
 L : number of bits per TAD
 TH : threshold

BEST TH (units of bits/TAD)
 BEST MH
 MEAN (units of bits)
 STD DEVIATION (units of bits)

TABLE 3.11: BEST TH, BEST NM, CORRESPONDING OPTIMUM MEAN AND STANDARD DEVIATION UNDER VARIOUS BIT ERROR PROBABILITIES P_b AND P_0

$P_b \backslash P_0$	0.1	0.2	0.3	0.4	0.5	0.6
0.1	1 256 256.00 0.29	1 256 256.00 0.29	1 256 256.00 0.29	1 256 256.00 0.29	1 256 256.00 0.29	1 256 256.00 0.29
0.2	1 256 256.07 14.91	1 256 256.07 14.91	1 256 256.07 14.91	1 256 256.07 14.91	1 256 256.07 14.91	1 256 256.07 14.91
0.3	1 256 267.31 54.99	1 256 267.31 54.99	1 256 267.31 54.99	1 256 267.31 54.99	1 256 267.31 54.99	1 256 267.31 54.99
0.35	1 256 279.90 81.79	1 256 279.90 81.79	1 256 279.90 81.79	1 256 279.90 81.79	1 256 279.90 81.79	1 256 279.90 81.79
0.4	1 256 298.76 113.02	1 256 298.76 113.02	1 256 298.76 113.02	1 256 298.76 113.02	1 256 298.76 113.02	1 256 298.76 113.02

BEST TH (units of bits/TAD)
 BEST NM
 MEAN (units of bits)
 STD DEVIATION (units of bits)

$M = 256$ $N = 1$ $L = 256$ chips

$P_{align} = N/L = 1.00000$

P_b : probability of bit error in the presence of the signal

P_0 : probability of bit error in the absence of the signal

N : number of TADs

NM : number of SDCs

M : number of bits per TAD

TH: threshold

L : code length

TABLE 3.2A : BEST TH, BEST MN, CORRESPONDING OPTIMUM MEAN AND STANDARD DEVIATION UNDER VARIOUS BIT ERROR PROBABILITIES P_b AND P_b'

$P_b \backslash P_b'$	0.1	0.2	0.3	0.4	0.5	0.6
0.1	82 1 16384.00 16319.87	82 1 16384.00 16319.87	82 1 16384.00 16319.87	89 1 16384.00 16319.87	94 1 16384.00 16319.88	100 1 16384.13 16320.40
0.2	62 1 16384.00 16319.87	65 1 16384.00 16319.87	72 1 16384.00 16319.87	79 1 16384.02 16319.89	85 1 16387.78 16323.67	91 1 16377.50 16314.14
0.3	47 1 16384.00 16319.87	57 1 16384.00 16319.87	64 1 16384.04 16319.91	71 1 16391.70 16327.61	78 1 16705.00 16641.94	83 1 20646.89 20599.59
0.35	43 1 16384.00 16319.87	53 1 16384.00 16319.88	61 1 16384.66 16320.54	68 1 16448.51 16448.51	74 1 17805.41 17805.41	78 1 27778.79 27754.42
0.4	40 1 16384.00 16319.87	50 1 16384.02 16319.89	58 1 16391.72 16327.61	64 1 16754.46 16692.08	70 1 21064.95 21018.95	80 1 32640.00 32639.75

$M = 1$, $M = 128$ $L = 128$ chips
 $P_{align} = M/L = 0.00781$

P_b : probability of bit error in the presence of the signal
 P_b' : probability of bit error in the absence of the signal
 MN: number of TADs
 MN: number of SDCs
 M: number of bits. per TAD
 TH: threshold
 L: code length

BEST TH (units of bits/TAD)
 BEST MN
 MEAN (units of bits)
 STD DEVIATION (units of bits)

TABLE 3-2B : BEST TH, BEST NM, CORRESPONDING OPTIMUM MEAN AND STANDARD DEVIATION UNDER VARIOUS BIT ERROR PROBABILITIES P_b AND P_b'

P _b \ P _b '	0.1	0.2	0.3	0.4	0.5	0.6
0.1	42 2 0192.00 0127.75	42 2 0192.00 0127.75	37 1 0192.00 0127.75	41 1 0192.00 0127.75	44 1 0192.04 0127.79	48 1 0194.84 0130.60
0.2	32 2 0192.00 0127.75	36 2 0192.00 0127.75	33 1 0192.00 0127.75	37 1 0192.30 0128.05	40 1 0205.00 0140.86	47 2 0427.46 0305.45
0.3	27 2 0192.00 0127.75	32 2 0192.00 0127.75	35 2 0192.45 0128.20	38 2 0213.58 0149.50	40 2 0336.64 0476.20	43 2 10019.46 10773.35
0.35	25 2 0192.00 0127.75	30 2 0192.03 0127.77	33 2 0195.53 0131.31	36 2 0294.51 0231.05	38 2 9270.05 9216.40	40 2 14119.22 14096.64
0.4	24 2 0192.00 0127.75	28 2 0192.26 0128.01	31 2 0212.38 0148.31	34 2 0363.50 0502.16	36 2 10985.93 10944.67	39 1 16156.00 16233.30

M = 2 M = 64 L = 128 chips

P_{align} = M/L = 0.01563

P_b: probability of bit error in the presence of the signal
 P_b': probability of bit error in the absence of the signal
 M: number of TADS
 NM: number of SDCS
 M: number of bits per TAD
 TH: threshold
 L: code length

BEST TH (units of bits/TAD)
 BEST NM
 MEAN
 STANDARD DEVIATION (units of bits)

TABLE 3.2C : BEST TH, BEST NM, CORRESPONDING OPTIMUM MEAN AND STANDARD DEVIATION UNDER VARIOUS BIT ERROR PROBABILITIES P_b AND P_b'

P _b \ P _b '	0.1	0.2	0.3	0.4	0.5	0.6
0.1	20 3 4096.00 4031.49	20 3 4096.00 4031.49	19 2 4096.00 4031.49	21 2 4096.00 4031.49	23 2 4096.04 4031.53	24 2 4097.33 4032.84
0.2	15 3 4096.00 4031.49	15 2 4096.00 4031.49	17 2 4096.00 4031.49	19 2 4096.22 4031.72	20 2 4103.26 4038.93	23 3 4211.50 4148.07
0.3	13 3 4096.00 4031.49	15 3 4096.00 4031.49	15 2 4096.30 4031.79	18 3 4108.04 4043.81	18 2 4282.61 4222.30	21 3 5435.25 5386.75
0.35	12 3 4096.00 4031.49	14 3 4096.02 4031.51	14 2 4098.27 4033.81	17 3 4149.08 4093.74	18 3 4685.08 4695.08	19 3 7053.89 7035.43
0.4	11 3 4096.00 4031.49	13 3 4096.20 4031.70	15 3 4108.89 4044.52	16 3 4284.36 4223.81	17 3 5548.34 5511.18	8 4 8064.00 8062.98

N = 4 M = 32 L = 128 chips
P_{align} = M/L = 0.03125

BEST TH (units of bits/TAD)
BEST NM
MEAN (units of bits)
STD DEVIATION (units of bits)

P_b: probability of bit error in the presence of the signal
P_b': probability of bit error in the absence of the signal
N: number of TADs
NM: number of SDCs
M: number of bits per TAD
TH: threshold
L: codeword length

TABLE 3.2D : BEST TH, BEST MN, CORRESPONDING OPTIMUM MEAN AND STANDARD DEVIATION UNDER VARIOUS BIT ERROR PROBABILITIES P_b AND P_b'

P _b \ P _b '	0.1	0.2	0.3	0.4	0.5	0.6
0.1	10 6 2048.00 1982.97	9 4 2048.00 1982.97	10 4 2048.00 1982.97	10 3 2048.00 1982.97	12 5 2048.02 1982.99	12 4 2048.53 1983.52
0.2	7 5 2048.00 1982.97	8 5 2048.00 1982.97	9 5 2048.00 1982.97	9 4 2048.14 1983.11	10 4 2050.80 1985.87	11 5 2099.89 2036.99
0.3	6 5 2048.00 1982.97	7 5 2048.00 1982.97	8 5 2048.17 1983.15	8 4 2054.05 1989.33	9 4 2129.33 2066.41	10 5 2690.40 2641.62
0.35	5 4 2048.00 1982.97	6 4 2048.01 1982.98	7 4 2049.12 1984.11	8 5 2073.59 1009.69	9 6 2343.54 2290.37	9 5 3477.44 3457.69
0.4	5 3 2048.00 1982.97	6 5 2048.12 1983.09	7 5 2054.72 1989.83	7 4 2150.86 2090.88	8 5 2745.93 2787.13	5 8 3968.00 3965.93

M = 8 N = 16 L = 128 chips

P_{align} = N/L = 0.06250

P_b: probability of bit error in the presence of the signal
 P_b': probability of bit error in the absence of the signal

N: number of TADs

MN: number of SDCs

M: number of bits per TAD

TH: threshold

BEST TH (units of bits/TAD)
 BEST MN
 MEAN (units of bits)
 STD DEVIATION (units of bits)

TABLE 3.1: BEST TH, BEST MN, CORRESPONDING OPTIMUM MEAN AND STANDARD DEVIATION UNDER VARIOUS BIT ERROR PROBABILITIES P_b AND P_b'

$P_b' \backslash P_b$	0.1	0.2	0.3	0.4	0.5	0.6
0.1	3 6 1024.00 957.86	4 6 1024.00 957.86	5 9 1024.00 957.86	5 7 1024.00 957.86	5 5 1024.01 957.87	6 9 1024.24 958.12
0.2	3 6 1024.00 957.86	4 10 1024.00 957.86	4 8 1024.00 957.86	4 5 1024.08 957.95	5 10 1025.73 959.75	5 7 1046.92 982.38
0.3	3 10 1024.00 957.86	3 8 1024.00 957.87	3 5 1024.17 958.05	4 9 1027.11 961.13	4 7 1062.69 999.96	4 5 1381.91 1344.83
0.35	2 6 1024.00 957.86	3 9 1024.01 957.87	3 7 1024.67 958.60	4 11 1046.02 988.83	4 9 1150.82 1092.85	4 8 1699.39 1678.26
0.4	2 7 1024.00 957.86	3 11 1024.14 958.01	3 9 1027.38 961.61	6 1078.55 1017.49	4 12 1396.32 1355.18	1 7 1920.00 1915.73

$M = 16$ $M = 8$ $L = 128$ chips

$P_{align} = M/L = 0.12500$

- P_b : probability of bit error in the presence of the signal
- P_b' : probability of bit error in the absence of the signal
- M : number of TADs
- MN : number of SDCs
- N : number of bits per TAD
- TH : threshold
- L : code length

BEST TH (units of bits/TAD)
 BEST MN
 MEAN (units of bits)
 STD DEVIATION (units of bits)

TABLE 3.27 : BEST TH, BEST MM, CORRESPONDING OPTIMUM MEAN AND STANDARD DEVIATION UNDER VARIOUS BIT ERROR PROBABILITIES P_b AND P_b'

P_b \ P_b'	0.1	0.2	0.3	0.4	0.5	0.6
0.1	2 21 512.00 443.41	2 14 512.00 443.41	2 11 512.00 443.41	2 8 512.00 443.41	3 24 512.02 443.43	3 21 512.21 443.64
0.2	1 9 512.00 443.41	2 22 512.00 443.41	2 19 512.00 443.41	2 16 512.02 443.43	2 12 512.64 444.19	2 8 528.02 462.95
0.3	1 13 512.00 443.41	1 9 512.00 443.41	1 5 512.79 444.36	2 24 516.35 448.49	2 21 535.62 471.01	2 16 649.44 579.98
0.35	1 16 512.00 443.41	1 11 512.01 443.42	1 7 513.44 445.17	2 27 538.91 475.17	2 25 603.33 549.88	2 24 822.13 798.69
0.4	1 19 512.00 443.41	1 14 512.06 443.48	1 10 515.28 447.49	1 6 562.87 504.67	2 28 738.30 705.17	1 32 896.00 886.81

$M = 32$ $M = 4$ $L = 128$ chips
 $P_{align} = M/L = 0.25000$

BEST TH (units of bits/TAD)
 BEST MM
 MEAN (units of bits)
 STD DEVIATION (units of bits)

P_b : probability of bit error in the presence of the signal
 P_b' : probability of bit error in the absence of the signal
 M : number of TADs
 MM : number of SDCs
 M : number of bits per TAD
 TH : threshold
 L : code length

TABLE 3.20 : BEST TH, BEST NM, CORRESPONDING OPTIMUM MEAN AND STANDARD DEVIATION UNDER VARIOUS BIT ERROR PROBABILITIES P_b AND P_b'

$P_b' \backslash P_b$	0.1	0.2	0.3	0.4	0.5	0.6
0.1	1 45 256.00 181.02	1 34 256.00 181.02	1 28 256.00 181.02	1 22 256.00 181.02	1 17 256.00 181.02	1 12 256.08 181.16
0.2	1 54 256.00 181.02	1 31 256.00 181.02	1 47 256.00 181.02	1 43 256.03 181.07	1 38 256.38 181.58	1 32 259.94 186.93
0.3	1 60 256.09 181.14	1 58 256.45 181.61	1 56 257.76 183.32	1 54 261.97 189.13	1 52 275.77 209.36	1 49 321.45 271.36
0.35	1 61 256.88 182.09	1 60 258.87 184.90	1 59 264.07 192.64	1 57 277.07 209.51	1 56 309.41 255.69	1 56 375.67 345.33
0.4	1 62 259.75 185.74	1 61 266.04 193.91	1 60 278.79 210.74	1 59 304.98 245.62	1 59 354.19 314.74	1 64 384.12 362.15

$N = 64$ $M = 2$ $L = 128$ chips
 $P_{align} = N/L = 0.50000$

P_b : probability of bit error in the presence of the signal
 P_b' : probability of bit error in the absence of the signal
 N : number of TADs
 M : number of SDCs
 L : number of bits per TAD
 TH : threshold
 L : code length

BEST TH (units of bits/TAD)
 BEST NM
 MEAN (units of bits)
 STD DEVIATION (units of bits)

TABLE 3-2B : BEST TH, BEST NM, CORRESPONDING OPTIMUM MEAN AND STANDARD DEVIATION UNDER VARIOUS BIT ERROR PROBABILITIES P_b AND P_e .

$P_b \backslash P_e$	0.1	0.2	0.3	0.4	0.5	0.6
0.1	1 128 128.00 0.15	1 128 128.00 0.15	1 128 128.00 0.15	1 128 128.00 0.15	1 128 128.00 0.15	1 128 128.00 0.15
0.2	128 128.43 7.45 0.15	1 128 128.43 7.45	1 128 128.43 7.45	1 128 128.43 7.45	1 128 128.43 7.45	1 128 128.43 7.45
0.3	1 128 133.66 27.49	1 128 133.66 27.49	1 128 133.66 27.49	1 128 133.66 27.49	1 128 133.66 27.49	1 128 133.66 27.49
0.35	1 128 139.95 40.90	1 128 139.95 40.90	1 128 139.95 40.90	1 128 139.95 40.90	1 128 139.95 40.90	1 128 139.95 40.90
0.4	1 128 149.38 56.51	1 128 149.38 56.51	1 128 149.38 56.51	1 128 149.38 56.51	1 128 149.38 56.51	1 128 149.38 56.51

$N = 128$ $M = 1$ $L = 128$ chips
 $P_{align} = N/L = 1.00000$

P_b : probability of bit error in the presence of the signal
 P_e : probability of bit error in the absence of the signal
 N : number of TADS
 NM : number of SDCs
 M : number of bits per TAD
 TH : threshold
 L : codeword length

BEST TH (units of bits/TAD)
 BEST NM
 MEAN (units of bits)
 STD DEVIATION (units of bits)

CHAPTER IV

IMPLEMENTATION AND TESTING OF THE SEQUENCE DETECTION CIRCUIT ACQUISITION TECHNIQUE

We have built a prototype implementing the theory in Chapter III. The system is composed of the tapped analogue delay (TAD) circuits and the SD sequence detection circuits (SDC). As previously mentioned the code length is divided into 8 subcodes. The TAD circuit system is programmed and used to detect the presence of these 8 subcodes and the SDC system is to check if they are in appropriate sequence. If this system detect that the 8 subcodes are present and in-sequence it will give a synchronous pulse for the system to go to the acquisition checking mode (verification or tracking).

The system is to detect and synchronize to a code length of 256 chips and a chip rate of 1 M chips/second. The code is divided into 8 shorter subcodes each of which is of 32 chips. Each chip is 1 μ -sec long.

4.1 The TAD circuits

The TAD circuit consists of 8 Reticon TADs which are used to detect the 8 subcodes of the code length and 8 level detectors (one for each TAD) (Fig. 4.1).

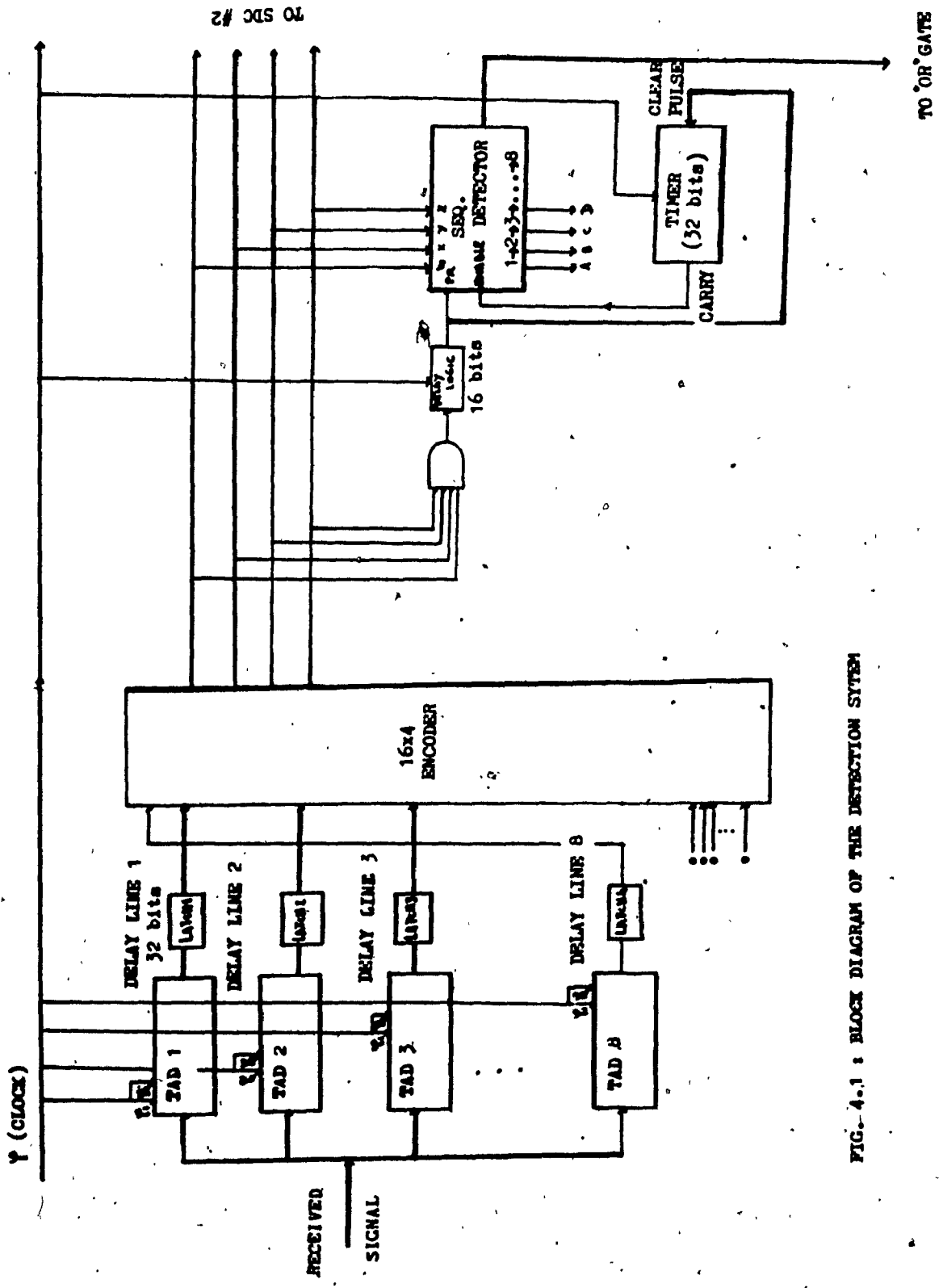


FIG. 4.1 : BLOCK DIAGRAM OF THE DETECTION SYSTEM

Each TAD is programmed so that wherever there is a match with the subcode it is to detect it will give a (detection) pulse. The level detector compares this pulse with a reference voltage used as the threshold. If the amplitude of the pulse exceeds this threshold voltage. The level detector will give a detection pulse. These 8 detection pulses corresponding to the subcodes will go to the SDC system to be checked if they are in-sequence.

4.2 The SDCS

Each SDC receives the pulses from the 8 TADS. It has 8 inputs and one output. The SDC will give a pulse when ever the TADS' pulses are in sequence. In another words, the SDC_i is used to detect the sequence

$$i \rightarrow i + 1 \rightarrow i + 2 \rightarrow \dots \rightarrow 8$$
$$i = 1, 2, \dots, 8$$

The SDC consists of the latches an 8 x 4 encoder, the counters and the main part (where details to be shown later) (Fig. 4.2).

The latches are used to reestablish the duration time of the pulse to 32 - μ sec long.

The encoder is used to encode the pulses. It has 4 outputs. If the pulse is pulse #1 (pulse from TAD 1), it will give a 0001₂

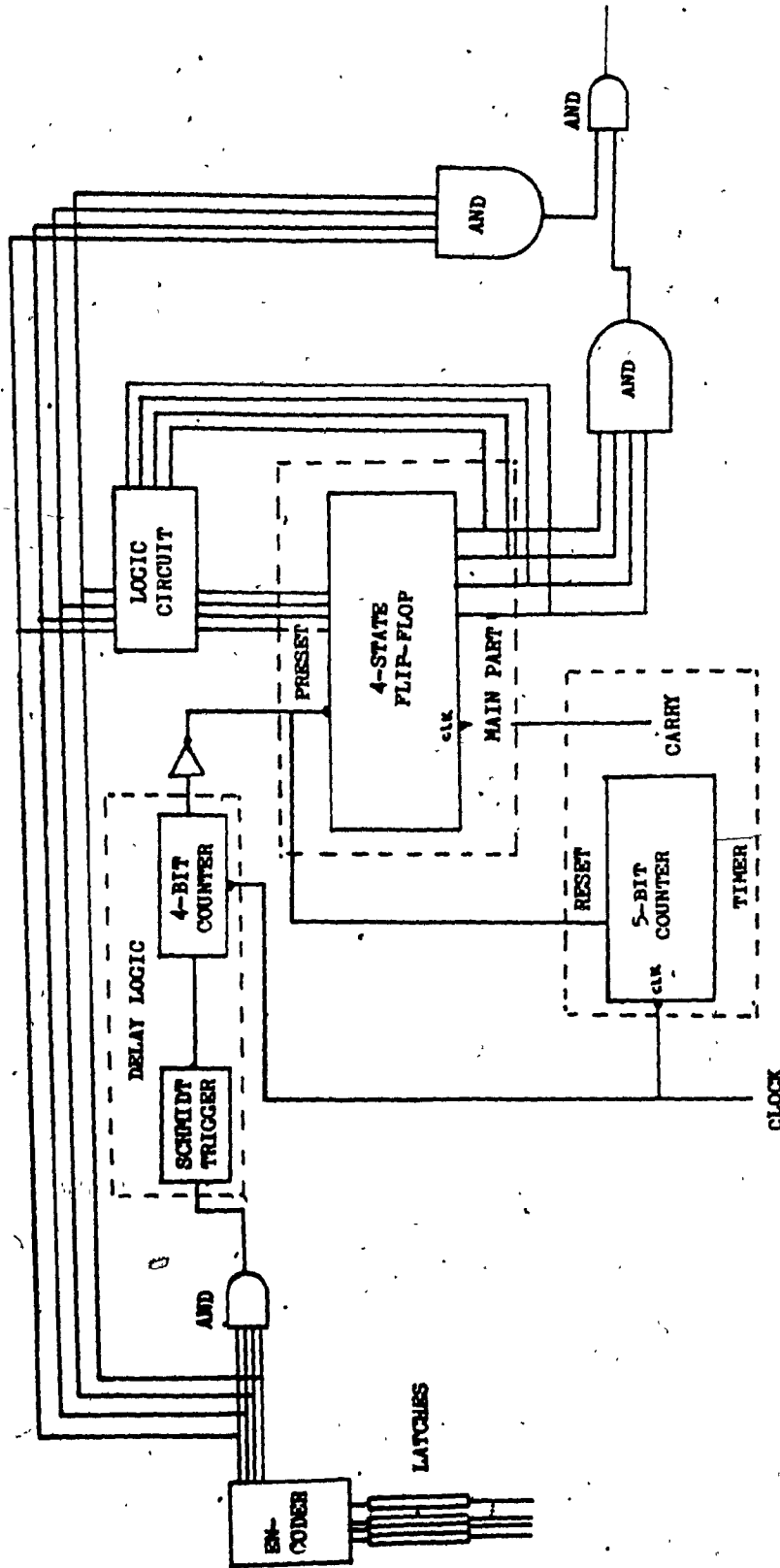


FIG. 4.2 : DETAILED STRUCTURE DIAGRAM OF AN SDC

and so on. If there is no detection pulse the output will be 0000_2 (state zero or waiting state). In this way the presence of a subcode will be represented by a 4-bit data.

If two or more pulses occurs at the same time priority will be given to the high-valued pulse, that is, pulse 8 has priority over pulses 7, 6, 5, ..., pulse 7 has priority over 6, 5, 4, ... and so on. Pulse 8 has highest priority, wherever. Pulse 8 (it does not matter these are other pulses or not) the encoder will encode as 1000_2 . Pulse 1 has the lowest priority.

The main part of the SDC is 2 J-K Flip-Flops. The main part has four inputs and four outputs (4-bit state). For the i th SDC, When ever the input is i_{10} (base 10) the Flip-Flops will be preset to i_{10} (for example if $i_{10} = 1$ the state (outputs) of the Flip-Flop will be preset to 0001. Wherever, the preset state is j_{10} where $j_{10} = 0, 1, 2, \dots$ and the input is $(j + 1)_{10}$ the Flip-Flops will change state to $(j + 1)_{10}$, other wise the Flip-Flops return to state 0_{10} (the waiting state). If the input becomes again i_{10} (the preset value) the Flip-Flops will be preset again to i_{10} , as stated above.

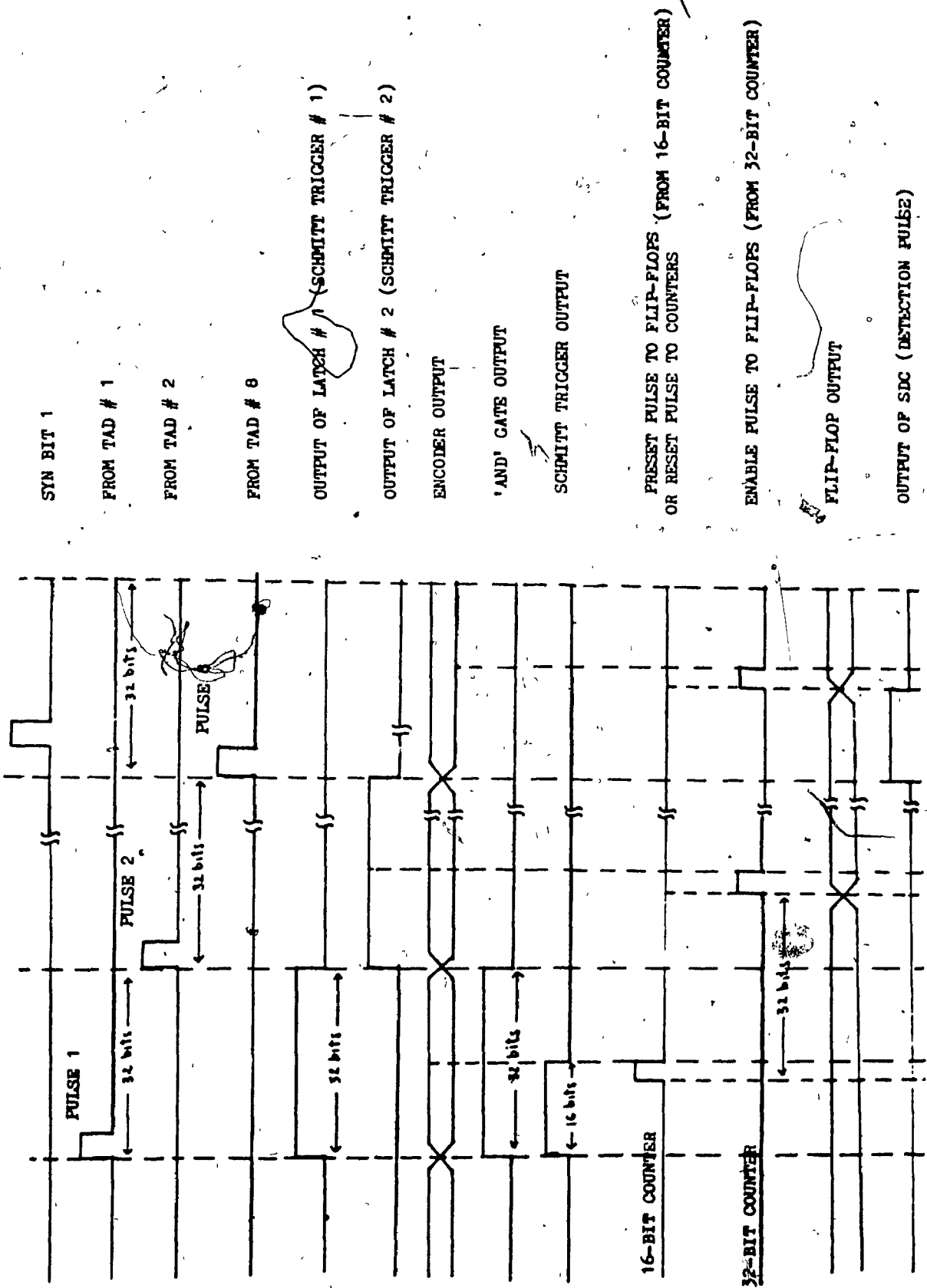
By this arrangement the Flip-Flops will only change state wherever their input has been encoded from a TADs' detection pulse representing the subcodes which is in-sequence with the preceding subcode otherwise it will return to the waiting state (state 0) or is preset again if the input pulse represents the i th subcode.

To generate the preset pulse of each SDC, a 4-input AND gate is used. It gives a 'high' output whenever it detects the value i_{10} from the output of the encoder. This high output will be latched for the duration of 16 chips. A counter uses this 16- μ sec pulse as its enable pulse and starts counting at the positive edge of the high AND gate output. This mechanism will generate a pulse after a 16-chip time. This pulse is used to preset the Flip-Flops (the reason for the preset pulse to be delayed 16 bit is to make sure that the Flip-Flops will be preset (or change state, as we will see below) at the middle of the duration time of each input (Fig. 4.3)

The preset pulse is also used to reset a 32-bit counter. This counter after resetting to 0_{10} will start counting and generates a pulse after every 32-bit time. This carry pulse from the 32-bit counter is used as an enable pulse to the Flip-Flop. Thus after the Flip-Flops is preset it can only change state after 32-bit time. Thus this will be at the middle of the next input. If the input is again i_{10} the Flip-Flops will be preset again and the 32-bit counter is reset to 0_{10} and the procedure (checking for in-sequence pulses) will start over. If the input pulse is not in-sequence the Flip-Flops will return to 0_{10} output (waiting state).

The detection pulse of the SDC is generated by using AND gates. These AND gates will give a high pulse wherever the present state of the Flip-Flops is 7_{10} and the input is 8_{10} . Since 7_{10} is followed by

FIG. 4.5 : TIMING DIAGRAM OF THE PROTOTYPE SYSTEM



8_{10} and state 7_{10} occurs only when the previous state is 6_{10} and state 6_{10} in turn only occurs when there has been state 5_{10} , state 4_{10} , ..., and the preset state i_{10} thus this ensures the AND gates only giving the detection pulse when there is the sequence $i \rightarrow i + 1 \rightarrow i + 2 \rightarrow \dots \rightarrow 8$. This detection pulse (from the SDC) will be "ored" with the output pulse of other SDC, and the results pulse is used for tracking.

The details of the prototype sequence of operation will be outlined next.

4.3 The detailed algorithm of the prototype

- 1- Power-on reset will clear the contents of all registers.
- 2- System 1 is waiting for delay line 1 (Fig. 4.1) to peak (other systems waiting for 2, 3, or 4... to peak).
- 3- When delay line (DL) 1 peaks, it forces a latch to momentarily go high (for a duration long enough to enable the sequence detector to read the encoder output).
- 4- The delay line 1 pulse will cause the encoder output to be (0001). This will propagate through the "AND" gate and appears as a high signal at the "AND" gate output.

- 5- This signal after propagating through some logic delay* and being transformed to a pulse will be used to
 - preset the sequence detector to state (ABCD = 0001).
 - reset (clear) the timer and the timer will count starting again from 0000.

- 6- After a set time (the timer will count 32 clock bits, sufficient time for DL 2 to peak and the value remains in the latch. Recall: the latch value will be reset shortly after its corresponding delay line peaks - step (3)). The timer will produce a carry, which will be used to clock in the next encoder output.

- 7- The next encoder output, corresponding to the next delay line pulse, will appear at the input of the sequence detector.

- 8- If it is correct (i.e. the proper one in the sequence) it will be loaded into SEQ detector and the state of the SEQ detector will be set to the value of this input. If proper sequence string occurs (i.e. 1 → 8) carry will result and be used for synch. (The carry will be $CAR = BCD + XYZ$ because only when the state of the SEQ detector goes from 1 through 8 there will be ABCD = [0111] (state 7) and WXYZ = [1000] (→ state 8) and the carry will be generated (CAR = 1).

* Since this delay is long enough the encoder output will be waiting to be loaded into the sequence detector (if the delay chosen to be of 16 clock bits (pulses) long, this (the loading) will happen approximately at the midpoint of the duration of the encoded output signal.

- 9- If the new input is out of sequence, it will cause the seq. detector state to be reset to [0000] and the process returns to step (2).

4.4 Testing performance of the system

In our built system we used only 5 SDC, (from SDC1 to SDC5). Thus whenever 1 or more of these 5 SDCs peak the system will give a detection pulse.

To count the correct detection pulse N_D we "AND" this pulse with the SYN BIT of the received signal and use the result output to trigger a counter display.

We also count the total number of pulse (false and correct) directly from the "OR" gate output of the SDCs system. The number of false detection pulses N_{FA} is then calculated by subtracting the total number of the pulses recorded with the number of correct pulses.

We set up the circuit, feed the signal and use the signal generator to simulate the noise. Photographs of the testing system, the prototype modern implemented are shown in Figures 4.4 to 4.7.

After feeding the signals and noise, we have the system activated for a period of time, t_{test} , and count the number of correct pulse and the total number of pulses which are counted by two counter displays.

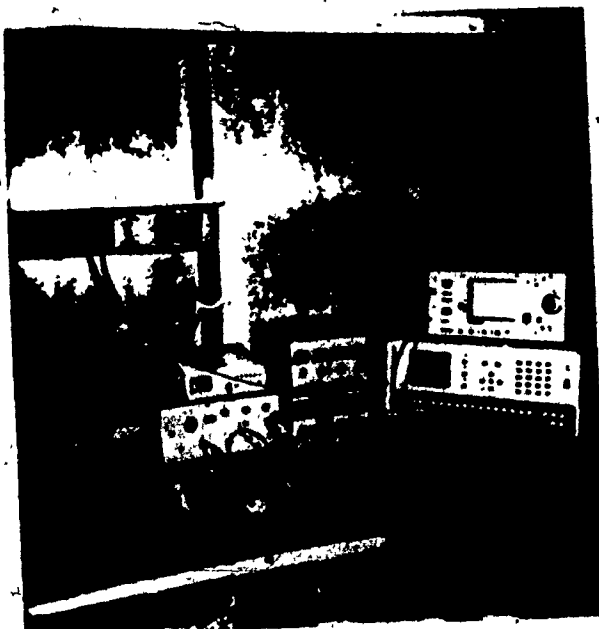


FIG. 4.4 : TESTING EQUIPMENT

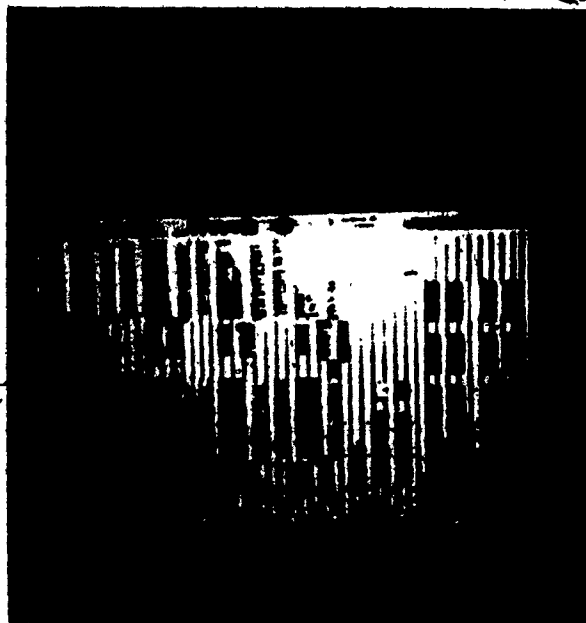


FIG. 4.5 : TAD CIRCUITS BOARD



FIG. 4.6 : SDCs BOARD



FIG. 4.7 : CIRCUIT BOARD CONNECTION
AND LAYOUT

The code acquisition time \bar{T} is calculated by multiplying N_{FA} with the penalty time and added to the test time t_{test} . The result is divided by the no of correct detector pulses to give \bar{T} as shown in the following formula

$$\bar{T} = \frac{t_{test} + (K + 1) \times 256 \times N_{FA} \times 10^{-6}}{N_D}$$

We did a number of tests, in each tests we check the behaviour of the system with respect to a parameter by changing the value of this parameter in each test.

The results of the testings are shown below.

TEST RESULTS

A - TEST CONDITIONS

** Power supplies

* TADS: $V_{DD} = +15V$, $V_{SS} = GND$

$V_{BB} = +14V$, $BIAS = +5V$

* OP. AMPS: $V_+ = +15V$, $V_- = -15V$

* TTL components: $V_{CC} = +5V$, $GND = 0V$

** Clock: TADS = 2 MHz $\phi_1 = 1MHz$

$\phi_2 = 1MHz$

$SDC_5 = CLK = 1MHz$ (Gated clock from signal generator)

** SIGNAL IN: 256 bits of 8 subsections (32 bits/subsection)

bit duration time: $1 \mu s$

B - RESULTS

4.4.1 Test result report 1

In this test we varied the value of the SNR

= 0.13V

= 10 min or 600 secs

Results:

a) Test 1: $SNR = 1 \approx$ odb

$N_{TOT} = 118,041$

$N_D = 103,704 \rightarrow N_{FA} = 118,041 - 103,704 = 14,337$

$$\bar{T} = \frac{600 + 2 \times 256 \times 14,337 \times 10^{-6}}{103,704}$$

= 5856.48 μ secs or 22.88 code lengths

b) Test 2: SNR = 0.673 \approx -2db

$$N_{TOT} = 83,933$$

$$N_D = 70,571$$

$$N_{FA} = 83,933 - 70,571 = 13,362$$

$$\bar{T} = \frac{600 + 2 \times 256 \times 13,362 \times 10^{-6}}{70,571}$$

= 8599.02 μ secs or 33.59 code lengths

c) Test 3: SNR = 0.563 \approx -3db

$$N_{TOT} = 76,095$$

$$N_D = 57,256$$

$$N_{FA} = 76,095 - 57,256 = 13,489$$

$$\bar{T} = \frac{600 + 2 \times 256 \times 13,489 \times 10^{-6}}{57,256}$$

= 10,603.46 μ secs or 41.42 code lengths

we see that whenever the SNR decreases, the acquisition time increases.

4.4.2 Test result report 2

(Date 09-08-1986)
at 12:00:00

T_{test} : 30 min or 1800 secs

$$SNR = \frac{(1/1)^2}{(1/0.820)^2} = \frac{(.820)^2}{1^2} = 0.673 \approx -2db$$

Test 1: 5 SDCs

SIGNAL + 1 NOISE SOURCE (PSEUDO-NOISE)

= 0.52V

NOISE sequence: 6A9E DCF7 6521 3045

89D2 165A ED5G 2BF7

39CD 8456 ADEC F9BA

10A5 2E39 CF2G DA54 (in Hex)

Results:

$$N_{TOT} = 1,108,557 \quad N_{FA} = 1,108,552 - 633,276 = 475,281$$

$$N_D = 633,276$$

$$\bar{T} = \frac{1800 - 2 \times 256 \times 475,281 \times 10^{-6}}{633,276}$$

$$= 3,226.62 \mu \text{ secs or } 12.60 \text{ code lengths}$$

Test 2: 4 SDCS

SIGNAL (NO PSEUDO NOISE)

= 0.01 V

$$N_{TOT} = 22,428 \quad N_{FA} = 22,428 - 22,388 = 40$$

$$N_D = 22,388$$

$$\bar{T} = \frac{1800 + 2 \times 256 \times 40 \times 10^{-6}}{22,388}$$

$$= 80,401.13 \mu \text{ secs or } 314.07 \text{ code lengths.}$$

Conclusion: when the number of SDCS changes from 5 to 4 (decreases) the mean acquisition time increases.

4.4.3 Test result set 3

(24-9-86)
8:30:00 pm

NN = 5SDCS

t_{test} = 20 minutes or 1200 secs

1) Under no interference noise

$$SNR = \frac{(5)^2}{(4)^2} = \frac{25}{16} = 1.56 \approx 4.46 \text{ dB}$$

Voltage level: 5 units (5V)

V_{ref} : -0.31V

$$N_D = 631,277 = N_{FA} = 1,022,379 - 631,277$$

$$N_{TOT} = 1,022,379 = 391,102$$

$$\bar{T} = \frac{t_{test} + (K+1) \times 256 \times N_{FA} \times 10^{-6}}{N_D}$$

$$= \frac{1200 + 2 \times 256 \times 391,102 \times 10^{-6}}{631,277}$$

$$= 2,218.11 \text{ } \mu\text{Sec or } 8.66 \text{ code lengths}$$

2) Single interference noise

$$V_{ref} \text{ } -0.31V \quad SNR = \frac{(5)^2}{(4)^2} = \frac{25}{16} = 1.56 \approx 4.46 \text{ dB}$$

$$N_D = 511,855 = N_{FA} = 536,682 - 511,855$$

$$N_{TOT} = 536,682 = 24,827$$

$$\bar{T} = \frac{1200 \times 2 \times 256 \times 24,827 \times 10^{-6}}{511,855}$$

$$= 2,369.25 \mu\text{Sec or } 9.25 \text{ code lengths}$$

3) 2 interference noises

$$V_{\text{ref}} : \pm 0.31V \quad \text{SNR} = 4.46 \text{ dB}$$

$$N_D = 17,365$$

$$\rightarrow N_{FA} = 25,557 - 17,365$$

$$N_{TOT} = 25,557$$

$$= 8,192$$

$$\bar{T} = \frac{1200 + 512 \times 8,192 \times 10^{-6}}{17,365}$$

$$= 69,346 \mu\text{Sec or } 270.88 \text{ code lengths}$$

4) 3 interference noises

$$: -0.31V \quad \text{SNR} = 4.46 \text{ dB}$$

$$N_D = 20$$

$$N_{FA} = 38,072 - 20 = 38,052$$

$$N_{TOT} = 38,072$$

$$\bar{T} = \frac{1200 + 512 \times 38,052 \times 10^{-6}}{20}$$

$$= 60,974,131 \mu\text{Sec or } 238,180.206 \text{ code lengths}$$

4.4.4 Conclusion

From the above three tests we see that the result is in consistent with the result shown in Chapter II. The acquisition time \bar{T} for some values of P_b and P_b' (i.e. the SNR or noise level) increases as the noise sources number and amplitude increases and decreases as the number of the SDCs increases.

CHAPTER V

A HYBRID SEQUENTIAL/PARALLEL SCHEME FOR COMBINED ACQUISITION/SEARCH LOCK OF SPREAD SPECTRUM CODES

5.1 Introduction

Several techniques have been proposed for minimizing the mean acquisition time in spread spectrum communications. Among those currently attracting interest are the multi-dwell [3] and parallel acquisition methods [4]. Also, [6] describes a dwell matched filter technique for SS codes acquisition. Recently a parallel processing technique [2] using the SDCs (sequence detection circuits) was proposed, and the mean acquisition time was shown to be improved. The optimal case of this scheme has also been studied by the authors in [8]. However, with this technique the threshold had to be exactly equal to a fixed number corresponding to a certain SDC and all SDCs were programmed to end with the same code subsection (among other conditions as well). In this paper we propose a new system in which these assumptions are relaxed, meaning that, some subcode detection pulses may be missing but synchronization can still be declared.

The new technique can be described as follows:

- (1) A long code is divided into a number of code subsections (subcodes) and a bank of matched filters (M.F.)^S is programmed to receive these different subcodes.

- (2) A number of transitions counting circuits try to count the number of subcodes coming in sequence. The 1st circuit starts at the very beginning, considers the first peaking subcode as legitimate and counts the total number of subsequent subcodes coming in the right order and compatible with the first subcode (pivot subcode). The 2nd circuit performs the same function except that it starts its operation delayed by 1 subcode from the 1st circuit. The 3rd circuit start is delayed by 2 subcodes and so forth.
- (3) At the end of one codelength (trial period) comparison will be made by a code combiner block only among those transition counts of different circuits which exceed their prespecified thresholds. The right code phase then (i.e. whether the starting subcode phase should be 3 or 1 or 7 etc.) will be the one identified by the circuit having the maximum count.

To enhance subsequent search lock (code verification) [1], it is seen that this new acquisition scheme while looking similar to [2] differs in the way of utilizing the inherent sequence in the long code. While [2] assumes the reception of completely independent code subsections, our scheme utilizes this inherent coding efficiently and tolerates partial loss of subcodes due to jamming, etc. For example, if the 1st received subcode samples are greatly distorted by jamming the

decision from the first circuit will be erroneous and will be neutralized by hopefully the correct decision of the other circuits. This provides the first measure of jamming tolerance during acquisition.

The second measure is provided by noting that each circuit can jump few subcodes sections received in heavy jamming and still provide the right code phase as will become clear shortly.

In the following, we describe in details the new sequential/parallel scheme. The mean and variance of the acquisition time will be found and computed in terms of the various thresholds and many parameters involved. These mean and variance actually pertain to the total (acquisition + verification times).

5.2 The new acquisition/search lock scheme

The problem is to acquire the phase (code epoch) of a direct sequence (DS) spread spectrum signal. Fine acquisition (tracking) is assumed achieved by the use of the appropriate non-coherent tracking or tau-dither loops [9]. Also, code rate changes due to Doppler and other effects are not treated. However, their effects on the mean and standard deviation of the acquisition time can be easily evaluated [1]. The long code (or in some systems the short synchronization preamble) is divided into N subsections each consisting of (M) DS chips. At the receiver N matched filters (M.F.)^S are programmed to match the N subsections of the code to be acquired. In the process of programming the matched filters the effects of the carrier phase will be accounted for [1].

The receiver is simplified in Fig. (5.1) with the only emphasis put on the initial code acquisition building blocks. If the local code is in perfect coarse and fine acquisition with the received signal code then this will be indicated as a high signal at the output of an integrate and dump filter following the front mixer of Fig. (5.1). If this high signal exceeds a certain final threshold (TH^F) then synchronization will be declared and meaningful data demodulation will start. This integrate, dump and threshold comparison will continue with data demodulation and the current code phase (epoch) will be retained as long as this threshold is exceeded. While this is taking place, the

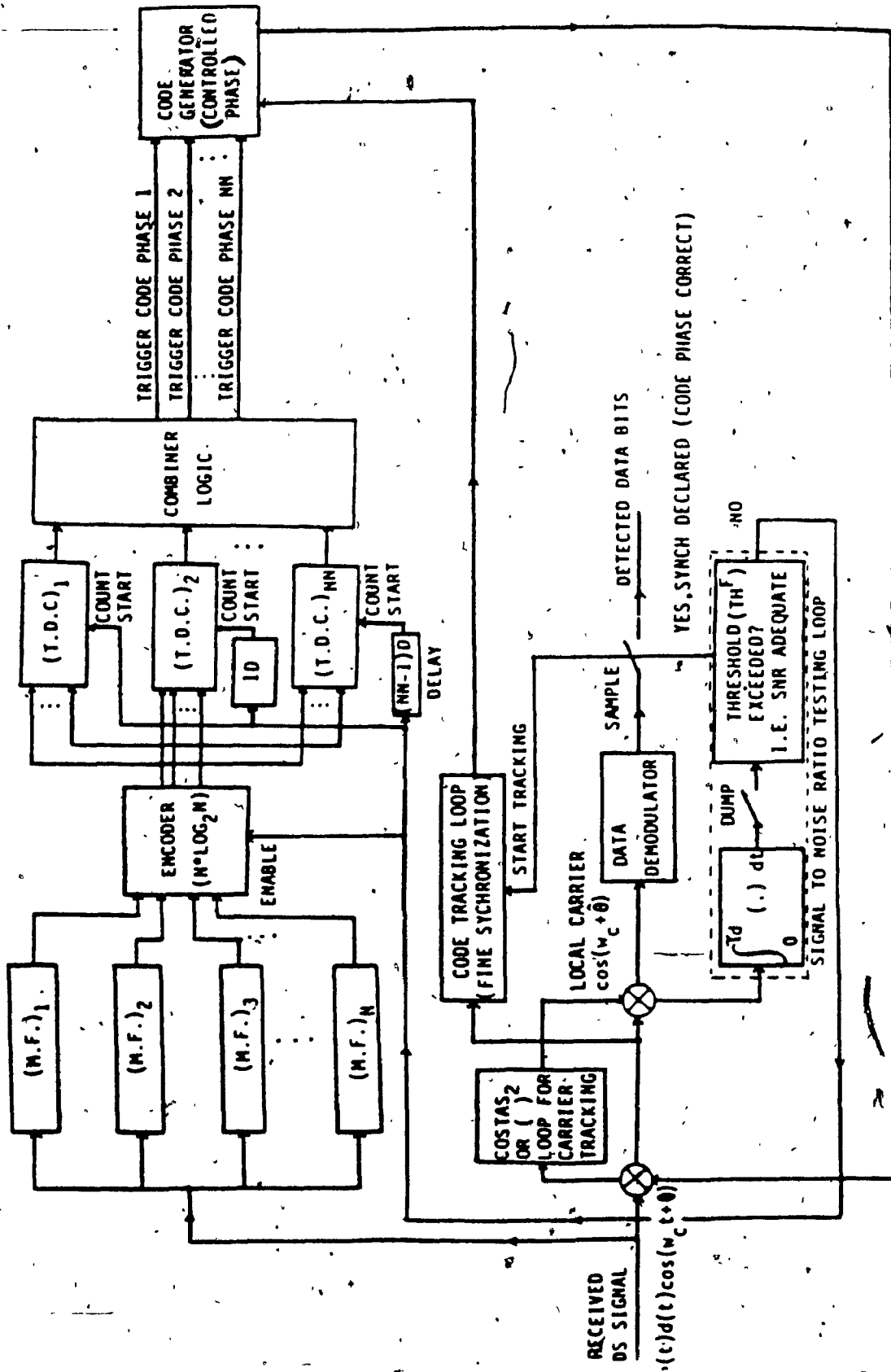


FIG. 5.1 : DS RECEIVER BLOCK DIAGRAM WITH EMPHASIS ON ACQUISITION DETAILS.

received signal is continually fed into the N matched filter banks. However, the encoder, the (NN) Transition Detection Circuits (T.D.C.)^S and the combiner logic (Fig. 5.1) are disabled as long as the current code phase is retained meaning that the received samples stored in the N (M.F.)^S are not needed. Once synchronization is lost (as indicated by the aforementioned threshold comparator) or once we start receiving, the $(N \cdot \log_2 N)$ encoder, the (T.D.C.)^S and the combiner logic are enabled. At the end of each M chips of the received code, only one (M.F.) (hopefully one) will peak i.e., the number of agreements of chips of the received code section with the stored replica exceeds a certain (M.F.) threshold (TH). In case more than one (M.F.) exceeds its threshold having received the same code subsection (M chips), only one of them is randomly selected (by the encoder). The encoder dumps a binary number consisting of $(\log_2 N)$ bits (representing the identity of the subcode received) at the end of each received code subsection to the inputs of (NN) (T.D.C.)^S. The sequence of numbers generated by the arrival of successive code subsections is what is shown in Figures (5.3) and (5.4) (in decimal). Fig. (5.2) depicts the sequence of events taking place in the (T.D.C.)^S and combiner logic one the loss of sync is declared. - The loss-of-sync pulse triggers the first circuit immediately, triggers the second after M chips (delay D), triggers the third after $2 M$ chips (delay $2D$)... etc. The reason for these delays will be explained shortly.

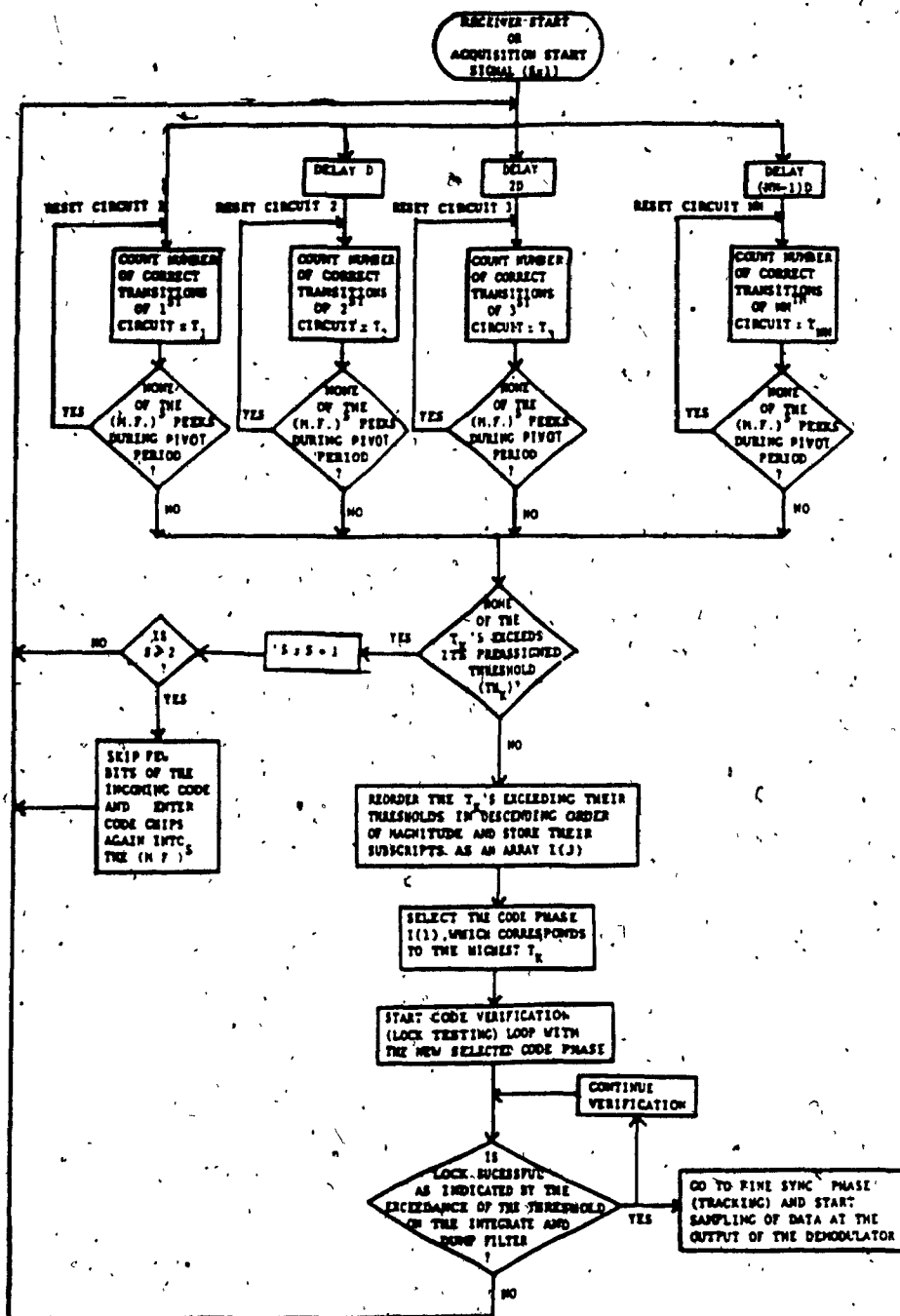


FIG. 5.2 : DETAILS OF THE COMBINER LOGIC AND (T.D.C.)^S ACTIONS

NOTE: THE PARAMETER S IS USED AS A PROTECTION AGAINST DEAD LOCK, FOR EXAMPLE IF $S > 2$ IN THE FIG. THIS INDICATES THAT DURING 2 CODE LENGTHS (2 CODE TRIALS) NONE OF THE (T.D.C.) COUNTS EXCEEDS THE THRESHOLD ($T_i < T_{th}$). IN THIS CASE FEW CHIPS OF THE ARRIVING CODE ARE SKIPPED BEFORE BEING FED AGAIN INTO THE (M.F.)^S.

Once enabled, the first (T.D.C.) accepts the number of the (M.F.) that peaks as an established (pivot) and waits for the subsequent (M.F.) to peak in the next code subsection interval and an accumulator T_1 is incremented with the occurrence of each correct subsection transition. As an example, the following M.F. peakings $\overbrace{4\ 1\ 2\ 7\ 8} \quad \overbrace{1\ 5\ 7}$ implies $T_1 = 3+1 = 4^*$, while during the same period of time the second (T.D.C.) will ignore the first number 4 (because of the previously mentioned delay), consider 1 as the right start and count $\overbrace{1\ 2\ 7\ 8} \quad \overbrace{1\ 5\ 7}$ resulting in $T_2 = 2+1 = 3$.

Corresponding to the same received DS subsection, the third (T.D.C.) ignores 4 and 1 and considers 2 as the pivot (right code start) and count $\overbrace{2\ 7\ 8} \quad \overbrace{1\ 5\ 7}$ resulting in only one transition i.e. $T_3 = 1+1 = 2$. The fourth (T.D.C.) ignores 4 and 1 and 2 (having 3-D delay) and consider 7 as the correct code subsection start and count $\overbrace{7\ 8} \quad \overbrace{1\ 5\ 7}$ i.e. $T_4 = 2+1 = 3$.

Another typical example is self-illustrated in Fig. (5.4) while the ideal error-free reception of the code is shown in Fig. (5.3). The reader is urged to carefully inspect these figures since they reflect themselves on the performance analysis.

* A bias of 1 is added to the count of the (T.D.C.) whenever it detect the pivot subcode (right or wrong it does not matter). We note that it is possible that sometimes none of the (M.F.)^s will peak.

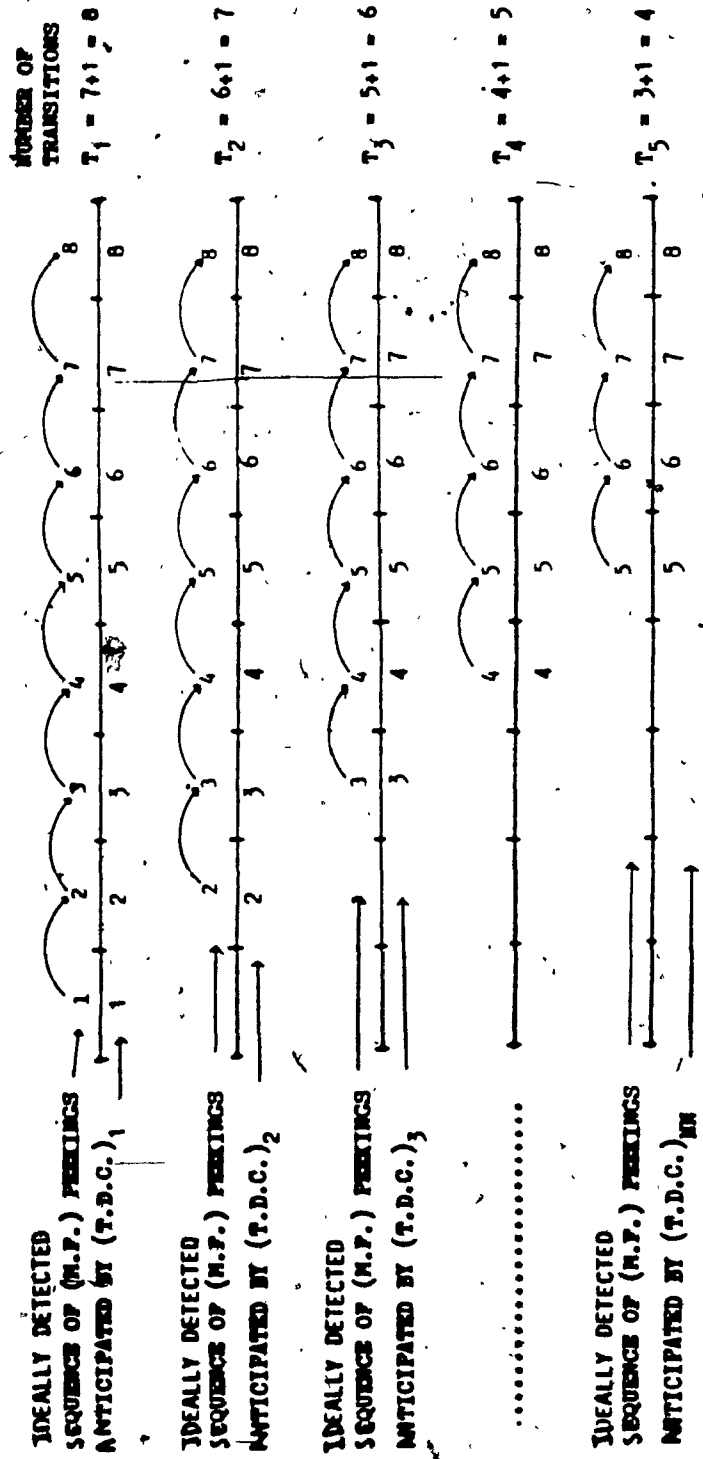


FIG. 5.3 : IDEAL CASE OF ERRORLESS RECEPTION OF THE SPREAD SPECTRUM CODE. THE FIRST (T.D.C.) COUNT $T_1 = 8$, THE SECOND COUNTS $T_2 = 7, \dots$ THE COMBINER SELECTS THE CODE PHASE GIVING THE MAXIMUM COUNT I.E. T_1 (SINCE ALL OF THE COUNTS EXCEED THEIR THRESHOLDS)

NOTE: A BIAS OF 1 IS ADDED TO THE COUNT OF EACH (T.D.C.) WHENEVER THIS CIRCUIT DETECTS THE PIVOT (CORRECTLY OR INCORRECTLY). IT IS POSSIBLE TO HAVE NO (N.F.) PEEKINGS DURING THE PIVOT.

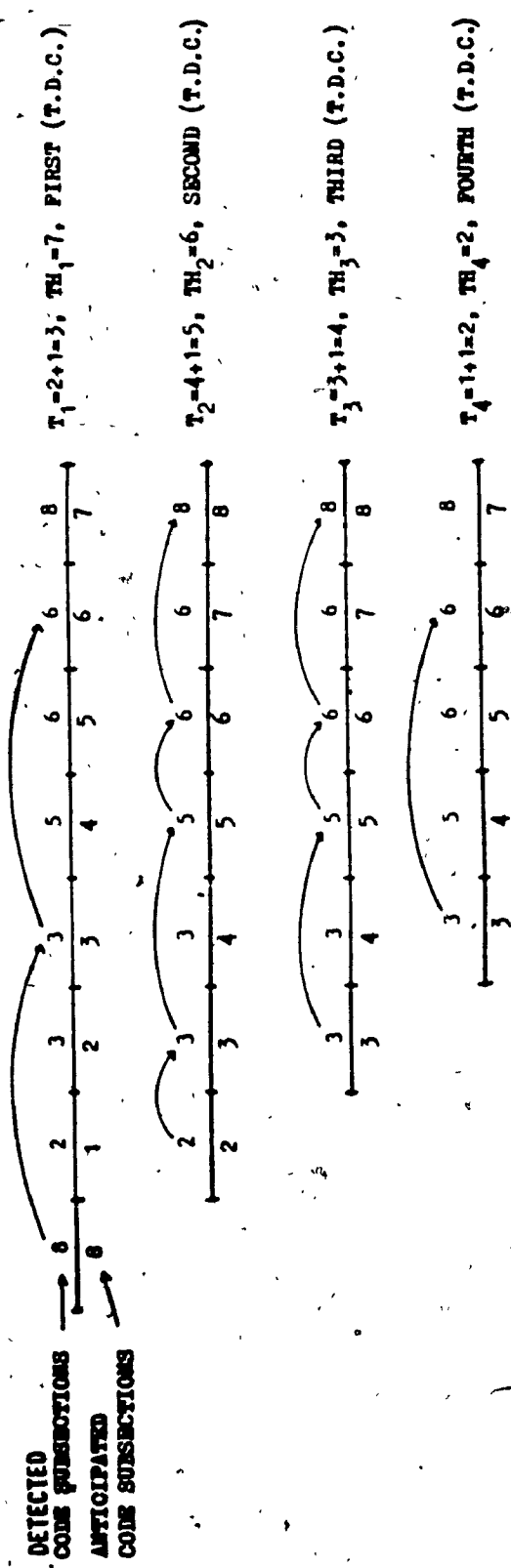


FIG. 5.4 : TYPICAL EXAMPLE FOR THE ACTION OF THE (T.D.C.)^m AND COMBINED LOGIC. HERE $m=8, M=4, N=1/M=1/8$. ACCORDING TO THE FLOW CHART OF FIG. (2) WE HAVE $I(1)=3, I(2)=4$, COMPARISON IS MADE ONLY AMONG THE (T.D.C.) COUNTS EXCEEDING THRESHOLD AND THE THIRD (1(1)-3) CODE PHASE IS SELECTED

NOTE: 1 IS ADDED TO ALL T. EXCEPT WHEN NONE OF THE (M.F.)^m PEAKS DURING THE PIVOT PERIOD, IN THAT CASE $T_k=0$.

The acquisition algorithm (stored in the combiner logic) proceeds by comparing each of $T_1, T_2, \dots, T_K, \dots, T_{NN}$'s exceed its threshold, the combiner logic will find the highest T_K by ordering these T_K 's as an integer valued array $I(j)$ where $j = 1, 2, \dots, U$ (U is the number of T_K 's exceeding their thresholds). In the aforementioned example $T_1 = 4, T_2 = 3, T_3 = 2, T_4 = 3$, if, for example, T_1, T_2 and T_4 exceed their corresponding threshold then $I(1) = 1, I(2) = 2, I(3) = 4$. The code phase indicated by $I(1)$, which corresponds to the highest T_K , is selected as the right code subsection.

The code generator is adjusted accordingly, and code verification starts (with the code phase (subcode) 1 being selected in the example). However, if none of the T_K 's exceeds its preassigned threshold, this is an indication that the specific received subsections have been terribly destroyed due, for example, to a pulsed jammer, etc., in which case a new record of the received signal must be shifted into the (M.F.)^S and the above procedure repeated (Fig. 5.2). This means inspecting another search cell of the uncertainty region of the code search. There might be some cases where none of the T_K 's exceed their threshold even though the acquisition procedure has been started over many times. To prevent this deadlock from happening, the number of repeated acquisition trials is limited (in our case it is 2, see Fig. (5.2). After which, a few bits (chips) of the incoming code will be skipped and the code is reentered into the (M.F.)^S.

5.3 Analysis

We start the analysis by evaluating α and β , i.e., the probability of one (M.F.) correctly peaking and falsely peaking respectively. We prefer (since we are going to change the length of the subcodes) to define α and β for different cases in terms of the intrinsic bit probabilities of miss and false alarms P_b and P'_b . These will be the same for all cases of division of the code into subcodes and so, fair comparison will prevail. Assuming independence of the various chips constituting one (M.F.) and a threshold (TH) for the M.F. to peak we obtain α and β as follows.

$$\alpha = \sum_{R=TH}^M \binom{M}{R} (1 - P_b)^R (P_b)^{M-R} \quad (5.1)$$

and

$$\beta = \sum_{Q=TH}^M \binom{M}{Q} (1 - P'_b)^{M-Q} (P'_b)^Q \quad (5.2)$$

In the literature α , β are defined by P_d , P_{fa} [9, p. 501] and they are based on a certain integration time (in our case, the length of one (M.F.) in seconds). It is important to compare the definition of P_d and P_{fa} of our scheme to those in the classic literature [2, pp. 492,501]. There, P_d and P_{fa} are based on a certain integration over a certain dwell time following despreading. Our α and β can be rela-

ted to P_d and P_{fa} if we make the length of each TAD equal to the dwell time of these approaches. By all means, whether they are P_d and P_{fa} or α and β they have to be assumed a priori and jointly.

The analysis is involved by the complexity of the (T.D.C.)^S and combiner logic. We start by evaluating the following probabilities and postpone the presentation of the acquisition time Markovian chain for now.

a = Probability that (the correct (M.F.) that should peak but did not while one or more other (M.F.)^S peaked, the correct (M.F.) was one of them, but was not selected by the random tie breaker of the encoder).

$$= (1-\alpha) \sum_{i=1}^{N-1} \binom{N-1}{i} \beta^i (1-\beta)^{N-1-i} + \alpha \sum_{i=1}^{N-1} \binom{N-1}{i} \frac{1}{i+1} \beta^i (1-\beta)^{N-1-i} \quad (5.3)^*$$

where α is the probability that a certain (M.F.) peaks when the code subsection passes through (i.e. probability of detection of a single (M.F.))

β is the probability that a certain (M.F.) peaks due to the wrong code subsection (i.e. probability of a (M.F.) false alarm).

In (5.3), $(1-\alpha)$ is the probability of miss of the correct M.F. peak and the first summation over the binomial coefficients represents

* Independence of all events leading to equations (5.3) -- (5.8) is more or less supported by physical independence of the (M.F.)^S. However, it is assumed to prevail.

all possibilities of one or more M.F. falsely peaking (each with prob. β).

The second term in (5.3) represents all those cases of existence of a tie between the (M.F.) matching the signal (with prob. α) and some of the other (M.F.)^S (i of them) peaking falsely (with prob. β). Multiplication by $\binom{i}{i+1}$ corresponds to picking one of the (i) falsely peaking (M.F.)^S out of the total (i+1) peaking (M.F.)^S. The combination $\binom{N-1}{i}$ of both summations in (5.3) represent all possibilities of picking i false peaking M.F. out of a total of (N-1) of them.

~~b~~ Probability that the correct (M.F.) (matching the arriving code subsection) has peaked (as well other (M.F.)^S) and was selected by the tie breaker (encoder).

$$b = \alpha \sum_{i=0}^{N-1} \frac{1}{i+1} \binom{N-1}{i} (\beta)^i (1-\beta)^{N-1-i} \quad (5.4)$$

The R.H.S. of (5.4) is similar to the second term of (5.3) except for the fact that the probability of picking the right (M.F.) by the breaker (out of (i+1) peaking (M.F.)^S) is now $\frac{1}{(i+1)}$. Worth mentioning here, is that for the first (T.D.C.) a, b are based on the first arriving code subsection following the initiation of the coarse acquisition during which the 1st (T.D.C.) will determine its pivot. Similarly, for the 2nd (T.D.C.) a, b will be based on the second code subsection

where its pivot is obtained and so on. For all (T.D.C.)^s and for the code subsections following that corresponding to pivot determination we define:

c = Conditional probability that the expected (M.F.) matched to the arriving code subsection missed the detection (did not peak) and at least one of the remaining (M.F.)^s has peaked conditioned on correct identification of the pivot (M.F.).

$$= (1-\alpha) \sum_{i=1}^{N-1} \binom{N-1}{i} \beta^i (1-\beta)^{N-1-i} \quad (5.5)$$

We note that the events constituting c will not increase the count of the applicable (T.D.C.).

d = Conditional probability that the expected (M.F.) matched to the arriving code subsection has peaked together with zero or more of the remaining (M.F.)^s. This is conditioned on correct identification of the pivot (M.F.).

$$= \alpha \sum_{i=0}^{N-1} \binom{N-1}{i} \beta^i (1-\beta)^{N-1-i} = \alpha \quad (5.6)$$

We note that the events constituting d will correctly increase the count of the (T.D.C.).

e = Conditional probability that the expected (M.F.) has not peaked due to the passage of the wrong subsection (no false alarms) but one or more of the other (M.F.)^s have falsely peaked. This is conditional on false determination of the identity of the pivot (M.F.).

$$= (1-\beta) \sum_{i=1}^{N-2} \binom{N-2}{i} \beta^i (1-\beta)^{N-i-1} \quad (5.7)$$

We note that the events constituting e will not increase the count of the (T.D.C.).

Interesting to see in (5.7) that the index of summation is (N-2) rather than (N-1) since the condition of the correct (with prob. α) or not correct (with prob. $(1-\alpha)$) of the subcode in sequence following the wrong determination of the pivot is irrelevant.

f = Conditional probability that one of the (M.F.)^s has suffered a false alarm so has a zero or more of the remaining (N-2) (M.F.)^s. This is again conditional on wrong determination of the pivot subcode.

$$= \beta \sum_{i=0}^{N-2} \binom{N-2}{i} (\beta)^i (1-\beta)^{N-i-1} = \beta \quad (5.8)$$

We note that the events constituting f will falsely increase the count of the applicable (T.D.C.).

Here too we point out that c, d, e, f are calculated for the various (T.D.C.)^s only during the non-pivoted code sections, i.e., excluding 1st code subsection for the first (T.D.C.), excluding 1st and 2nd code subsections for the second (T.D.C.), etc.

The second step is to compute the probability distribution of the number of transitions counted by one of the (T.D.C.)^s say the K^{th} . This is denoted by $P_{j,K}(i)$, where j is the circuit number, i is the number of (presumably correct) transitions counted by K^{th} (T.D.C.) at the end of the search interval (one code length) and $j = 0, 1$ used to indicate whether this count is correct ($j = 1$) or not correct ($j = 0$). A probability tree for the succession of events leading to a certain count i of circuit number K with probability $P_{j,K}(i)$ is shown in Fig. (5.5). To draw such a probability tree we need to imagine a few scenarios of what could happen to circuit K , recall how the transitions are counted..., etc.

At the end of code subsection K (the pivot of circuit K), the (T.D.C.) will store the number of the code subsection that has been received, consider it legitimate and expect to receive the following code subsection in a logical order e.g. 3 after 2, 4 after 3, ..., etc.

At the end of the K^{th} subsection, the transition count can be 1 or 0. Moreover, there are two ways to get 1, the first due to the right M.F. peaking (with prob. (b) defined before) or the wrong M.F.

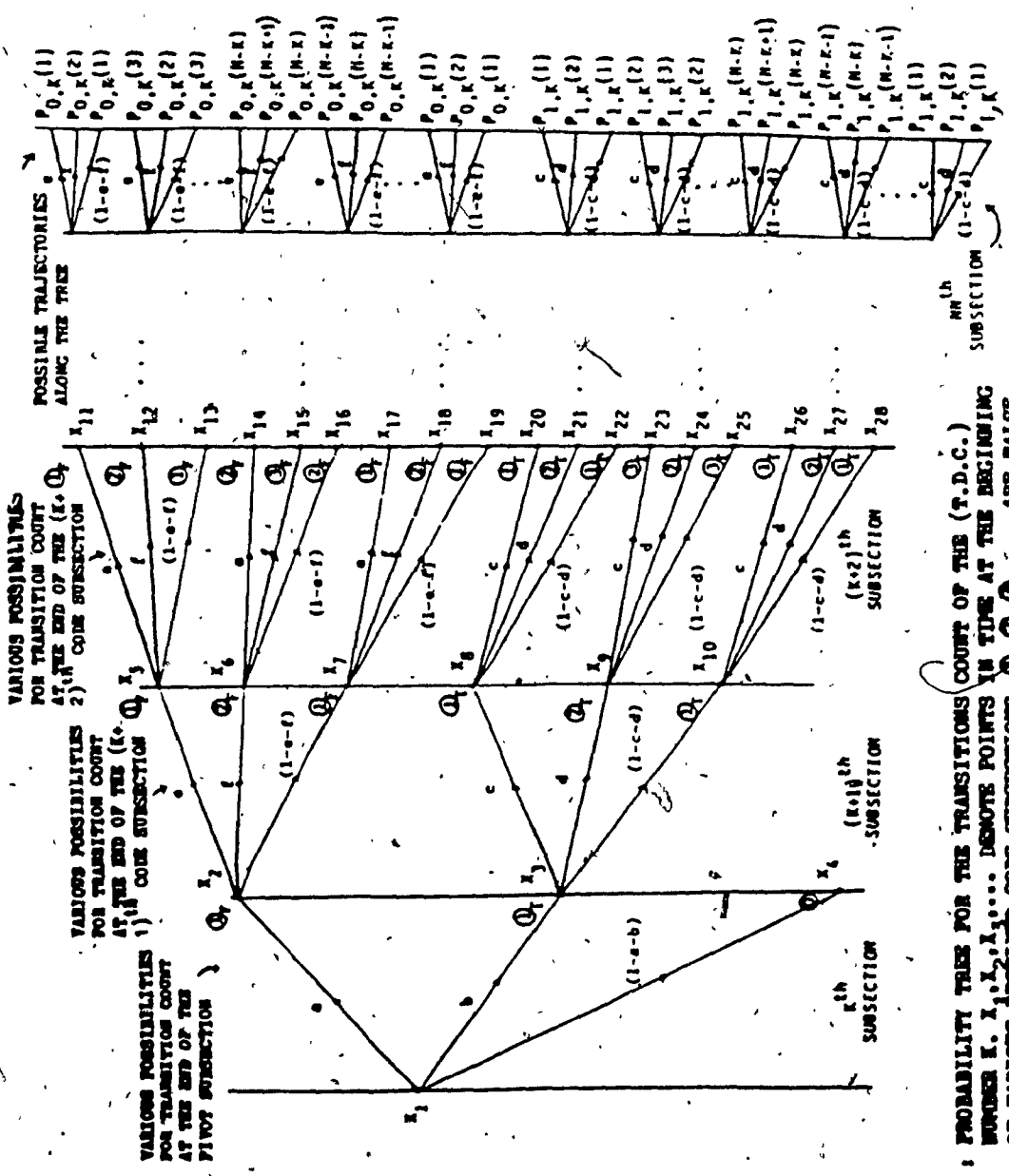


FIG. 5.5 : PROBABILITY TREE FOR THE TRANSITIONS COUNT OF THE (T.D.C.) NUMBER k . X_1, X_2, X_3, \dots DENOTE POINTS IN TIME AT THE BEGINNING OF VARIOUS RECEIVED CODE SUBSECTIONS. Q_1, Q_2, Q_3, \dots ARE FALSE TRANSITION COUNTS (RESULTING FROM FALSE PIVOT CODE SUBSECTION) OBTAINED DUE TO FALSE ALARM; Q_1, Q_2, Q_3, \dots DENOTE CORRECT TRANSITION COUNTS OBTAINED BY CORRECT DETECTION OF THE CORRECT CODE SUBSECTIONS.

peaking (with prob. (a) defined before). The probability of getting 0 transitions is equal to $(1-a-b)$. In Fig. (5.5) we write the accumulated number of transitions (encircled) at the end of each possible transition and note that with probability $(1-a-b)$ giving $P_K(0)$ we do not proceed any further in the tree.

The reason stems from the action of the (T.D.C.) number K which will rest itself (wait for the next search (acquisition trial)) if it finds at the end of the K^{th} pivot subsection that all (M.F.)^S have not peaked (with probability $(1-a-b)$ in Fig. (5.5)). We expect that this part of the acquisition policy gives us the ability to ignore received code subsections heavy in noise or jamming (as indicated by having all (M.F.)^S not peaked). Also, it will be meaningless for circuit K to proceed further after that (in the tree and in action) if this happens since the next subsection is a pivot for the next (T.D.C.) which then takes over at the $(K+1)^{\text{th}}$ subcode.

Back to the tree, the K^{th} (T.D.C.) receives now the ~~next~~ code subsection, if it is in order (with prob. (d)) the transition count is incremented, if it is not in order (with prob. (c)) the count remains the same. If none of the matched filters peak (with prob. $(1-c-d)$) then the count remains the same. The above events taking place during the $(K+1)^{\text{th}}$ subsection are based on having the right pivot during the K^{th} subsection (i.e. start with X_3 on the tree Fig. (5.5)). It is also possible to arrive at the same count even if we start with

the wrong code subsection in the K^{th} period (point X_2 on the tree).

The transition probabilities in this case are given by:

Probability of going from X_2 to $X_5 = e$

Probability of going from X_2 to $X_6 = f$

Probability of going from X_2 to $X_7 = (1-e-f)$

where e, f defined previously.

The difference however, between starting with X_2 or X_3 (Fig. 5.5) will finally show up in having probabilities $P_{1,K}(i)$ for all events branches emanating from X_3 and $P_{0,K}(i)$ for all branches emanating from X_2 . Here we recall that X_2 corresponds to starting with a false pivot code subsection and X_3 corresponds to starting with the true pivot code subsection.

To finalize the description of the tree of Fig. (5.5) we take few examples for possible events and trace them through the tree. In one example, if $K = 3, N = 5$ and the following sequence of numbers is dumped by the encoder 34125. The 3rd (T.D.C.) will ignore 3, 4 and consider 1 as the pivot, receives 2, increment the transition count to 2, receives 5 and ignore it since it does not fall in the right sequence after 1, 2. The corresponding path on the tree is X_1, X_3, X_9, X_{25} , the corresponding transition probability is $bd(1-c-d)$ and the final probability of this combined event is $P_{1,3}(2)$.

The above example assumes that the pivot was correctly detected by (T.D.C.) number 3 i.e. the real transmitted code subsection was 1 in the pivot followed by 2. It is also possible to have (M.F.)₁ peaking due to false alarms while the real status of the code is 4 or 3..., etc. This implies that for the sequence 34125 the corresponding path on the tree is $X_1 X_2 X_6 X_{14}$, the combined transition probability is (afe) and the final probability becomes $P_{0,3}(2)$.

In another example, $K = N = 8$, and the received sequence is 48121345. The 6th (T.D.C.) will ignore 48121, consider 3 as the right pivot, finally count 3. If the pivot was correct, then the path on the tree will be $X_1 X_3 X_9 X_{24}$, the combined transition probability is (bdd) and the final probability is $P_{1,6}(3)$. On the other hand, for the later example, if the pivot was falsely detected then for the same sequence and same circuit, the path on the tree will be $X_1 X_2 X_6 X_{15}$, the transition probability of this combined event is (aff) and the final probability is $P_{0,6}(3)$.

Next, we evaluate the various final probabilities in terms of the transition probabilities. We note that several paths on the tree may lead to the same outcome i.e., same final probability. Summing over all those paths we obtain:
The distribution of the final count of (T.D.C.) number K due to false pivot is, (summing over the appropriate events of Fig. (5.5)).

$$P_{0,K}^{(1)} = a(1-f)^{N-K}$$

$$P_{0,K}^{(2)} = a \binom{N-K}{1} f(1-f)^{N-K-1}$$

$$P_{0,K}^{(3)} = a \binom{N-K}{2} f^2(1-f)^{N-K-2}$$

⋮

$$P_{0,K}^{(N-K+1)} = af^{N-K}$$

(5.9)

While the part of distribution due to correct pivot detection is,

$$P_{1,K}^{(1)} = b(1-d)^{N-K}$$

$$P_{1,K}^{(2)} = b \binom{N-K}{1} d(1-d)^{N-K-1}$$

$$P_{1,K}^{(3)} = b \binom{N-K}{2} d^2(1-d)^{N-K-2}$$

⋮

$$P_{1,K}^{(N-K+1)} = bd^{N-K}$$

(5.10)

Also, for further development, we will need the marginal distribution $P_K(i)$ defined as follows:

$$P_K(0) = (1-a-b)$$

$$P_K(1) = P_{1,K}(1) + P_{0,K}(1)$$

$$P_K(2) = P_{1,K}(2) + P_{0,K}(2)$$

⋮

$$P_K(N-K+1) = P_{1,K}(N-K+1) + P_{0,K}(N-K+1) \quad (5.11)$$

All the aforedefined probabilities so far serve as the building blocks for the three basic probabilities μ_f, μ_d, μ_m to be used in the Markovian state diagram of the acquisition time. These are defined as follows:

μ_f = Probability that although the arriving code at the time the acquisition trial started was not aligned with any of the (M.F.)^s, the combiner finally gave the code phase of one of the (T.D.C.)^s based of course on erroneous decisions by this (T.D.C.) and/or other (T.D.C.) (caused by noise or jamming in the received signal). Of course the given code phase is wrong and subsequent code verification (locks) will fail.

μ_m = Probability that although the arriving code at the time the acquisition trial started was aligned with one of the (M.F.)^s, the combiner finally selected the code phase given by the wrong (T.D.C.)^s, which leads naturally to unsuccessful locks. This may happen due to erroneous transition counts in the part of one

or more of the (T.D.C.)^s caused by noise or jamming in the received code signal.

μ_d = Probability that the arriving code at the time the acquisition trial started was aligned with one of the (M.F.)^s, the combiner finally gave the correct code phase (leading naturally to successful locks (verification)).

The word "aligned" has to be explicitly explained. In classic serial search [1] this is defined as the probability that the arriving code is aligned with the stored replica (active correlation or passive M.F.) chip by chip. So, if the number of such cells is q and the serial algorithm searches one chip per acquisition trial [1], then the probability that the local code is in epoch with the arriving code when the serial search starts is,

$$P_{\text{align}} = \frac{1}{q} \quad (5.12)$$

Moreover, in many cases q = the total code length (in general)

i.e.,

$$q = L^* \quad (5.13)$$

* Unless special alterations are made to the acquisition scheme as in [5] such that the system returns to the most probable cell after loss of synchronization.

Also, P_{align} should naturally improve with the elapse of acquisition trials one after another, i.e.,

$$P_{align} = \frac{1}{(q-\ell+1)} \tag{5.14}$$

where ℓ is the number of the acquisition trial.

In our new parallel search, P_{align} is undoubtedly better than that of serial search since we have N matched filters and so an arriving code which is not aligned with many of those (M.F.)^s will be aligned with one of them in most cases. So, with our parallel scheme the uncertainty region is effectively reduced and,

$$P_{align} = \frac{1}{M} \text{ which is } N \text{ times better than } \frac{1}{L} \tag{5.15}$$

Moreover, P_{align} expectedly improves with each acquisition trial and so,

$$P_{align} = \frac{1}{(M-\ell+1)} \tag{5.16}$$

where ℓ is the number of the transition trial.

We continue the development now by evaluating the probabilities μ_d, μ_m, μ_f . We start with the false alarm probability μ_f . This occurs because of false alarm of one or more of the (T.D.C.)^s i.e.

$$\mu_f = 1 - (1-\mu_{f_1})(1-\mu_{f_2}) \dots (1-\mu_{f_{NN}}) \tag{5.17}$$

where $|f_1, |f_2, \dots, |f_{NN}$ are the false alarm probabilities of different (T.D.C.)^s.

The independence assumption implied by (5.17) of the decisions taken by various (T.D.C.)^s will also prevail in the calculations leading to μ_d, μ_m . This is more or less justified by the differing (M.F.)^s and also because the false alarm, or detection decisions of different (T.D.C.)^s are based on a different offset code subsection (see the delays of Fig. (5.2) and Ref. [1] pp. 73, part III which bears similar meaning). In all, the independence assumption is a convenience tool for the analysis of elaborate algorithms such as that of Fig. (5.2). The false alarm probability μ_{f_k} of circuit K is given by,

$$\mu_{f_k} = \bar{\beta} \sum_{i=TH_K}^{N-K+1} \binom{N-K}{i-1} \beta^{i-1} (1-\beta)^{N-K-i+1} ; \quad 1 < TH_K < (N-K+1) \quad (5.18)$$

where

$$\bar{\beta} = 1 - (1-\beta)^N. \quad (5.19)$$

$\bar{\beta}$ is defined as the probability that during the period of determination of the code pivot section of the K^{th} (T.D.C.), one or more (M.F.)^s will falsely peak (each with a given probability β) while the received code is not aligned. The summation in (5.18) represents the probability that circuit K count of transitions falsely exceeds TH_K i.e. the prefixed threshold of this (T.D.C.). It is to be noted that during the

determination of the pivot, we have up to a total of N (M.F.)^S that can falsely peak and so in this case the false alarm probability is \bar{p} (eq. 5.19) rather than just p .

However, in succeeding transitions the (T.D.C.) counts only those transitions that are in logical order (i.e. 2 following 1, 3 following 2, ..., etc.) i.e. (T.D.C.) waits only for a specific (M.F.) to peak, and the probability of false alarm in this case is for each logical transition count. This does not necessarily lead to correct acquisition due to wrong pivot determination.

The evaluation of the detection probability μ_d (of the whole configuration of all (T.D.C.)^S and the combiner) is rather involved by the many situations that may exist. We write μ_d in terms of the various distributions defined before, then explain those various situations:

$$\mu_d = \sum_{i_1=0}^N \sum_{i_2=0}^{N-1} \dots \sum_{i_{NN}=0}^{N-NN+1} \frac{1}{I} (P_{1,1}(i_1))^{i_1} (P_{2,2}(i_2))^{i_2} (P_{3,3}(i_3))^{i_3} \dots (P_{NN}(i_{NN}))^{i_{NN}}$$

$$i_1 > (i_2, i_3, \dots, i_{NN})$$

$$i_1 > TH_1$$

$$+ \sum_{i_1=0}^N \sum_{i_2=0}^{N-1} \dots \sum_{i_{NN}=0}^{N-NN+1} \frac{1}{i} (P_1(i_1))^1 (P_{1,2}(i_2))^1 (P_3(i_3))^1 \dots (P_{NN}(i_{NN}))^1$$

$$i_2 > (i_1, i_3, \dots, i_{NN})$$

$$i_2 > TH_2$$

$$+ \sum_{i_1=0}^N \sum_{i_2=0}^{N-1} \dots \sum_{i_{NN}=0}^{N-NN+1} \frac{1}{i} (P_1(i_1))^1 (P_2(i_2))^1 (P_{1,3}(i_3))^1 \dots (P_{NN}(i_{NN}))^1$$

$$i_3 > (i_1, i_2, \dots, i_{NN})$$

$$i_3 > TH_3$$

+ . . .

$$+ \sum_{i_1=0}^N \sum_{i_2=0}^{N-1} \dots \sum_{i_{NN}=0}^{N-NN+1} \frac{1}{i} (P_1(i_1))^1 (P_2(i_2))^1 (P_3(i_3))^1 \dots (P_{1,NN}(i_{NN}))^1$$

$$i_{NN} > (i_1, i_2, \dots, i_{NN-1})$$

$$i_{NN} > TH_{NN}$$

$$+ \sum \sum \dots \sum \frac{1}{2} (P_{1,1}(i_1))^1 (P_{0,2}(i_2))^1 (P_3(i_3))^1 \dots (P_{NN}(i_{NN}))^1$$

$$(i_1=i_2) > (i_3, i_4, \dots, i_{NN})$$

$$i_1 > TH_1, i_2 > TH_2$$

$$+ \sum \sum \dots \sum \frac{1}{2} (P_{1,1}(i_1))^1 (P_2(i_2))^1 (P_{0,3}(i_3))^1 \dots (P_{NN}(i_{NN}))^1$$

$$(i_1=i_3) > (i_2, i_4, \dots, i_{NN})$$

$$i_1 > TH_1, i_3 > TH_3$$

+ . . .

$$+ \Sigma \Sigma \dots \Sigma \frac{1}{2} (P_{1,1}(i_1))^1 (P_{2,2}(i_2))^1 (P_{3,3}(i_3))^1 \dots (P_{0,NN}(i_{NN}))^1$$

$$(i_1 = i_{NN}) > (i_2, i_3, \dots, i_{NN-1})$$

$$i_1 > TH_1, i_{NN} > TH_{NN}$$

$$+ \Sigma \Sigma \dots \Sigma \frac{1}{2} (P_1(i_1))^1 (P_{1,2}(i_2))^1 (P_{0,3}(i_3))^1 (P_4(i_4))^1 \dots (P_{NN}(i_{NN}))^1$$

$$(i_2 = i_3) > (i_1, i_4, \dots, i_{NN})$$

$$i_2 > TH_2, i_3 > TH_4$$

+ . . .

$$+ \Sigma \Sigma \dots \Sigma \frac{1}{2} (P_1(i_1))^1 (P_{1,2}(i_2))^1 (P_3(i_3))^1 (P_4(i_4))^1 \dots P_{0,NN}(i_{NN}))^1$$

$$(i_2 = i_{NN}) > (i_1, i_3, \dots, i_{NN-1})$$

$$i_2 > TH_2, i_{NN} > TH_{NN}$$

+ . . .

$$+ \Sigma \Sigma \dots \Sigma \frac{2}{2} (P_{1,1}(i_1))^1 (P_{1,2}(i_2))^1 (P_3(i_3))^1 \dots (P_{NN}(i_{NN}))^1$$

$$(i_1 = i_2) > (i_3, i_4, \dots, i_{NN})$$

$$i_1 > TH_1, i_2 > TH_2$$

$$+ \Sigma \Sigma \dots \Sigma \frac{2}{2} (P_{1,1}(i_1))^1 (P_2(i_2))^1 (P_{1,3}(i_3))^1 (P_4(i_4))^1 \dots (P_{NN}(i_{NN}))^1$$

$$(i_1 = i_3) > (i_2, i_4, \dots, i_{NN})$$

$$i_1 > TH_1, i_3 > TH_3$$

+ . . .

$$+ \Sigma \Sigma \dots \Sigma \frac{2}{2} (P_{1,1}(i_1))^1 (P_{2,2}(i_2))^1 (P_{3,3}(i_3))^1 (P_{4,4}(i_4))^1 \dots (P_{1,NN}(i_{NN}))^1$$

$$(i_1 = i_{NN}) > (i_2, i_3, i_4, \dots, i_{NN})$$

$$i_1 > TH_1, i_{NN} > TH_{NN}$$

$$+ \Sigma \Sigma \dots \Sigma \frac{2}{2} (P_1(i_1))^1 (P_{1,2}(i_2))^1 (P_{1,3}(i_3))^1 \dots (P_{NN}(i_{NN}))^1$$

$$(i_2 = i_3) > (i_1, i_4, \dots, i_{NN})$$

$$i_2 > TH_2, i_3 > TH_3$$

+ . . .

$$+ \Sigma \Sigma \dots \Sigma \frac{2}{2} (P_1(i_1))^1 (P_{1,2}(i_2))^1 (P_3(i_3))^1 (P_4(i_4))^1 \dots (P_{1,NN}(i_{NN}))^1$$

$$(i_2 = i_{NN}) > (i_1, i_3, \dots, i_{NN})$$

$$i_2 > TH_2, i_{NN} > TH_{NN}$$

$$+ \Sigma \Sigma \dots \Sigma \frac{2}{2} (P_1(i_1))^1 (P_2(i_2))^1 (P_{1,3}(i_3))^1 (P_{1,4}(i_4))^1 \dots (P_{NN}(i_{NN}))^1$$

$$(i_3 = i_4) > (i_1, i_2, i_5, \dots, i_{NN})$$

$$i_3 > TH_3, i_4 > TH_4$$

+ . . .

$$+ \Sigma \Sigma \dots \Sigma \frac{2}{2} (P_1(i_1))^1 (P_2(i_2))^1 (P_{1,3}(i_3))^1 (P_4(i_4))^1 \dots (P_{1,NN}(i_{NN}))^1$$

$$(i_3 = i_{NN}) > (i_1, i_2, i_4, \dots, i_{NN-1})$$

$$i_3 > TH_3, i_{NN} > TH_{NN}$$

+ . . .

$$+ \Sigma \Sigma \dots \Sigma \frac{1}{3} (P_{1,1}(i_1))^1 (P_{0,2}(i_2))^1 (P_{0,3}(i_3))^1 (P_4(i_4))^1 \dots (P_{NN}(i_{NN}))^1$$

$$i_1 > TH_1, i_2 > TH_2, i_3 > TH_3$$

$$(i_1 - i_2 - i_3) > (i_4, i_5, \dots, i_{NN})$$

+

$$+ \Sigma \Sigma \dots \Sigma \frac{1}{3} (P_{1,1}(i_1))^1 (P_{0,2}(i_2))^1 (P_3(i_3))^1 \dots (P_{0,NN}(i_{NN}))^1$$

$$i_1 > TH_1, i_2 > TH_2, i_{NN} > TH_{NN}$$

$$(i_1 - i_2 - i_{NN}) > (i_3, i_4, \dots, i_{NN-1})$$

$$+ \Sigma \Sigma \dots \Sigma \frac{1}{3} (P_1(i_1))^1 (P_{1,2}(i_2))^1 (P_{0,3}(i_3))^1 (P_4(i_4))^1 \dots (P_{0,NN}(i_{NN}))^1$$

$$i_2 > TH_2, i_3 > TH_3, i_{NN} > TH_{NN}$$

$$(i_2 - i_3 - i_{NN}) > (i_1, i_4, \dots, i_{NN-1})$$

+

$$+ \Sigma \Sigma \dots \Sigma \frac{2}{3} (P_{1,1}(i_1))^1 (P_{1,2}(i_2))^1 (P_{0,3}(i_3))^1 (P_4(i_4))^1 \dots (P_{NN}(i_{NN}))^1$$

$$i_1 > TH_1, i_2 > TH_2, i_3 > TH_3$$

$$(i_1 - i_2 - i_3) > (i_4, i_5, \dots, i_{NN})$$

+

$$+ \Sigma \Sigma \dots \Sigma \frac{2}{3} (P_{1,1}(i_1))^1 (P_{1,2}(i_2))^1 (P_3(i_3))^1 (P_4(i_4))^1 \dots (P_{0,NN}(i_{NN}))^1$$

$$i_1 > TH_1, i_2 > TH_2, i_{NN} > TH_{NN}$$

$$(i_1 - i_2 - i_{NN}) > (i_3, i_4, \dots, i_{NN-1})$$

+

$$+ \Sigma \Sigma \dots \Sigma \frac{3}{3} (P_{1,1}(i_1))^1 (P_{1,2}(i_2))^1 (P_{1,3}(i_3))^1 (P_4(i_4))^1 \dots (P_{NN}(i_{NN}))^1$$

$$i_1 > TH_1, i_2 > TH_2, i_3 > TH_3$$

$$(i_1=i_2=i_3) > (i_4, i_5, i_6, \dots, i_{NN})$$

+ . . .

$$+ \Sigma \Sigma \dots \Sigma \frac{3}{3} (P_{1,1}(i_1))^1 (P_{1,2}(i_2))^1 (P_{1,3}(i_3))^1 (P_{1,4}(i_4))^1 \dots (P_{1,NN}(i_{NN}))^1$$

$$i_2 > TH_2, i_3 > TH_3, i_{NN} > TH_{NN}$$

$$(i_2=i_3=i_{NN}) > (i_1, i_4, i_5, \dots, i_{NN-1})$$

+ . . .

$$+ \Sigma \Sigma \dots \Sigma \frac{1}{4} (P_{1,1}(i_1))^1 (P_{0,2}(i_2))^1 (P_{0,3}(i_3))^1 (P_{0,4}(i_4))^1 \dots (P_{NN}(i_{NN}))^1$$

$$i_1 > TH_1, i_2 > TH_2, i_3 > TH_3, i_4 > TH_4$$

$$(i_1=i_2=i_3=i_4) > (i_5, i_6, \dots, i_{NN})$$

+ . . .

$$+ \Sigma \Sigma \dots \Sigma \frac{2}{4} (P_{1,1}(i_1))^1 (P_{1,2}(i_2))^1 (P_{0,3}(i_3))^1 (P_{0,4}(i_4))^1 \dots (P_{NN}(i_{NN}))^1$$

$$(i_1=i_2=i_3=i_4) > (i_5, i_6, \dots, i_{NN})$$

$$i_1 > TH_1, i_2 > TH_2, i_3 > TH_3, i_4 > TH_4$$

+ . . .

+ . . .

$$+ \Sigma \Sigma \dots \Sigma \frac{1}{NN} (P_{1,1}(i_1))^1 (P_{0,2}(i_2))^1 (P_{0,3}(i_3))^1 (P_{0,4}(i_4))^1 \dots (P_{0,NN}(i_{NN}))^1$$

$$i_1=i_2=i_3=\dots=i_N$$

$$i_1 > TH_1, i_2 > TH_2, i_3 > TH_3, \dots, i_N > TH_N$$

+ . . .

$$+ \sum \sum \dots \sum \frac{2}{NN} (P_{1,1}(i_1))^1 (P_{1,2}(i_2))^1 (P_{0,3}(i_3))^1 (P_{0,4}(i_4))^1 \dots (P_{0,NN}(i_{NN}))^1$$

$$i_1 = i_2 = i_3 = \dots = i_N$$

$$i_1 > TH_1, i_2 > TH_2, \dots, i_N > TH_N$$

+

$$+ \sum \sum \dots \sum \frac{3}{NN} (P_{1,1}(i_1))^1 (P_{1,2}(i_2))^1 (P_{1,3}(i_3))^1 (P_{0,4}(i_4))^1 \dots (P_{0,NN}(i_{NN}))^1$$

$$i_1 = i_2 = i_3 = \dots = i_{NN}$$

$$i_1 > TH_1, i_2 > TH_2, \dots, i_{NN} > TH_{NN}$$

+

+

$$+ \sum \sum \dots \sum \frac{NN}{NN} (P_{1,1}(i_1))^1 (P_{1,2}(i_2))^1 (P_{1,3}(i_3))^1 (P_{1,4}(i_4))^1 \dots (P_{1,NN}(i_{NN}))^1$$

$$i_1 = i_2 = i_3 = \dots = i_{NN}$$

$$i_1 > TH_1, i_2 > TH_2, \dots, i_{NN} > TH_{NN}$$

(5.20)

The first NN fold summation of (5.20) represents all these situations of transition counts $(i_1, i_2, i_3, \dots, i_{NN})$ of the NN (T.D.C.)^S where the first (T.D.C.) count exceeds the counts of all the other circuits $(i_2, i_3, \dots, i_{NN})$ and the count of first (T.D.C.) has exceeded its prefixed threshold $(i_1 > TH_1)$. We note that the subscript associated with

$P_{1,1}(i_1)$ indicates a correct detection (see the definitions and equations (5.9), (5.10), (5.11) while those of i_2, i_3, \dots, i_{NN} have only one subscript. The events leading to all probabilities relevant to this NN fold summation contributes to the probability of correct detection (P_d).

The following 2nd and 3rd ... and the NNth summations are similar to the first except that the 2nd or 3rd ... etc. (T.D.C.) count now exceeds the rest and its preassigned threshold. Again we note the subscripts which imply a correct detection of one (T.D.C.). It is also possible for a tie to exist between a correct (T.D.C.) and a false alarm count of another ($i_1 = i_2$) and both counts have exceeded their thresholds and they are the maximum of the list of counts according to Fig. (5.2). In this case the right (T.D.C.) will be picked randomly* with probability $\frac{1}{2}$. (Please see line 5 of (5.20) and note the subscript $P_{1,1}(i_1)$ indicating a correct (T.D.C.) count and $P_{0,2}(i_2)$ indicating a false count). The following NN summations of (5.20) all have the same form except the identity of the maximum count circuits (i_1, i_3) or (i_1, i_4), ..., (i_2, i_3), (i_2, i_4) ... etc. yielding correct detection.

However, a tie may exist between 2 or more correct (T.D.C.) circuits in which case the combiner of Fig. (5.2) will pick any one. They all are correct, so the final decision regarding the code phase is always correct, meaning the factor $\frac{1}{2}$ or $\frac{1}{3}$... etc. will disappear from the NN fold summations and we will have instead $\frac{2}{2}$ or $\frac{3}{3}$... etc. The reader is urged to pick those cases of equation (5.20) and note that the subscripts of the circuits indicate correct (T.D.C.) circuits.

*Fig. (5.2) implies that the highest count (T.D.C.) exceeding its threshold is always (not randomly) selected but in the long run, the reordering of (T.D.C.) count (to become $I(j)$) will scramble the circuits resulting in the same effect on random selection of tied circuits.

$$\mu_d = \sum_{i_1=0}^N \sum_{i_2=0}^{N-1} \dots \sum_{i_{NN}=0}^{N-NN+1} \left[\begin{array}{cccccc} 1 & 1 & 1 & 1 & 1 & 1 \\ \Sigma & \Sigma & \Sigma & \Sigma & \Sigma & \Sigma & \dots \\ j_1=0 & k_1=0 & l_1=0 & j_2=0 & k_2=0 & l_2=0 & \dots \end{array} \right]$$

$$\sum_{j_{NN}=0}^1 \sum_{k_{NN}=0}^1 \sum_{l_{NN}=0}^1 \frac{j_1 + j_2 + \dots + j_{NN}}{j_1 + j_2 + \dots + j_{NN} + k_1 + k_2 + \dots + k_{NN}}$$

$$\cdot (P_{1,1}(i_1))^{j_1} (P_{0,1}(i_1))^{k_1} (P_1(i_1))^{l_1} (P_{1,2}(i_2))^{j_2} (P_{0,2}(i_2))^{k_2} (P_2(i_2))^{l_2} \dots$$

$$(P_{1,NN}(i_{NN}))^{j_{NN}} (P_{0,NN}(i_{NN}))^{k_{NN}} (P_{NN}(i_{NN}))^{l_{NN}} \quad (5.21)$$

Subject to the conditions,

$$j_1 + k_1 + l_1 = j_2 + k_2 + l_2 = \dots = j_{NN} + k_{NN} + l_{NN} = 1$$

where $j_1 = 1$ (say) indicates a correct count on first (T.D.C.) and automatically implies $k_1 = l_1 = 0$

$k_4 = 1$ (say) indicates a false count on the 4th (T.D.C.) and automatically implies $j_4 = l_4 = 0$

$l_3 = 1$ (say) indicates a false alarm or true detection on circuit number 3.

Also, if any of the j 's or the k 's is equal to 1 the corresponding i 's must be greater than the thresholds and equal to each other. For example, if $j_1 = j_2 = k_3 = 1$ then $i_1 > TH_1, i_2 > TH_2, i_3 > TH_3$ and $i_1 = i_2 = i_3$.

μ_m is derived similarly as μ_d but the ratio in each of the multiple sums of (5.20) is changed (since the false (T.D.C.) is picked in this case)

so that the general equation for μ_m becomes

$$\mu_m = \begin{matrix} N & N-1 & & N-NN+1 \\ \Sigma & \Sigma & \dots & \Sigma \\ i_1=0 & i_2=0 & & i_{NN}=0 \end{matrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ \Sigma & \Sigma & \Sigma & \Sigma & \Sigma & \Sigma \\ j_1=0 & k_1=0 & l_1=0 & j_2=0 & k_2=0 & l_2=0 \end{bmatrix} \dots$$

$$\begin{matrix} 1 & 1 & 1 \\ \Sigma & \Sigma & \Sigma \\ j_{NN}=0 & k_{NN}=0 & l_{NN}=0 \end{matrix} \frac{k_1+k_2+\dots+k_{NN}}{j_1+j_2+\dots+j_{NN}+k_1+k_2+\dots+k_{NN}}$$

$$(P_{1,1}(i_1))^{j_1} (P_{0,1}(i_1))^{k_1} (P_1(i_1))^{l_1} (P_{i,2}(i_2))^{j_2} (P_{0,2}(i_2))^{k_2} (P_2(i_2))^{l_2} \dots$$

$$\left. \begin{matrix} P_{1,NN}(i_{NN})^{j_{NN}} (P_{0,NN}(i_{NN}))^{k_{NN}} (P_{NN}(i_{NN}))^{l_{NN}} \end{matrix} \right] \quad (5.22)$$

subject to the same conditions as in (5.21).

Having defined μ_d , μ_m , μ_f or the overall scheme, the Markovian state diagram representing the acquisition time is shown in Fig. (5.6). Point 1 on Fig. (5.6) corresponds to the very beginning of parallel acquisition search. At this point, the alignment probability is $P_1 = 1/M$. The reader is referred to [7] for the general description of such flow diagrams. However, we only outline here the differences and the distinct feature of our diagram (Fig. (5.6)) which reflect the new sequential parallel technique.

1. The system moves from state 1 to 2 to 3 ... and finally to M, thus increasing P_{align} from $\frac{1}{M}$ to $\frac{1}{M-1}$ to $\frac{1}{M-2}$ until it becomes finally 1.
2. In each state, the probability of reaching the final state (code acquired) is μ_d (if the code is aligned) and the time spent is one code length z .
3. μ_m represents the detection of the wrong code phase (different from false alarm and unique to our scheme), however if it happens, un-

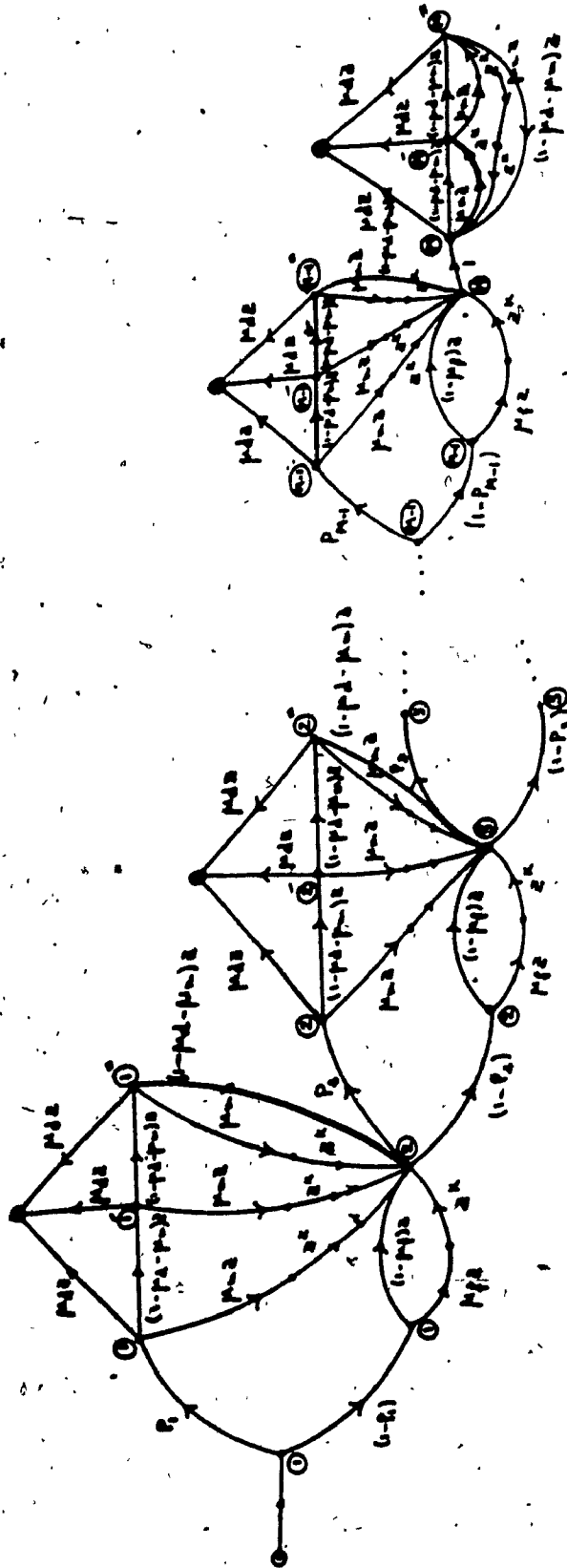


FIG 5.6 : MARKOVIAN STATE DIAGRAM FOR THE ACQUISITION TIME

S : START OF ACQUISITION

P : FINAL STATE (CODE ACQUIRED)

P_{11} : DETECTION PROBABILITY OF OVERALL SCHEME (T.D.C.)⁰ AND COMBINER

P_{12} : WRONG PHASE DETECTION PROBABILITY OF OVERALL SCHEME (T.D.C.)⁰ AND COMBINER

P_{13} : FALSE ALARM PROBABILITY OF OVERALL SCHEME (T.D.C.)⁰ AND COMBINER

P_{14} : UNIT OF DELAY = ONE CODELENGTH (L BITS)

P_{15} : DETECT ACQUISITION TRIAL NUMBER

P_1, P_2, P_3, \dots REPRESENT ALIGNMENT PROBABILITY CORRESPONDING TO VARIOUS ACQUISITION TRIALS

K : FALSE TRACKING PENALTY FACTOR, TYPICALLY $K = 1$

successful tracking will take place and so an extra penalty of z^k is encountered during code verification. Following this unsuccessful verification we go to the next acquisition trial, but naturally assume that the status of incoming code is now close to the stored M.F. so we move on the diagram from 1 to 2 (better code alignment).

4. It is possible (with probability $(1 - \mu_d - \mu_m)$) to have no code phase whatsoever at the end of the acquisition trial (neither right nor wrong, for example, if none of the pivot (M.F.)^s exceeded the threshold (see Fig. (5.2)). In this event we move from 1 to 1' on Fig. (5.6) and if the same happens again (with a very small probability) in the next trial period (one code length) few chips of the incoming code are skipped (as in Fig. (5.2)) which will guarantee the system moving from 1' to a better alignment possibility (Point 2).
5. The alignment probabilities P_1, P_2, \dots , have no penalty or time loss (z) associated with them.
6. The system eventually moves to state (M-1) then state M where $P_M = 1$ and remain there until acquisition is obtained (moving to F). However, few oscillations might exist (as expressed by the branches $(1 - \mu_d - \mu_m)z$ and $\mu_m z^{k+1}$) which may prolong final acquisition of the code.

The moment generating function for the acquisition time ($U(z)$) is now defined (from the flow of Fig. (5.6)) as,

$$\begin{aligned}
 U(z) &= \frac{f}{s} = A_1(z) + H_1(z)A_2(z) + H_1(z)H_2(z)A_3(z) + \dots + H_1(z)H_2(z)\dots H_{M-1}(z)A_M(z) \\
 &= \sum_{i=1}^M A_i(z) \prod_{\substack{j=1 \\ i \neq j}}^{(i-1)} H_j(z)
 \end{aligned} \tag{5.23}$$

where

$$A_i(z) = P_i [1 + (1-\mu_d-\mu_m)z + (1-\mu_d-\mu_m)^2 z^2] \mu_d z \quad (5.24)$$

$$A_M(z) = \frac{1 + [(1-\mu_d-\mu_m)z + \mu_m z^{k+1}] + [(1-\mu_d-\mu_m)z + \mu_m z^{k+1}]^2 \mu_d}{1 - [(1-\mu_d-\mu_m)z + \mu_m z^{k+1}]^3} \quad (5.25)$$

$$B_i(z) = P_i [1 + (1-\mu_d-\mu_m)z + (1-\mu_d-\mu_m)^2 z^2] \mu_m z^{k+1} + (1-\mu_d-\mu_m)^3 z^3 \quad (5.26)$$

$$C_i(z) = (1-P_i) [\mu_f z^{k+1} + (1-\mu_f)z] \quad (5.27)$$

$$H_i(z) = B_i(z) + C_i(z) \quad (5.28)$$

where in (5.23), (5.24), (5.26), (5.27), (5.28); $i = 1, 2, \dots, (M-1)$.

Now $U(z)$ is redefined as (to facilitate finding the derivatives with respect to z)

$$U(z) = A_1(z) + G(z) + G_M(z) \quad (5.29)$$

where

$$G(z) = \sum_{i=2}^{(M-1)} [A_i(z) \prod_{j=1}^{(i-1)} H_j(z)] \quad (5.30)$$

$$G_M(z) = A_M(z) \prod_{j=1}^{(M-1)} H_j(z) \quad (5.31)$$

$$A_1(z) = P_1 [1 + (1-\mu_d-\mu_m)z + (1-\mu_d-\mu_m)^2 z^2] \mu_d z \quad (5.32)$$

The mean acquisition time is defined as,

$$\left. \frac{dU(z^{\tau_d})}{dz} \right|_{z=1} = \left. \frac{dG(z^{\tau_d})}{dz} \right|_{z=1} + \left. \frac{dG_M(z^{\tau_d})}{dz} \right|_{z=1} + \left. \frac{dA_1(z^{\tau_d})}{dz} \right|_{z=1} \quad (5.33)$$

The evaluation of the moment generating function $U(z)$ obtained by classic flow graph techniques and its derivatives is a simple (though tedious) exercise. The result is,

$$\begin{aligned} \bar{T} &= \left. \frac{dU(z^{\tau_d})}{dz} \right|_{z=1} = \mu_d P_1 \tau_d (1 + 2\theta + 3\theta^2) \\ &+ \tau_d \left\{ \sum_{i=2}^{(M-1)} \mu_d P_i [(1 + 2\theta + 3\theta^2)] \prod_{j=1}^{(i-1)} \psi_j \right. \\ &\quad \left. + (1 + \theta + \theta^2) \sum_{k=1}^{(i-1)} \gamma_k \prod_{\substack{l=1 \\ l \neq k}}^{(i-1)} \psi_l \right\} \\ &+ \left[\begin{array}{l} \xi \prod_{j=1}^{(M-1)} \psi_j + \sum_{k=1}^{(M-1)} \gamma_k \prod_{\substack{l=1 \\ l \neq k}}^{(M-1)} \psi_l \end{array} \right] \end{aligned} \quad (5.34)$$

where

$$\psi_i = \left. (H_i(z^{\tau_d})) \right|_{z=1/\tau_d} = 1 - P_i [1 - \mu_m (1 + \theta + \theta^2) - \theta^3] \quad (5.35)$$

$i = 1, 2, \dots, M-1$

where τ_d is the code length (L chips)

$$\begin{aligned} \gamma_j &= \left. \frac{dH_j(z^{\tau_d})}{dz} \right|_{z=1/\tau_d} \\ &= P_j [\mu_m [(k+1) + (k+2)\theta + (k+3)\theta^2] + 3\theta^3] + (1 - P_j)(1 + k\mu_m) \\ &\quad j = 1, 2, \dots, M-1 \end{aligned} \quad (5.36)$$

$$\xi = \left. \frac{dA_H(z^{\tau_d})}{dz} \right|_{z=1/\tau_d} = \frac{(1 + k\mu_m)}{\mu_d} \quad (5.37)$$

$$\theta = 1 - \mu_d - \mu_m \quad (5.38)$$

The variance of the acquisition time is defined as,

$$\sigma_T^2 = \left[\frac{d^2 U(z^{\tau_d})}{dz^2} + \frac{dU(z^{\tau_d})}{dz} - \left(\frac{dU(z^{\tau_d})}{dz} \right)^2 \right]_{z=1} \quad (5.39)$$

where $U(z)$ has been defined in (5.33). This time the evaluation of the derivatives fills few pages (though straightforward) and the result is,

$$\sigma_T^2 = \frac{d^2 U(z^{\tau_d})}{dz^2} \Big|_{z=1} + \frac{dU(z^{\tau_d})}{dz} \Big|_{z=1} - \left(\frac{dU(z^{\tau_d})}{dz} \Big|_{z=1} \right)^2 \quad (5.40)$$

where $\frac{dU(z^{\tau_d})}{dz^2} \Big|_{z=1}$ is given in (5.33) and (5.34)

$$\text{and } \frac{d^2 U(z^{\tau_d})}{dz^2} \Big|_{z=1} = \frac{d^2 A_1(z^{\tau_d})}{dz^2} \Big|_{z=1} + \frac{d^2 G(z^{\tau_d})}{dz^2} \Big|_{z=1} + \frac{d^2 G_M(z^{\tau_d})}{dz^2} \Big|_{z=1} \quad (5.41)$$

$$\frac{d^2 A_1(z^{\tau_d})}{dz^2} \Big|_{z=1}, \quad \frac{d^2 G(z^{\tau_d})}{dz^2} \Big|_{z=1} \quad \text{and} \quad \frac{d^2 G_M(z^{\tau_d})}{dz^2} \Big|_{z=1}$$

are given as follows:

$$\frac{d^2 A_1(z^{\tau_d})}{dz^2} \Big|_{z=1} = P_1 \mu_d^{\tau_d} [(\tau_d - 1) + 2(2\tau_d - 1)\theta + 3(3\tau_d - 1)\theta^2] \quad (5.42)$$

$$\left. \frac{d^2 G(z^{\tau_d})}{dz^2} \right|_{z=1} = \tau_d \mu_d \sum_{i=2}^{(M-1)} P_i \left\{ \left[(\tau_d - 1) + 2(2\tau_d - 1)\theta \right. \right.$$

$$\left. \left. + 3(3\tau_d - 1)\theta^2 \right] \prod_{j=1}^{(i-1)} \psi_j + \tau_d (1 + 2\theta + 3\theta^2) \right.$$

$$\left. \left. \sum_{k=1}^{(i-1)} \gamma_k \prod_{\substack{l=1 \\ l \neq k}}^{(i-1)} \psi_l \right] + \sum_{j=1}^{(i-1)} \left[\tau_d (1 + 2\theta + 3\theta^2) \gamma_j \right. \right.$$

$$\left. \left. + (1 + \theta + \theta^2) \phi_j \right] \prod_{\substack{k=1 \\ k \neq j}}^{(i-1)} \psi_k + \tau_d (1 + \theta + \theta^2) \gamma_j \right.$$

$$\left. \left. \sum_{\substack{k=1 \\ k \neq j}}^{(i-1)} \gamma_k \prod_{\substack{l=1 \\ l \neq k \\ l \neq j}}^{(i-1)} \psi_l \right] \right\}$$

(5.43)

$$\left. \frac{d^2 G_M(z^{\tau_d})}{dz^2} \right|_{z=1} = \tau_d \left\{ \eta \sum_{j=1}^{(M-1)} \psi_j + \tau_d \xi \sum_{j=1}^{(M-1)} \gamma_j \prod_{\substack{k=1 \\ k \neq j}}^{(M-1)} \psi_k \right.$$

$$\left. \left. + \sum_{j=1}^{(M-1)} \left[(\tau_d \xi \gamma_j + \phi_j) \prod_{\substack{k=1 \\ k \neq j}}^{(M-1)} \psi_k \right. \right. \right.$$

$$\left. \left. \left. + \tau_d \gamma_j \sum_{\substack{k=1 \\ k \neq j}}^{(M-1)} \gamma_k \prod_{\substack{l=1 \\ l \neq k \\ l \neq j}}^{(M-1)} \psi_l \right] \right\}$$

(5.44)

where ψ_j , γ_k , ξ , and θ are given in (5.35 - 5.38) and

$$\psi_j = \frac{d^2 H_j(z^{\tau_d})}{dz^2} \Big|_{z=1} / \tau_d$$

$$= P_j [[(k+1)((k+1)\tau_d - 1) + (k+2)((k+2)\tau_d - 1)\theta + (k+3)((k+3)\tau_d - 1)\theta^2] \mu_m + 3(3\tau_d - 1)\theta^3]$$

$$+ (1 - P_j) [(k+1)((k+1)\tau_d - 1)\mu_f + (\tau_d - 1)(1 - \mu_f)]$$

$$j = 1, 2, \dots, M-1 \tag{5.45}$$

$$\eta = \frac{d^2 A_M(z^{\tau_d})}{dz^2} \Big|_{z=1} / \tau_d$$

$$= \left[\frac{2(1+k\mu_m)^2}{\mu_d^2} - \frac{(1-k^2\mu_m)}{\mu_d} \right] \tau_d - \frac{1+k\mu_m}{\mu_d} \tag{5.46}$$

5.4 Results and conclusions

The mean and standard deviation of the acquisition time were computed from equations (5.34), (5.40) respectively using the Cyber 835. Varying the value of L, M, P_b, P'_b, NN , and TH resulted in Figures (5.7) - (5.12) and Tables 5.1 and 5.3.

Figs. (5.7) and (5.8) show the mean acquisition time (in units of code lengths) versus P'_b and P_b , respectively for $TH_K = (N-K+1)$ and $TH_K = (N-K+1)-2$ ($L = 256, N = 8, NN = 5, TH = 10M/16$). We see that in both figures for low values of P_b and P'_b the mean acquisition time remains unchanged or increases slightly with P_b and P'_b . However, with higher values

of P_b and P'_b as implicated for example by heavy jammings the mean acquisition time increases sharply. In Fig. (5.7) for low value of P_b ($P_b = 0.01, 0.1$) indicating high detection probability, and higher thresholds ($TH_K = (N-K+1)$) the performance is better than those cases with lower thresholds $TH_K = (N-K+1)-2$. However though, at higher values of P_b ($P_b = 0.2, 0.3$) the former case is better than the latter case only below a certain value of P'_b beyond which the situation reverses. The latter effect will be more pronounced at values of $P_b \leq 0.4$.

In Fig. (5.8) at values of $P'_b \leq 0.4$ the performance is better with $TH_K = (N-K+1)-2$. For the values of $P'_b > 0.4$ the case of $TH_K = (N-K+1)$ is better than the other only below a certain value of P_b ($P_b \leq 0.3$ at $P'_b = 0.5$). The above results seem to suggest that the performance is improved in severe jammings (as reflected by the high values of P_b and P'_b), when the threshold $TH_{(K)}$ is decreased.

Fig. (5.9) portrays the effect of MM on the mean acquisition time for varying probabilities of bit miss and false alarm, P_b and P'_b , where MM is defined from the expression $TH_K = (N-K+1)-MM$ as the tolerable number of transition for the K^{th} detecting logic. MM has the values ranging from zero to $(N-K+1)$. At high values of P_b and P'_b (heavy jammings) the performance is improved as MM increases (TH_K decreases). There exists an optimal MM at $P_b = 0.3, 0.4$ in Fig. (5.9a)

and at $P'_b \neq 0.5$ in Fig. (5.9b), and this optimal MM clearly moves to the right in the case of Fig. (5.9a) and to the left in that of Fig. (5.9b) when P_b and P'_b are increased respectively.

The effect of varying the code length (L) and the number of bits per TAD (M) is depicted in Fig. (5.10). The mean acquisition time is plotted versus P_b and P'_b for $L = 256$ ($M = 32$) and $L = 128$ ($M = 16$) (N is kept constant and $TH_K = (N-K+1)$). It is clearly seen that for lower L (and M) the result is better, as we can predict. The curves for the case of $TH_K = (N-K+1)2$ were also plotted. These curves are similar to those of the case $TH_K = (N-K+1)$, $L = 128$ ($M = 16$) and $L = 256$ ($M = 32$).

In Fig. (5.11) the number (NN) of (T.D.C.)^S was varied for $TH_K = (N-K+1)-2$ and we note the existence of an optimal NN. This optimum moves to the right as P_b increases (at $P'_b = 0.5$, Fig. (5.11a) and moves to the left as P'_b increases (at $P_b = 0.4$, Fig. (5.11b)). Naturally, as P_b and P'_b increases, the mean acquisition time increases.

The effect of TH follows from Fig. (5.12), in which the curves for the mean acquisition time were plotted versus TH at $P_b = 0.4$ and $P'_b = 0.5$ for $TH_K = (N-K+1)-2$. There also exists an optimal TH at $P_b \neq 0.4$ (Fig. (5.12a)) and $P'_b \neq 0.5$ (Fig. (5.12b)). The optimal TH seems to be fixed at $TH = 10M/16$ as P_b and P'_b increase within their above respective range.

Tables 5.1 and 5.2 show the mean acquisition time, the corresponding standard deviation (both in unit of chips), the (M.F.)^s, probabilities of detection (α) and false alarm (β), the overall probabilities of correct detection (μ_d), false alarm (μ_f) and wrong phase detection (μ_m) for varying values of P_b and P'_b for $L = 256$ ($M = 32$) and $L = 128$ ($M = 16$). In Table 5.1, P'_b is set equal to 0.5 and P_b is varied. As we can predict for both, $L = 128$ and 256, α decreases as P_b increases, β does not change since P'_b is kept constant, μ_d decreases, μ_m increases and μ_f is unchanged. The mean acquisition time changes slightly for $P_b \leq 0.4$ and increases sharply for higher value of P_b ; so does the standard deviation, which has value roughly half of the value of the mean acquisition time.

In Table 5.2, P_b is kept constant at 0.3 and P'_b is varied. It is noticed that α remains unchanged and as P'_b increases β , μ_m and μ_f increases, μ_d decreases. The mean acquisition time and the standard deviation increase most obviously at $P'_b > 0.4$. If we compare the results of the two cases ($L = 128$ and $L = 256$) we see that in Table 5.1 ($P'_b = 0.5$, P_b is varied), β in the case of $L = 128$ is greater than that of $L = 256$. Also, α for $L = 128$ is less than α for $L = 256$ where $P_b \leq 0.3$, for higher P_b the situation is reversed. The final outcome of the comparison between the cases of $L = 256$, $L = 128$ reveals that μ_d is less and μ_f , μ_m are greater in the case of $L = 128$ compared to those of $L = 256$. Amazingly enough, for $L = 128$, the mean acquisition time and the standard deviation is roughly one fourth of those in the

case of $L = 256$ due to decreasing the code length and the corresponding increase in P_{align} (other parameters being the same).

Table 5.2 supports the conclusions drawn from Table 5.1 in the case of $P_b = 0.3$.

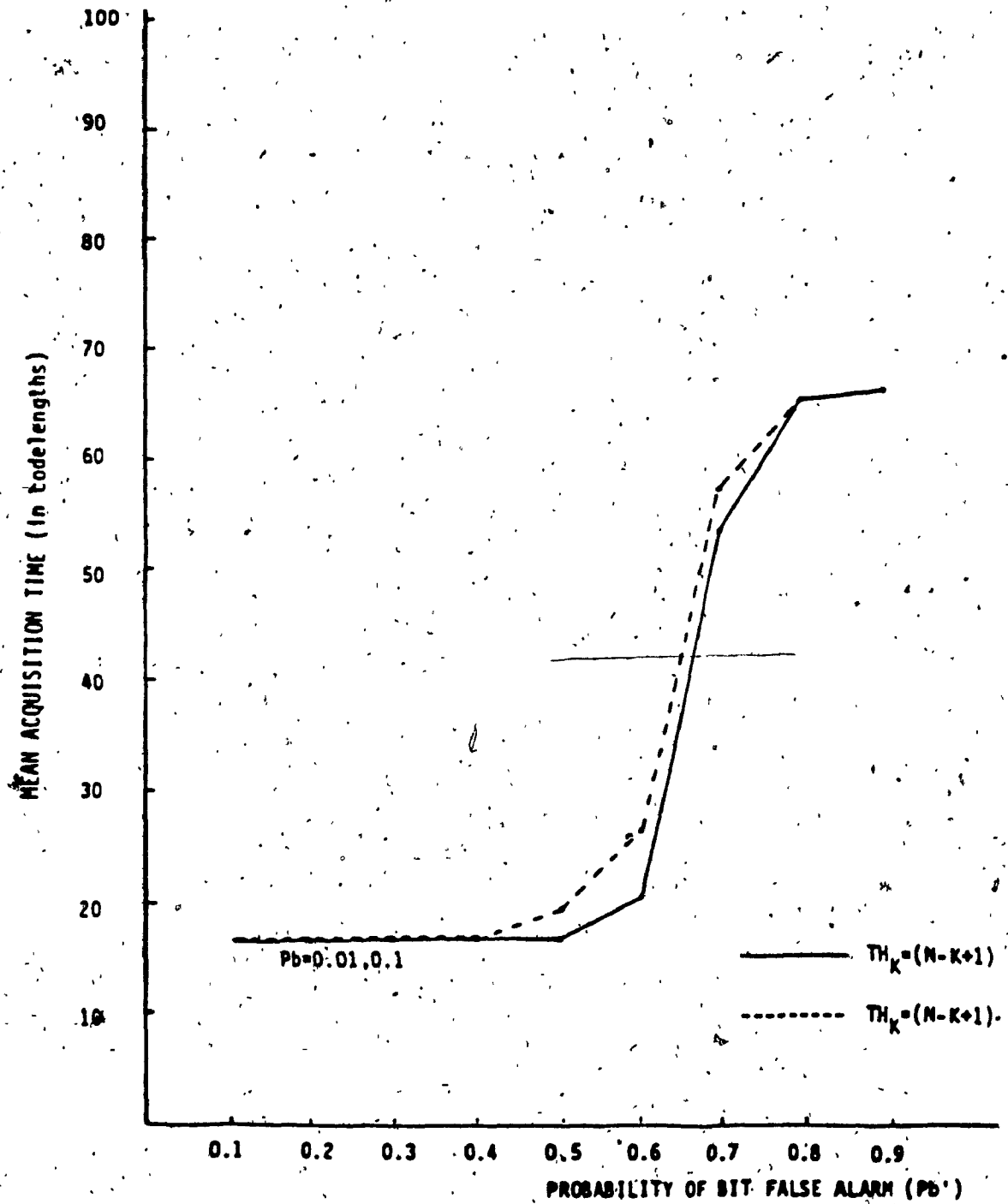


FIG. 5.7 a) : MEAN ACQUISITION TIME VERSUS P_b' FOR $TH_K = (N-K+1)$ AND $(N-K+1)-2$
($P_b = 0.01, 0.1$, $N = 8$, $M = 32$, $NK = 5N/8$, $TH = 10M/16$)
CODELENGTH = 256 chips.

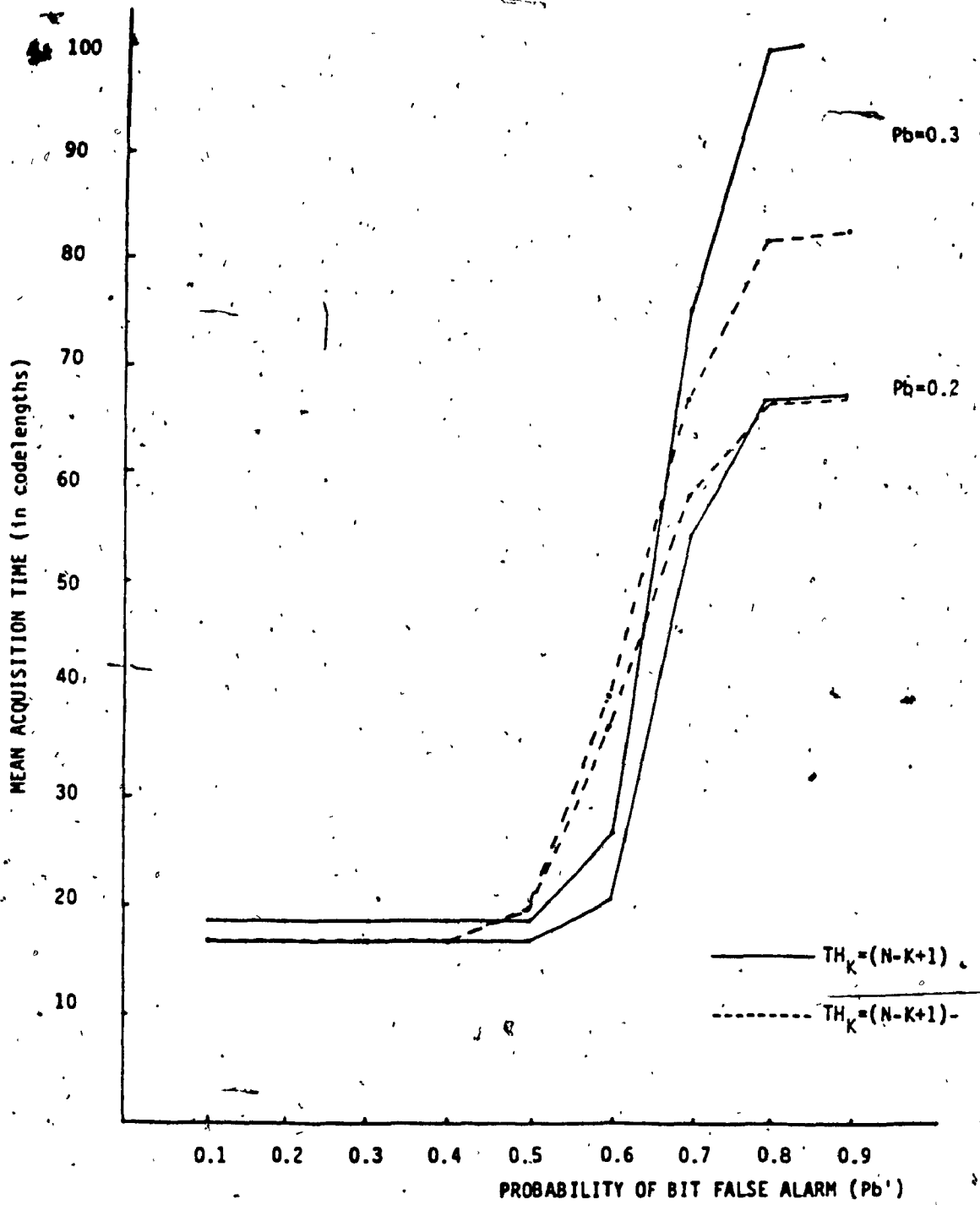


FIG. 5.7 b) : MEAN ACQUISITION TIME VERSUS P_b' FOR $TH_K = (N-K+1)$ AND $(N-K+1)-2$
($P_b = 0.2, 0.3$, $N = 8$, $M = 32$, $NN = 5N/8$, $TH = 10M/16$)
CODELENGTH = 256 chips.

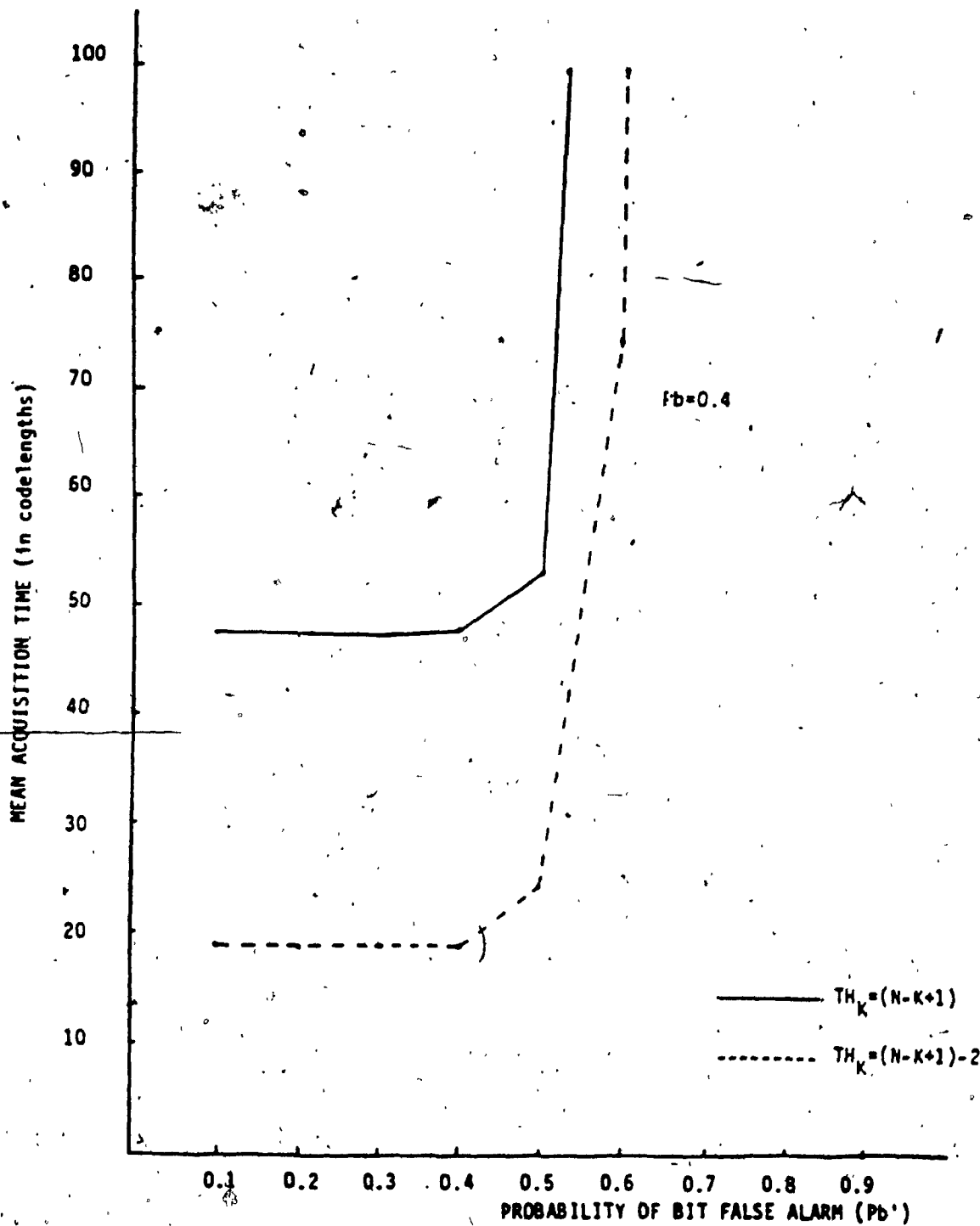


FIG. 5.7 c) : MEAN ACQUISITION TIME VERSUS P_b' FOR $TH_K = (N-K+1)$ AND $(N-K+1)-2$
($P_b = 0.4$, $N = 8$, $M = 32$, $NN = 5N/8$, $TH = 10M/16$)
CODELENGTH = 256 chips.

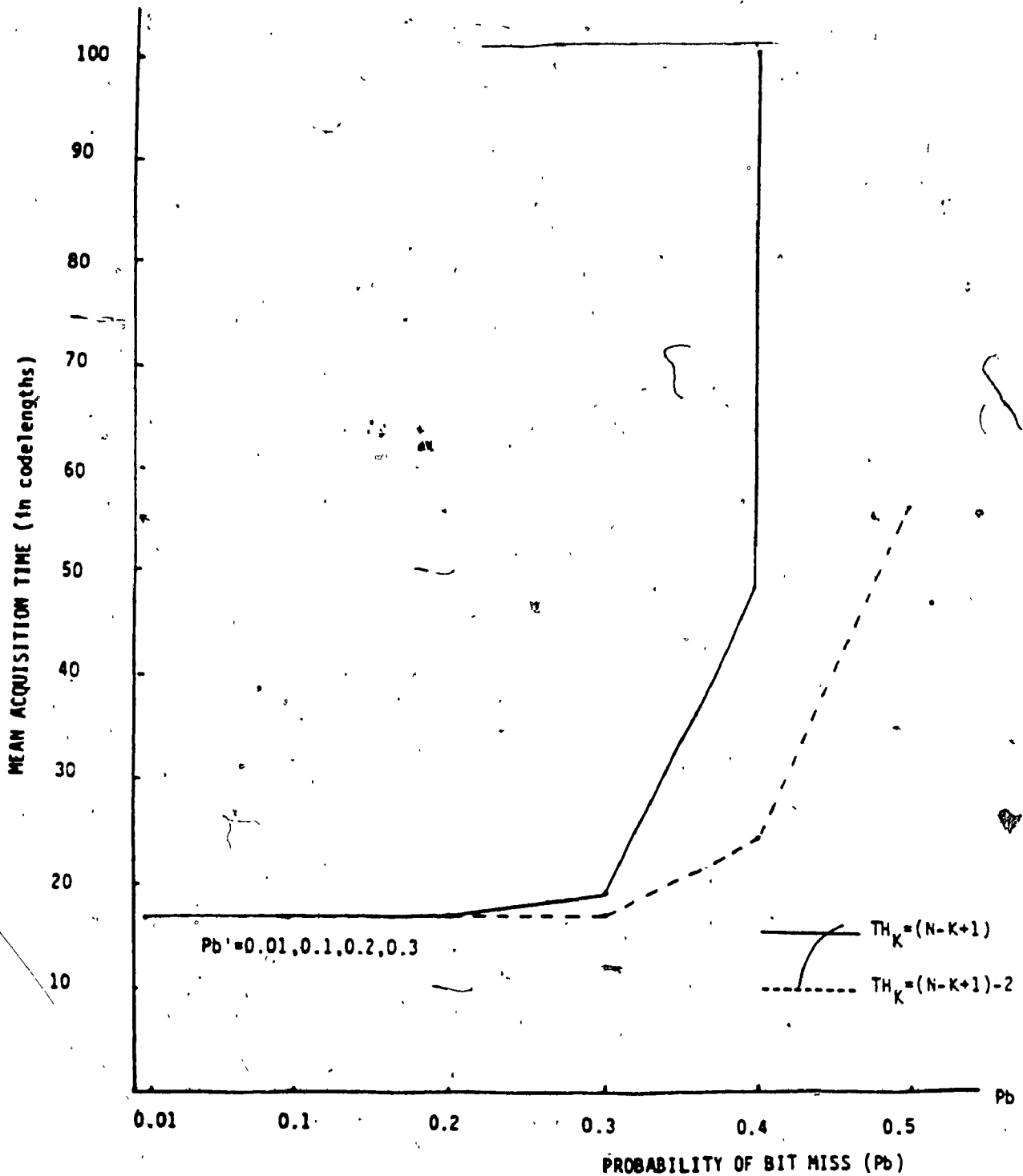


FIG. 5.8 a) ; MEAN ACQUISITION TIME VERSUS P_b FOR $TH_K = (N-K+1)$ AND $(N-K+1)-2$
($P_b' = 0.01-0.3$, $N = 8$, $M = 32$, $NN = 5N/8$, $TH = 10M/16$)
CODELENGTH = 256 chips.

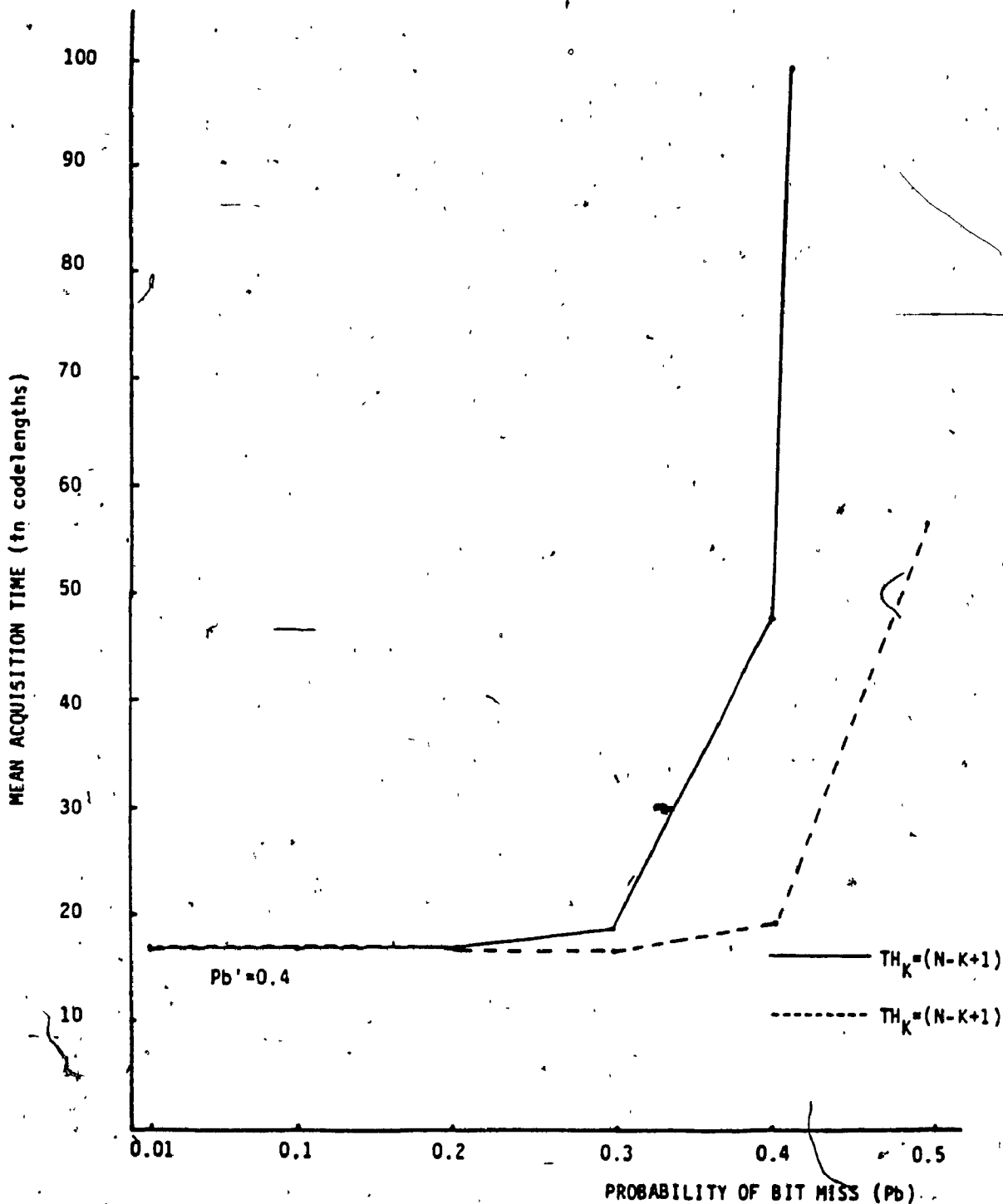


FIG. 5.8 b) : MEAN ACQUISITION TIME VERSUS P_b FOR $TH_K = (N-K+1)$ AND $(N-K+1)^2$
($P_b' = 0.4$, $N = 8$, $M = 32$, $NN = 5N/8$, $TH = 10M/16$)
CODELENGTH = 256 chips.

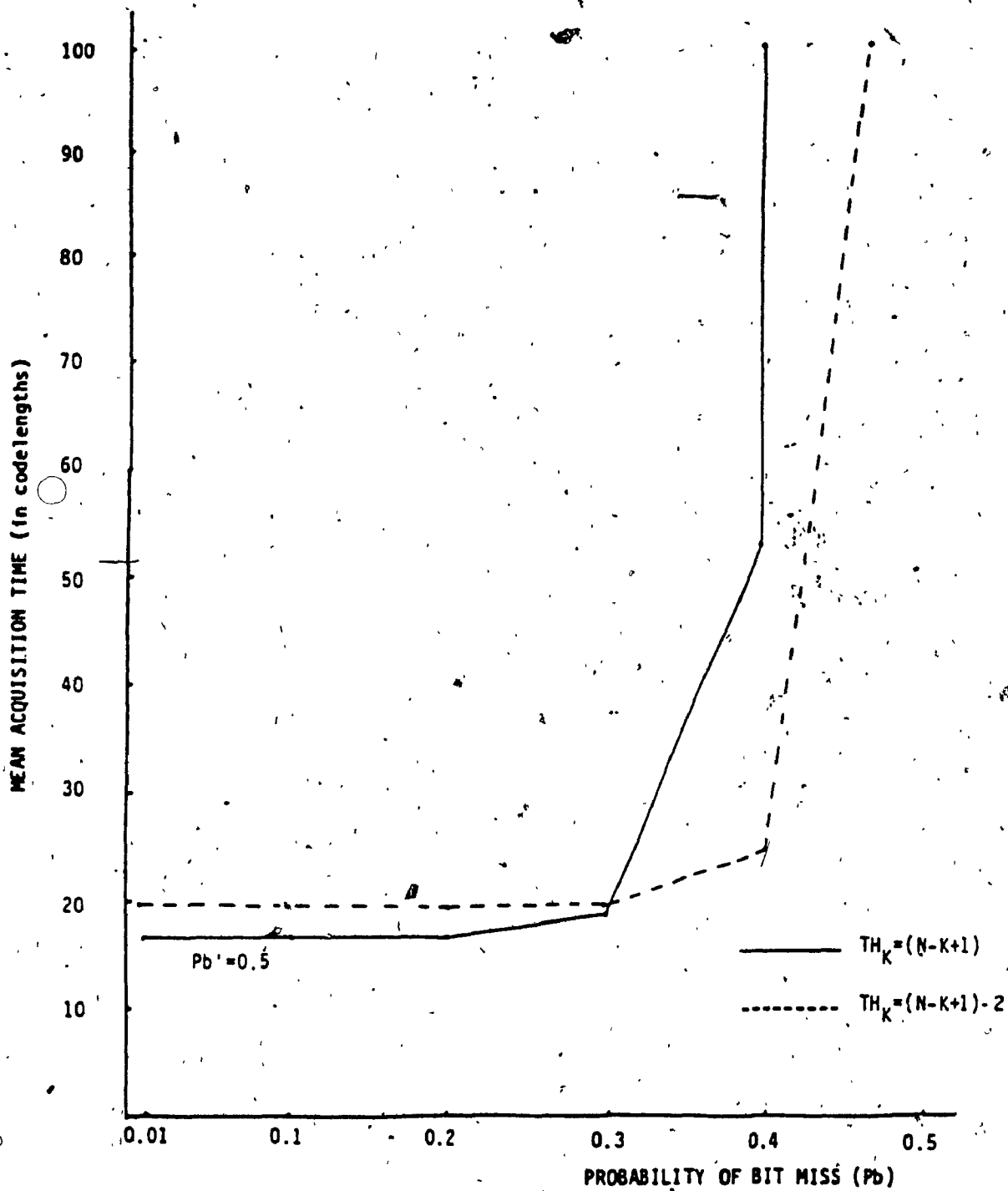


FIG. 5.8 c) : MEAN ACQUISITION TIME VERSUS P_b FOR $TH_K = (N-K+1)$ AND $(N-K+1)-2$
($P_b' = 0.5$, $N = 8$, $M = 32$, $NN = 5N/8$, $TH = 10M/16$)
CODELENGTH = 256 chips.

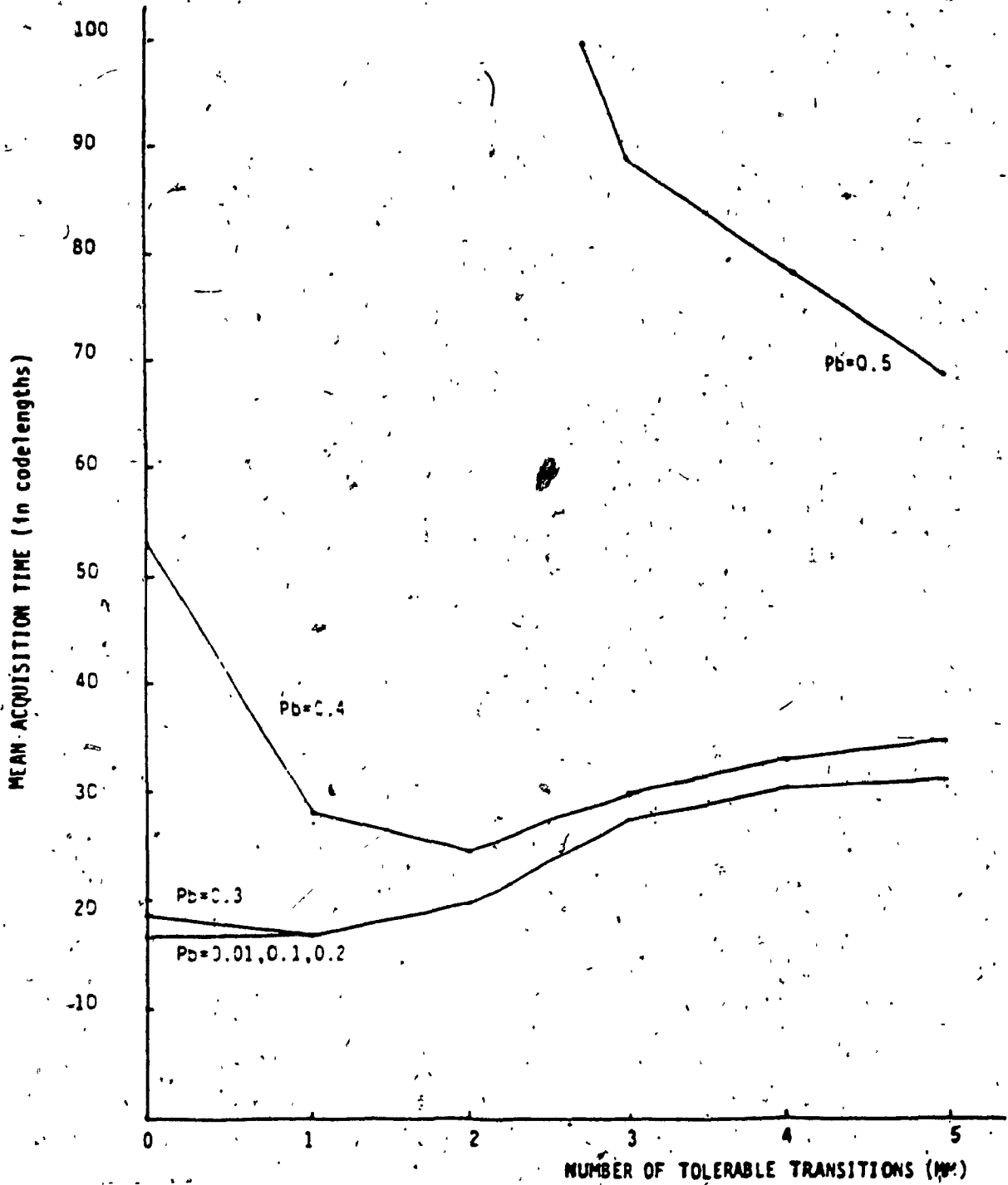


FIG. 5.9 a) : MEAN ACQUISITION TIME VERSUS NUMBER OF TOLERABLE TRANSITIONS
MM FOR VARYING VALUES OF Pb
(Pb = 0.5, N = 8, M = 32, NN = 5N/8, TH = 10M/16, TH_K =
(N-K+1)-MM)
CODELENGTH = 256 chips.

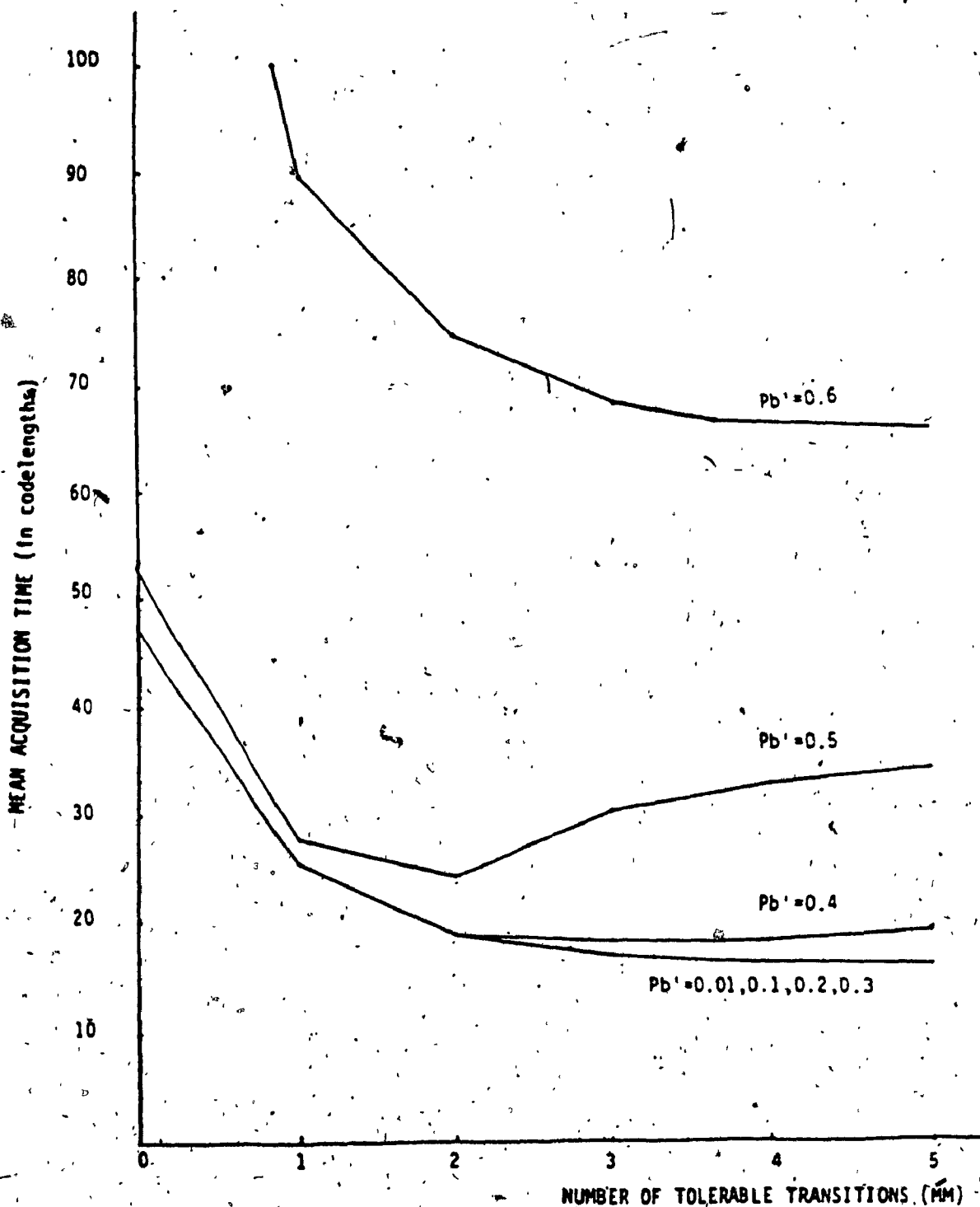


FIG. 5.9 b) : MEAN ACQUISITION TIME VERSUS NUMBER OF TOLERABLE TRANSITIONS MM FOR VARYING VALUES OF P_b'
($P_b = 0.4$, $N = 8$, $M = 32$, $N_N = 5N/8$, $T_H = 10M/16$, $T_{H_K} = (N-K+1)-MM$)
CODELENGTH = 256 chips.

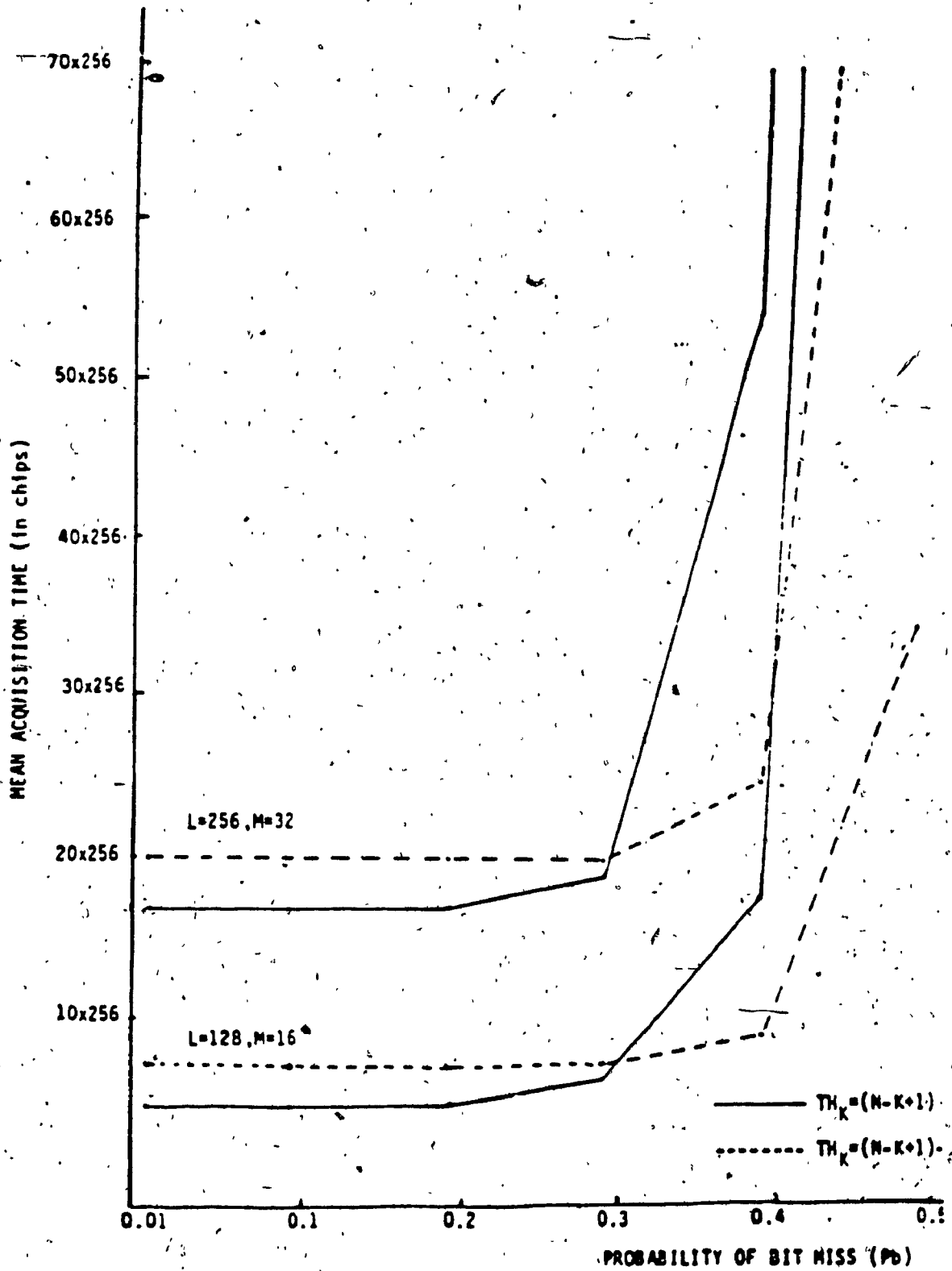


FIG. 5.10 a) : MEAN ACQUISITION TIME VERSUS P_b FOR $L = 256$ ($M = 32$) AND $L = 128$ ($M = 16$)
($P_s = 0.5$, $N = 8$, $MM = 5N/8$, $TH = 10M/16$).

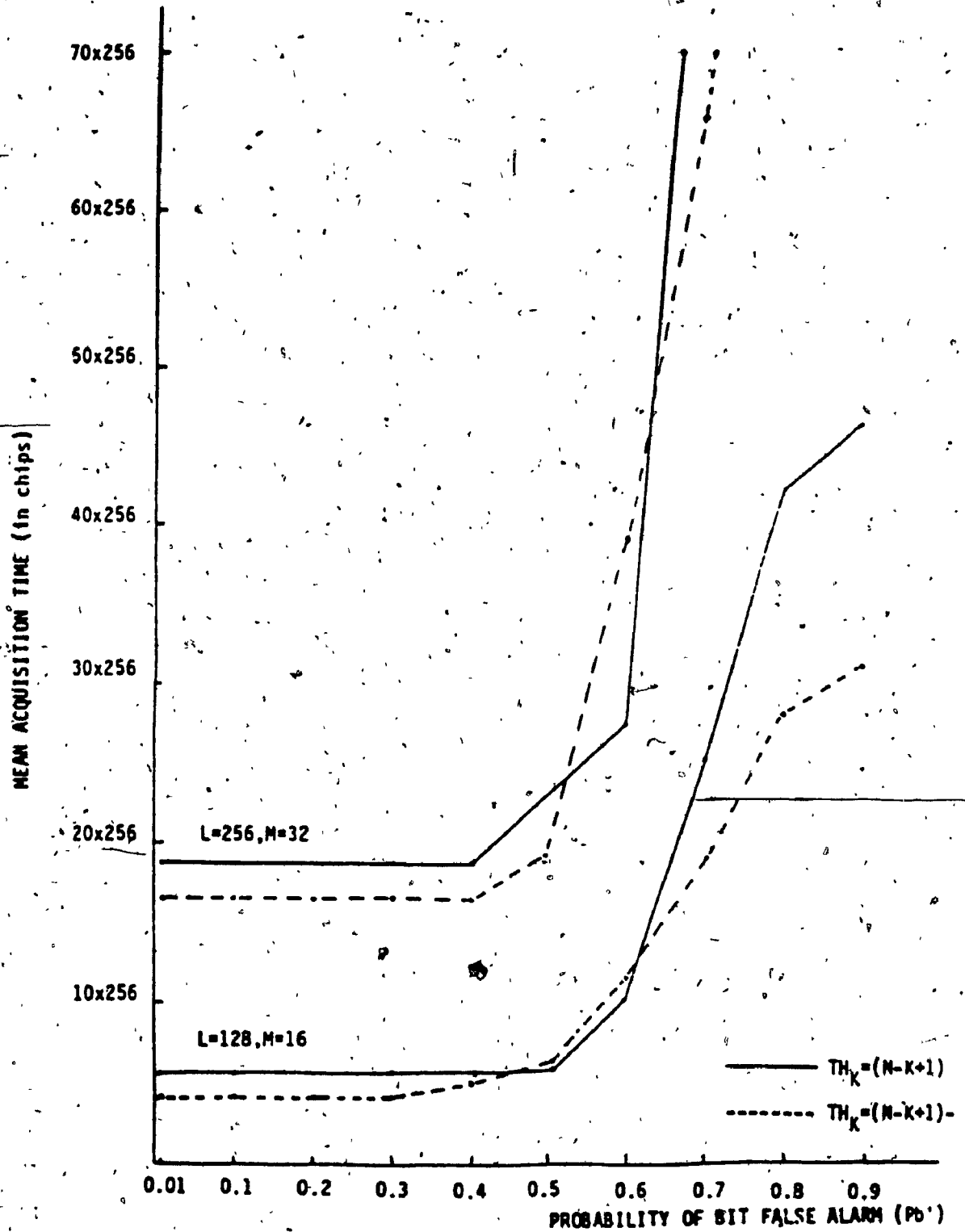


FIG. 5.10 b) : MEAN ACQUISITION TIME VERSUS P_b FOR $L = 256$ ($M = 32$) AND $L = 128$ ($M = 16$) ($P_b = 0.3, N = 8, NN = 5N/8, TH = 10M/16$).

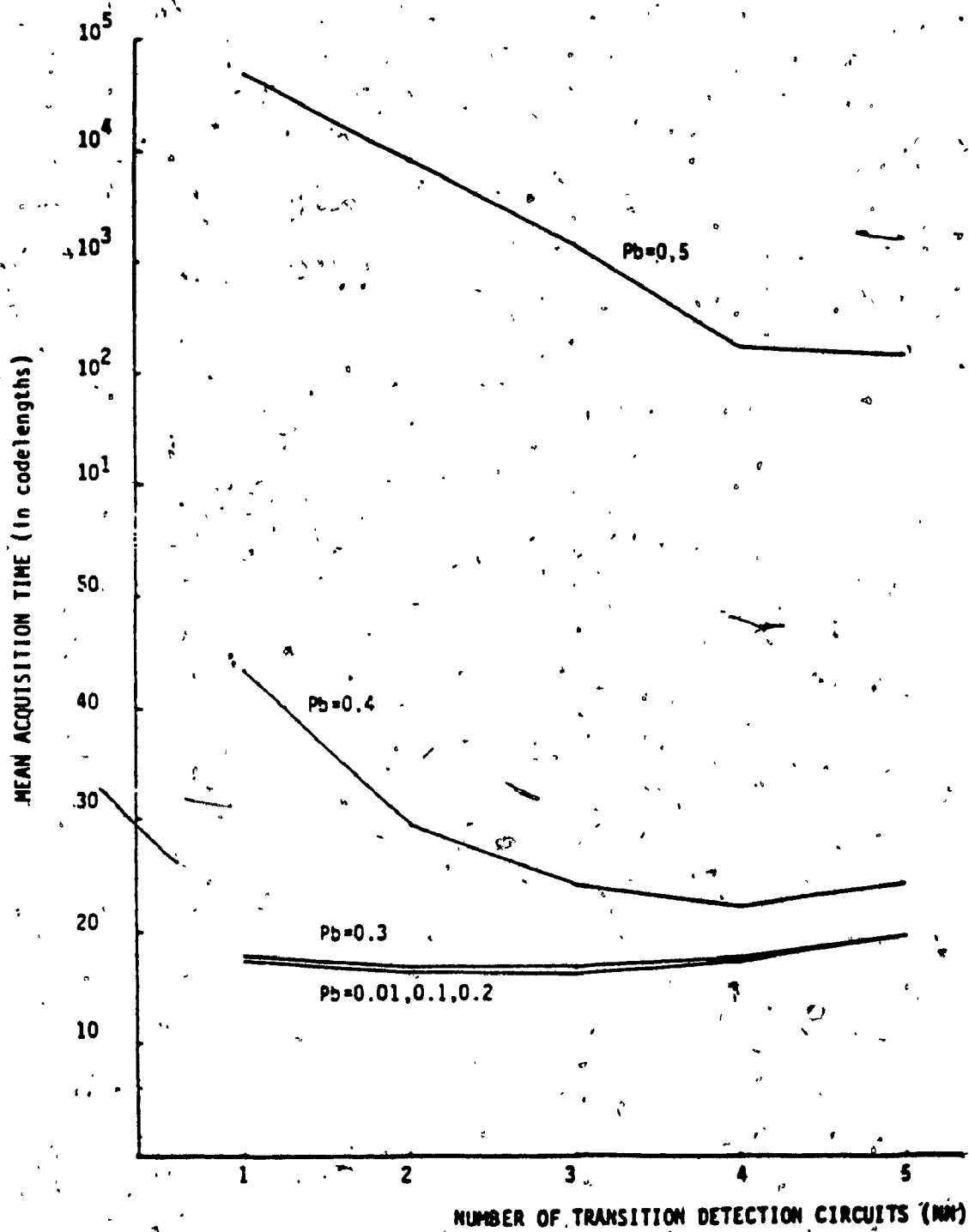


FIG. 5.11 a) : MEAN ACQUISITION TIME VERSUS NUMBER OF TRANSITION DETECTION CIRCUITS NM FOR VARYING VALUES OF Pb
 (Pb = 0.5, N = 8, M = 32, NM = 5N/8, TH = 10M/16, TH_K = (N-K+1)-2)
 CODELENGTH = 256 chips.

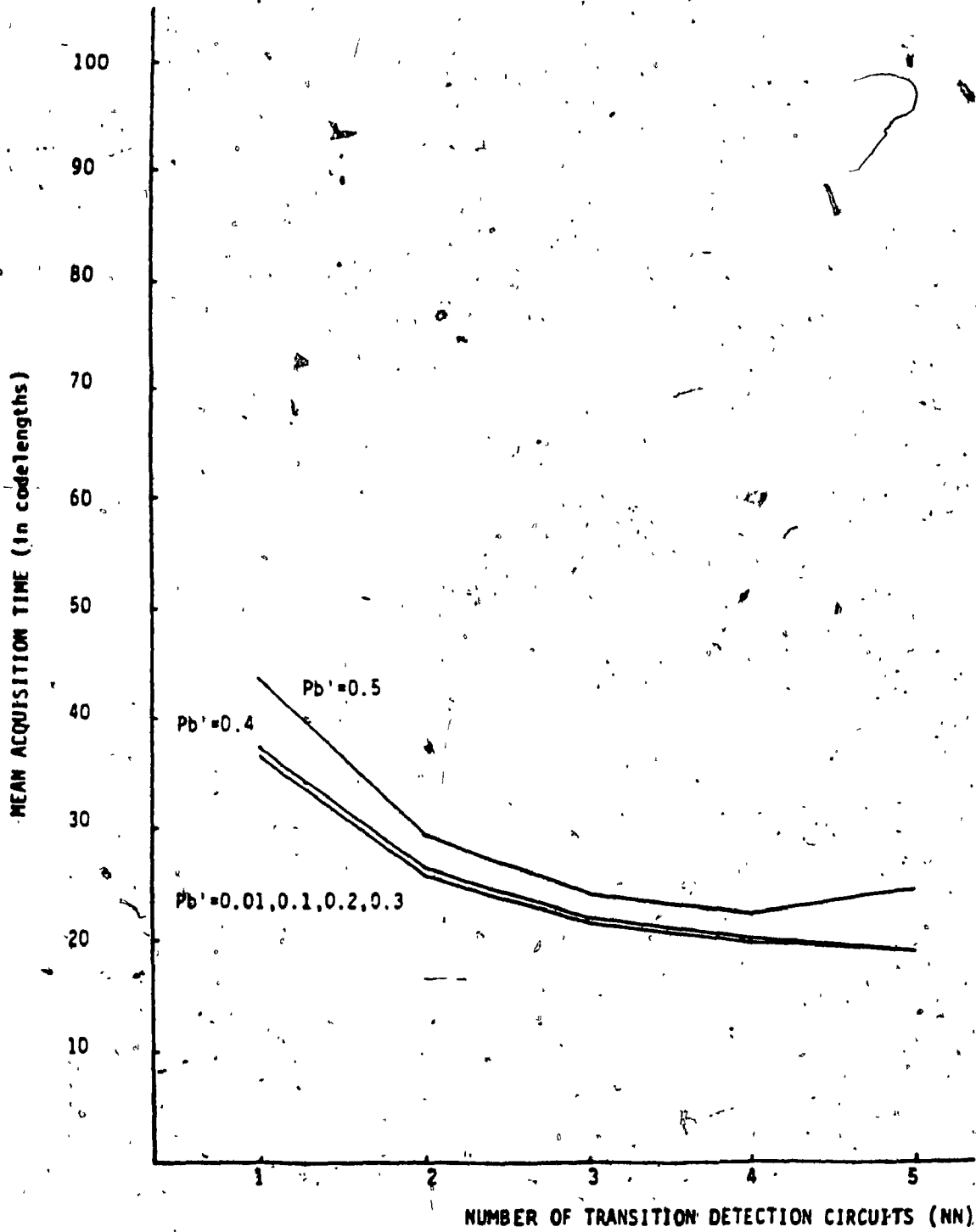


FIG. 5.11 b) ; MEAN ACQUISITION TIME VERSUS NUMBER OF TRANSITION DETECTION CIRCUITS NN FOR VARYING VALUES OF P_b'
($P_b = 0.4$, $N = 8$, $M = 32$, $NN = 5N/8$, $TH = 10M/16$, $TH_K = (N-K+1)-2$)
CODELENGTH = 256 chips.

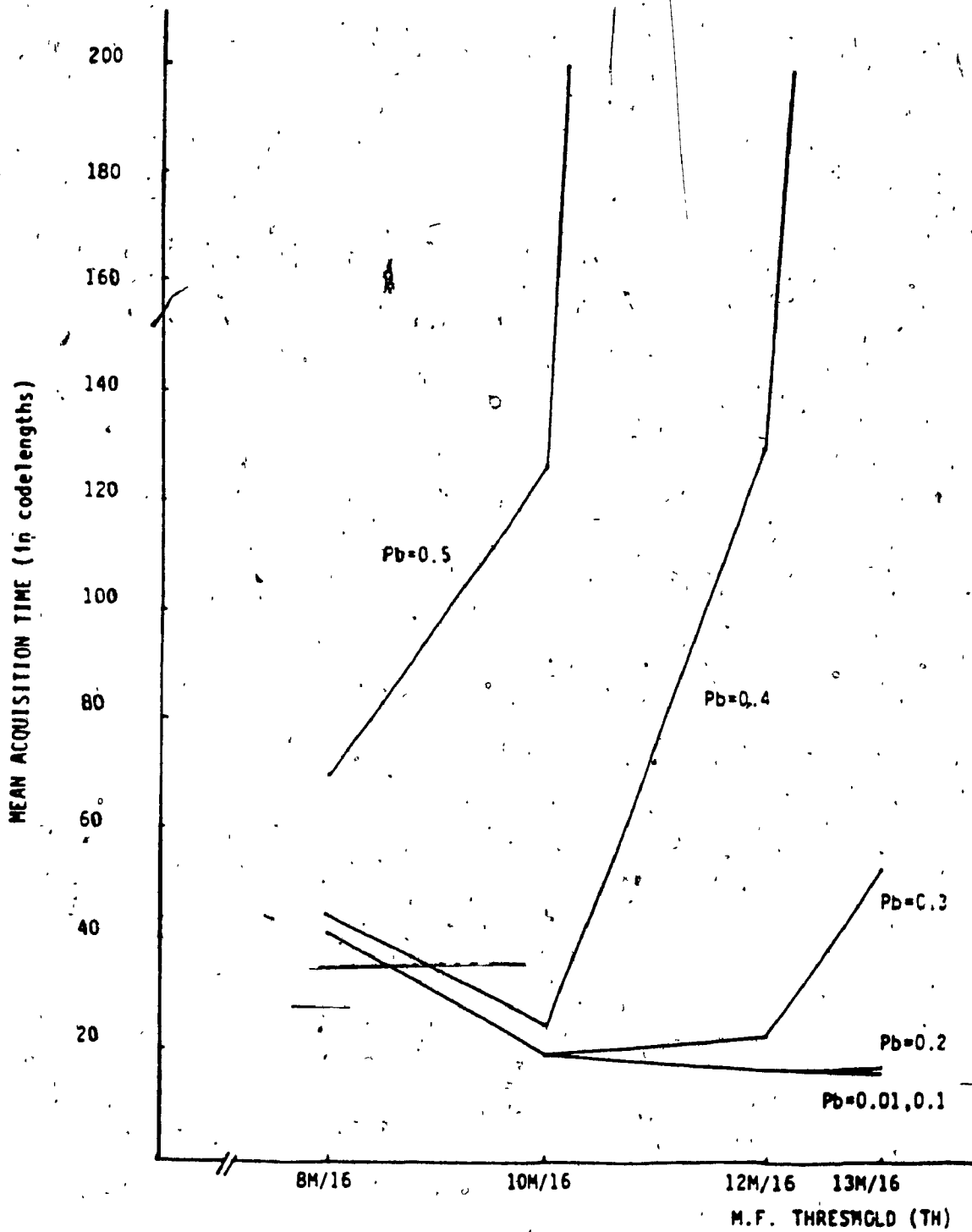


FIG. 5.12 a) : MEAN ACQUISITION TIME VERSUS TAD THRESHOLD TH FOR VARYING VALUES OF Pb
(Pb = 0.5, N = 8, M = 32, NN = 5M/8, TH_k ≡ (N-K+1)-2)
CODELENGTH = 256 chips.

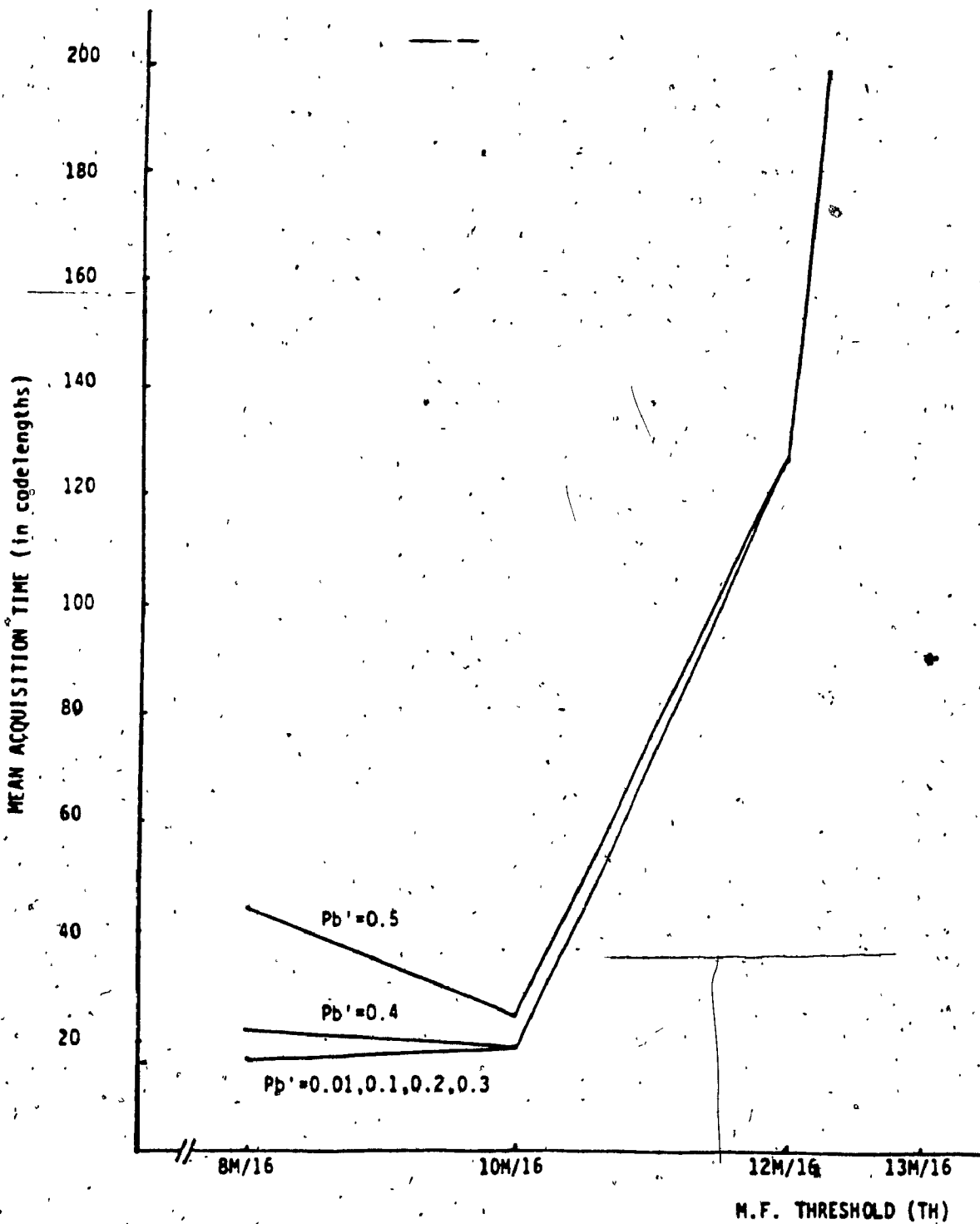


FIG. 5.12 b) : MEAN ACQUISITION TIME VERSUS TAD THRESHOLD TH FOR VARYING VALUES OF P_b'
($P_b = 0.4$, $N = 8$, $M = 32$, $NN = 5M/8$, $TH_{K'} = (N-K+1)-2$)
CODELENGTH = 256 chips.

TABLE 5.1A : α , β , μ_d , μ_m , μ_f AND CORRESPONDING MEAN AND STANDARD DEVIATION UNDER VARIOUS PROBABILITY OF BIT MISS P_b FOR $L = 128$ ($M = 16$) AND $P_b = 0.5$

P_b	0.1	0.2	0.3	0.4	0.5
α	0.999495	0.973343	0.824676	0.527174	0.227249
β	0.227249	0.227249	0.227249	0.227249	0.227249
μ_d	0.943813	0.933260	0.817298	0.310516	0.022783
μ_m	0.012640	0.014655	0.02931	0.081870	0.159484
μ_f	0.615920	0.615920	0.615920	0.615920	0.615920
\bar{T}	1697.69	1701.10	1738.74	2304.25	9036.36
σ	972.44	973.14	981.05	1145.95	6570.62

$P_{align} = 0.0625$ $M = 8$ $MN = 5$ $TH = 10M/16$ $TH_K = (M-K+1)-2$

- P_b : Probability of bit miss
- P_b' : Probability of bit false alarm
- N : number of (N.F.)⁰
- MN : number of (S.B.C.)⁰
- M : number of bits per (N.F.)
- TH : threshold at (N.F.)
- TH_K : threshold at (S.B.C.) member K
- L : code length
- α : Detection probability of (N.F.)
- β : false alarm probability of (N.F.)
- μ_d : overall probability of correct detection
- μ_m : overall probability of wrong phase detection
- μ_f : overall probability of false alarm
- \bar{T} : mean acquisition time (in units of bits)
- σ : standard deviation (in units of bits)

TABLE 5.18 : α , β , μ_d , μ_m , μ_f AND CORRESPONDING MEAN AND STANDARD DEVIATION UNDER VARIOUS PROBABILITY OF BIT MISS P_b FOR $L = 256$ ($N = 32$) AND $P_b = 0.5$

P_b	0.1	0.2	0.3	0.4	0.5
α	0.999994	0.999946	0.867428	0.461807	0.107664
β	0.107664	0.107664	0.107664	0.107664	0.107664
μ_d	0.996872	0.996597	0.972001	0.354740	0.011576
μ_m	0.000349	0.000377	0.001442	0.021448	0.081035
μ_f	0.208699	0.208699	0.208699	0.208699	0.208699
\bar{F}	5053.80	5053.94	5063.25	6267.75	32354.80
σ	2886.46	2886.49	2887.64	3187.78	24017.72

$P_{align} = 0.03125$ $N = 6$ $NM = 5$ $TE = 10M/16$ $TH_L = (N-K+1)-2$

- P_b : probability of bit miss
- P_b' : probability of bit false alarm
- β : number of (M.P.)⁰
- NM : number of (T.D.C.)⁰
- N : number of bits per (M.P.)
- TE : threshold at (M.P.)
- TH_L : threshold at (T.D.C.) number K
- L : codalength
- α : detection probability of (M.P.)
- β : false alarm probability of (M.P.)
- μ_d : overall probability of correct detection
- μ_m : overall probability of wrong phase detection
- μ_f : overall probability of false alarm
- \bar{F} : mean acquisition time (in units of bits)
- σ : standard deviation (in units of bits)

TABLE 5.2A : α ; β ; μ_d ; μ_m ; μ_f AND CORRESPONDING MEAN AND STANDARD DEVIATION UNDER VARIOUS PROBABILITY OF BIT FALSE ALARM P_b FOR $L = 128$ ($M = 16$) AND $P_b = 0.3$

P_b	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
α	0.824687	0.824687	0.824687	0.824687	0.824687	0.834687	0.824687	0.824687	0.824687
β	0.000000	0.000248	0.007130	0.058319	0.227249	0.527174	0.824687	0.973343	0.999495
μ_d	0.976702	0.976667	0.975684	0.969850	0.817298	0.358122	0.123062	0.060087	0.051455
μ_m	0.000000	0.000000	0.000000	0.000284	0.029310	0.389674	0.861436	0.939844	0.948527
μ_f	0.000000	0.000001	0.001198	0.070402	0.615920	0.994122	0.999999	1.000000	1.000000
F	1091.06	1091.06	1092.35	1160.30	1738.74	2963.36	4933.91	7247.64	7978.63
σ	590.39	590.39	91.24	638.40	981.05	1353.75	2256.30	4278.98	4971.15

$P_{align} = 0.0625$ $M = 8$ $MN = 5$ $TH = 10M/16$ $TH_K = (M-K+1)-2$

- P_b : probability of bit miss
- P_b' : probability of bit false alarm
- β : number of (N.F.)⁰
- μ_d : number of (T.B.C.)⁰
- μ_m : number of bits per (N.F.)
- μ_f : threshold at (N.F.)
- F : threshold at (T.B.C.) number K
- L : codeword length
- α : detection probability of (N.F.)
- β : false alarm probability of (N.F.)
- μ_d : overall probability of correct detection
- μ_m : overall probability of wrong phase detection
- μ_f : overall probability of false alarm
- F : mean acquisition time (in units of bits)
- σ : standard deviation (in units of bits)

TABLE 5.2B : α , β , μ_d , μ_w , μ_f AND CORRESPONDING MEAN AND STANDARD DEVIATION UNDER VARIOUS PROBABILITY OF BIT FALSE ALARM P_b FOR $L = 256$ ($M = 32$) AND $P_b = 0.3$

P_b	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
α	0.867428	0.867428	0.867428	0.867428	0.867428	0.867428	0.867428	0.867428	0.867428
β	0.000000	0.000000	0.000143	0.008411	0.107664	0.461807	0.867428	0.993946	0.999994
μ_d	0.992650	0.992650	0.992639	0.991997	0.972001	0.494179	0.124459	0.068203	0.065947
μ_w	0.000000	0.000000	0.000000	0.000000	0.001442	0.257063	0.871216	0.931788	0.934047
μ_f	0.000000	0.000000	0.000000	0.001662	0.208699	0.978840	0.999999	1.000000	1.000000
T_f	4225.90	4225.90	4225.90	4232.66	5063.25	10034.74	16934.46	20868.89	21147.60
σ	2363.78	2363.78	2363.78	2368.07	2887.60	4964.00	5717.78	8238.71	8458.84

$P_{align} = 0.03125$ $M = 8$ $MN = 5$ $TH = 10M/16$ $TH_L = (M-K+1)-2$

- P_b : probability of bit noise
- P_b' : probability of bit false alarm
- K : number of (M.F.)⁰
- M : number of (T.D.C.)⁰
- N : number of bits per (M.F.)
- TH : threshold at (M.F.)
- TH_L : threshold at (T.D.C.) number K
- L : code-length
- α : detection probability of (M.F.)
- β : false alarm probability of (M.F.)
- μ_d : overall probability of correct detection
- μ_w : overall probability of wrong phase detection
- μ_f : overall probability of false alarm
- T_f : mean acquisition time (in units of bits)
- σ : standard deviation (in units of bits)

CHAPTER VI

COMPARISON WITH DICARLO'S AND MILSTEIN'S

SCHEMES

In this chapter, we will compare our new hybrid acquisition scheme with those of Dicarlo and Milstein [3, 4] to see the advantages and disadvantages our scheme has over these two systems.

6.1 Comparison with Dicarlo's scheme

Dicarlo's scheme is basically the multiple dwell trial search technique which has been described in the Introduction. A brief description of Dicarlo's work [3] is given in chapter I.

Readers unfamiliar with Dicarlo's scheme are urged to refer to that before considering further reading.

Dicarlo's mean acquisition time and standard deviation were shown to be superior compared to those of single dwell search. However, our hybrid search has provided an important factor of the order of 30 and more.

In Table 6.1 we compare the mean acquisition time and standard deviation of our scheme with those of Dicarlo.

To have an appropriate and fair comparison between the two schemes, we must note the different assumptions and input data set as well as parameter values in the two cases.

First there is the integration time difference. The integration time in our scheme is subcode time (32 bits) whereas in Dicarlo's it is in the range of thousand bits.

Secondly, in Dicarlo, scheme the offset factor (search step in fraction of a chip) is $\frac{1}{2}$ while in our scheme it is 1.

Dicarlo starts with a given value of P_D and a preassigned value of P_F and find the minimum value of $\mu(N)$ (the mean acquisition time) and $\sigma(N)$ (the standard deviation) where N is the number of dwells in the search process.

From P_D , P_{di} is calculated by

$$P_{di} = P_D^{1/N} \text{ for all } i$$

P_{fi} is calculated from equation (44) in Dicarlo's [3]

$$P_{fi} = Q \left\{ \frac{\sqrt{C(SNR)} - X \sqrt{2+SNR}}{\sqrt{2 - 2(SNR)}} \right\} \quad (6.1)$$

where

$$C = \frac{\bar{\tau}}{\Delta} \text{ and } X = Q^{-1}(1 - P_{di}) \quad (6.2)$$

$\bar{\tau}$ = the average integration time in Dicarlo's scheme

$$= \frac{1}{N} \sum_{i=1}^N \tau_i$$

Δ = bit duration time

Q = Gaussian integral function

$$Q(x) = \frac{1}{\sqrt{\pi}} \int_x^{\infty} e^{-\mu^2/2} d\mu \quad (6.4)$$

P_{di} , P_{fi} correspond to the probabilities α and β in our scheme.

Since as stated above the integration time in the two schemes are different ($\frac{\bar{\tau}}{\Delta}$ in Dicarlo's, and 32 chips in our scheme). To derived proper corresponding values for α and β in our scheme, we defined a factor where

$$K^* = \frac{\bar{\tau}/\Delta}{32} \quad (6.5)$$

The corresponding α and β - then are

$$\alpha = P_{di}^{1/K^*} \quad (6.6)$$

$$\beta = P_{fi}^{1/K^*} \quad (6.7)$$

These values of α and β were used in our scheme as a starting point to calculate the mean acquisition time and standard deviation.

The results are compared with those results of $\frac{(N)}{L}$ and $\frac{6(N)}{N}$ in Dicarlo's scheme (Table 6.1).

Since our results are calculated in units of chips. The corresponding value of $\mu(N)$ and $6(N)$ will be

$$\frac{\mu(N)^n}{\Delta L} = 2 \times \frac{\mu(N)}{\Delta q} \quad (6.8)$$

$$\text{and } \frac{\mathcal{G}(N)}{\Delta L} = 2 \times \frac{\mathcal{G}(N)^2}{\Delta q} \quad (6.9)$$

Where

L: code length in chips

q = 2L as assumed in Dicarlo's

Since as stated above in our scheme we used the offset factor of 1 while in Dicarlo's it is $\frac{1}{2}$ so to have a fair comparison our \bar{T} and $\bar{\mathcal{G}}$ were multiplied by 2.

The fair comparison using the corresponding values of α , β , $\frac{\bar{\tau}}{\Delta}$, K for different values of \hat{P}_p , \hat{P}_f , N (taken from Dicarlo's [3]).

From the point of view of penalty time, it is interesting to see that our results in Table 6.1 are actually worse case results (we included the penalty time) while Dicarlo's are best cases (i.e. ignoring penalty time).

Finally considering Table 1 we see that a typical mean acquisition time case is roughly $63.5 \times 2 \approx 127$ chips which is better with a factor of $\frac{3558}{127} \approx 34$ than the best case of Dicarlo's at $P_d = 0.99$, $P_f = 10^{-3}$ and $N = 2$.

6.2 Comparison with Milstein's scheme

Milstein's scheme is a search - and lock mode-type using parallel detection in the search-mode. His method is summarized and presented in the chapter I.

TABLE 6.1 : COMPARISON WITH DICARLO'S SCHEME

\hat{P}_D	\hat{P}_F	N	P_{di}	P_{fi}	K^*	α	β	$\frac{\mu(N)}{9A}$	$\frac{\sigma(N)}{9A}$	\bar{T}	$\bar{\sigma}$
0.99	10^{-3}	1	0.9900	0.0001	230.28	0.9999956	0.960793	3070	1627	62.33071	21.52631
		2	0.9949	0.0351	120.97	0.9999958	0.972690	1779	943	63.47616	21.73278
	10^{-4}	1	0.9900	0.0001	239.34	0.9999958	0.962248	3764	1996	62.47025	21.55021
		2	0.9949	0.0119	147.03	0.9999965	0.970311	1849	980	63.24538	21.68911
		3	0.9966	0.0505	119.33	0.9999971	0.975289	1794	951	63.72544	21.78049
0.999	10^{-3}	1	0.9990	0.0010	240.47	0.9999995	0.971682	3905	2236	63.37345	21.71233
		2	0.9995	0.0336	163.47	0.9999996	0.979456	2436	1395	64.12523	21.85932
	10^{-4}	1	0.9990	0.0001	292.03	0.9999996	0.968953	4687	2684	63.11000	21.66327
		2	0.9995	0.0119	192.91	0.9999997	0.977291	2513	1439	63.91550	21.81721
		3	0.9997	0.0475	160.63	0.9999997	0.981208	2471	1415	64.29491	21.89398

As proved his scheme is better with a factor $\frac{2}{K}$ in the mean acquisition time compared with that of the serial search scheme, where K is the uncertainty region length in chips.

To compare our scheme with his scheme, we have to establish correspondence between Milstein's parameters and ours.

In his scheme the mean acquisition time is given by

$$T_{acq} = \frac{T + T_f Pe^{(s)}}{1 - Pe^{(s)}} \quad (6.10)$$

Where $Pe^{(s)}$ corresponds to $P_{align} \cdot \alpha$ in our scheme and T_f is the penalty time and is given as

$$T_f = \frac{2J(1-P_L)^2(1+P_S) + (N+1)P_S^2(2-P_L)}{(1-P_L)^2 [1-P_S(1-P_S)]} T \quad (6.11)$$

where P_S corresponds to $(1 - P_{align} \cdot \alpha)$ in our scheme as explained in the Appendix B.

P_L corresponds to our μ_f , i.e. the overall probability of false alarm.

Another parameter difference is the integration time ($2T$ in Milstein case and NT in our case).

So for fair comparison, if was rederived and $2T$ replaced by NT leading to the fair

$$T_f = \frac{N(1-P_L)^2(1+P_S) + (N+1)P_S^2(2-P_L)}{(1-P_L)^2 [1-P_S(1-P_S)]} \quad (6.12)$$

He also did not consider the verifying time spent after certain code phase has been selected to the time when the detector is declared. This time is T_i as explained in [1]. In our scheme this time is automatically and implicitly as we go from one subcode to another.

So the expression of the mean acquisition time derived by Milstein has to be modified taken into account T_i .

Based on the above the mean acquisition time of Milstein is then rederived using the flow diagram technique (this will give the same result).

The result is (the fair comparison version)

$$T_{acq} = \frac{NT + T_i Pe^{(s)} + T_f (1 - Pe^{(s)})}{Pe^{(s)}} \quad (\text{in seconds}) \quad (6.13)$$

$$= \frac{N + T_i' P_{align}^{\alpha} + T_f' (1 - P_{align}^{\alpha})}{P_{align}^{\alpha}} \quad (\text{in subcode lengths})$$

$$= \frac{N + T_i' P_{align}^{\alpha} + T_f' (1 - P_{align}^{\alpha})}{P_{align}^{\alpha} N} \quad (6.14)$$

$$(\text{in code lengths}) \quad (6.15)$$

where

$$T_i' = \frac{T_i}{T} \quad (6.16)$$

$$T_f' = \frac{T_f}{T} \quad (6.17)$$

which are the verification time and penalty time calculated in units of subcode length.

Table 2 shows a sample of the fair comparison between Milstein's acquisition time and standard deviation, and our corresponding results. It is easily seen that our technique outperforms Milstein's.

The initial search to choose a correct position is based on two possibilities. The first step is establishing the correspondance between the parameter so fair comparison will prevail i.e., the assign of fair values to α , β . It is readily seen the probability

$$1 - P_e^{(s)} = P_{\text{align}} \cdot \alpha \quad (6.18)$$

which is the probability that the code was aligned and the correct TAD peaked and was selected.

Similarly

$$P_e^{(s)} = 1 - P_{\text{align}} \cdot \alpha \quad (6.19)$$

i.e. the probability that the phase position has been decided upon two more successive "hits" at that phase position will be required in order to enter into the lockmode.

In the case of the correct phase is selected the probability of hit.

$$P_S = P_{align} \alpha \quad (6.20)$$

and in the case of incorrect code phase this probability will be

$$P_S = (1 - P_{align} \alpha) \beta \quad (6.21)$$

Where $(1 - P_{align} \alpha)$ is the probability where there is no alignment and the right TAD not peaking.

β is the probability that the TAD correspond to the wrong code phase peaking and assumed automatically selected.

Follow the same procedure in Milstein's paper using the new adjusted diagram and [1].

T_f was derived as:

$$T_f = \frac{1 + \beta(1 - P_{align} \alpha)}{1 - \beta(1 - P_{align} \alpha)(1 - \beta(1 - P_{align} \alpha))} \quad NT$$

$$+ \frac{\beta^2(1 - P_{align} \alpha)^2(2 - \mu_f)(N+1)}{(1 - \mu_f)^2 [1 - \beta(1 - P_{align} \alpha)(1 - \beta(1 - P_{align} \alpha))]} \quad T \quad (6.22)$$

and

$$T_i = \frac{1 + P_{align} \alpha}{1 - P_{align} \alpha (1 - P_{align} \alpha)} \quad (6.23)$$

The mean acquisition is derived following the flow diagram is

$$T_{acq} = \frac{N + T_i P_{align} \alpha + T_f (1 - P_{align} \alpha)}{P_{align} \alpha N} \quad (6.24)$$

With the above analysis, to include our scheme is better than Milstein's with a factor of about 2.

TABLE 6.2 : COMPARISION WITH MILSTEIN'S SCHEME

β	0.1	0.2	0.3	0.4	0.5	0.6	
Tacq	40.57515	45.45663	51.52922	59.24348	70.13420	94.13996	Milstein scheme
T	17.98705	18.39093	20.05697	24.56842	35.52173	64.49111	Our scheme

CHAPTER VII

SUMMARY, CONCLUSIONS AND SUGGESTIONS

Two new approaches were presented in this thesis to reduce the mean acquisition time, which were the optimal sequence detection circuit approach and the hybrid sequential/parallel scheme for combined acquisition/search lock of SS codes.

Synchronization is the most important aspect in a SS communication system. It is important because it allows the receiver to despread the received signal and then demodulate it to get the original information signal. The PN acquisition or coarse alignment is the first stage of the synchronization. It is to bring the local PN code to, say, within one chip of the received signal. After the code phase has been decided upon then the signal will go to the tracking phase. We have concentrated only in the acquisition phase. There have been several techniques which have the objective to minimize the mean acquisition time. Two most relevant techniques were those of Dicarlo and Milstein.

The scheme of Dicarlo is just the multi-dwell serial search technique which is the generalized version of the single serial search technique. It allows the code phase to be decided upon in a time less than the maximum dwell (integration) time.

Milstein's scheme uses a parallel technique to detect the synchronization. It has a search phase in which the code is fed into a band of parallel convolvers to decide the correct code phase. After the code

phase has been chosen the code goes to the lock phase. The code will be locked at that code phase and is fed into the convolvers in a sequential manner. The output of the convolvers is now summed with the integration time equal the whole code length time. If the output of the summer exceeds the threshold then synchronization will be declared and demodulation is proceeded.

As stated above we proposed two new techniques with the objective to improve the results of Dicarlo's and Milstein's schemes. In chapter II we described some background concerning some basic methods for the acquisition phase. Two most essential methods to be familiar with are the single dwell serial search technique and the multiple dwell serial search scheme.

We used chapter III to describe the optimal SDC approach. The SDC technique has been studied and examined in a previous research by Elhakeem and Shawn. Here we tried to optimize the system. We rederived some of the equations and relaxed some assumptions in Shawn paper especially that the number of the SDCs (NN) can be different from the number of TAD (N).

To optimize the scheme we changed all the parameters and tried to find the optimal mean acquisition time and standard deviation along with the best threshold (TH) and the best number of SDCs (NN) we should choose to have such an optimal result.

We recorded the results in a number of Tables to help the designer to choose the suitable parameters in designing his system.

We also plot a number of curves of the mean acquisition time for varying values of several of the parameters.

We have built a prototype model for the SDC acquisition scheme. The details of the design and components of the system were described carefully in chapter IV. We also did three set of testing on the circuit built, the objective of which was to confirm the results in the theoretical part and the results shown were consistent with the theory in chapter III.

In chapter V we described and analyzed the hybrid parallel/sequential system. It was nothing but the dynamic version of the system in chapter III, in which when testing for the in-sequence of the subcodes the presence of some subcodes could be missed but the synchronization was still declared. In this scheme we also used a band of parallel correlators in the search mode to detect the presence of the subcodes. In the lock mode instead of having the SDCs we have the transition detection circuits (T.D.C.)^S. We had a band of parallel (T.D.C.)^S each counted the in-sequence transitions of the subcodes. After the whole code length had been investigated the counts of the (T.D.C.)^S were compared with the preset thresholds and then were compared with each other to decide the highest count. The phase of the (T.D.C.) with the highest count was chosen as the code phase and synchronization was declared. This scheme was similar to Milstein's but in the lock mode Milstein only used an array of the convolvers to sum up all the subcodes in a code length time while we used a band of parallel detection circuits

to decide the code phase. Our scheme had advantage since several subcodes could be missed (probably due to heavy jamming and interference or noise) however synchronization was still declared.

For this scheme we also plot a number of curves for the mean acquisition time versus some of the parameters especially the number of tolerable transitions (MM) to show that if we allowed certain number of subcodes to be skipped the performance was better. There was also the optimum here and the optimum moved to a certain direction depending on the variation of the parameters.

Finally in chapter VI the hybrid sequential/parallel scheme described in chapter V was compared with the schemes of Dicarlo and Milstein. To do the comparison we recorded the results in the papers of Dicarlo and Milstein and used the same input data with some modification to be suitable with our scheme to compute the corresponding mean acquisition time in our scheme. The results have shown that our scheme outperforms these two systems (the mean acquisition time is better with a factor of 34 compared with Dicarlo's scheme and a factor 2 with Milstein's).

Future Research Suggestions

In this thesis only the DS/SS system was considered; fine acquisition (PN tracking) was assumed to exist and also the effect due to Doppler was not treated. Approaches for the FH/SS type, the code rate change of Doppler effect and further investigation in the tracking phase are suggested in future reseaches.

REFERENCES

- [1] M.K. Simon et al, "Spread Spectrum Communications", Vol. III, Computer Science Press, Rockville, Maryland, 1985.
- [2] A.K. Elhakeem and S.A.M. Liebrecht, "A Multi-processing Approach to Spread Spectrum Code Acquisition", IEEE MILCOM'85 Proceedings, Boston, MA, Oct. 1985.
- [3] D.M. Dicarlo and C.L. Weber, "Multiple Dwell Serial Search: Performance and Application to Direct Sequence Code Acquisition", IEEE Trans. Commun., Vol. COM-31, No. 5, pp. 650-659, May 1983.
- [4] Milstein et al, "A Rapid Acquisition Technique Employing Parallel SAW Processing", IEEE MILCOM'83 Proceedings, pp. 501-505.
- [5] W.R. Braun, "Performance Analysis for the Expanding Search PN Acquisition Algorithm", IEEE Trans. Commun., Vol. COM-30, NO. 3, March 1982.
- [6] P.W. Baier et al, "A Novel Spread Spectrum Receiver Synchronization Scheme using a SAW-Tapped Delay Line", IEEE Trans. Commun., Vol. COM-30, No. 5, pp. 1037-1047, May 1982.
- [7] J.K. Holmes and C.C. Chen, "Acquisition Time Performance of PN Spread Spectrum Systems", IEEE Trans. Commun., Vol. COM-25, No. 8, pp. 778-783, Aug. 1977.
- [8] A.K. Elhakeem and Ho.T. Minh, "Optimal Sequence Detection Circuit Approach to Spread Spectrum Code Acquisition", IEEE MONTECH'86 Proceedings, Montreal, Canada, Sept. 1986.

REFERENCES (cont'd)

- [9] R.E. Ziemer and R.L. Peterson, "Digital Communications and Spread Spectrum Systems", Macmillan, New York, 1985.
- [10] W.H. Huggings, "Signal-flow graph and random signals", PROC IRE, vol. 45, pp. 74-86, January 1957.
- [11] S.J. Mason, "Feedback theory - Some properties of signal flow graphs", PROC IRE, vol. 41, pp. 1144-1186, September 1953.
- [12] S.J. Mason, "Feedback theory - Further properties of signal flow graphs", PROC IRE, vol. 44, pp. 920-926, July 1956.
- [13] R.C. Dixon, "Spread Spectrum Systems", Wiley, U.S.A., 1976.
- [14] C.R. Cahn, "Spread Spectrum Application and State-of-the-Art Equipments", Magnavox/advanced products division, June 1983.
- [15] S. Rappaport and D.M. Grieco, "Spread Spectrum Signal Acquisition: Methods and Technology", IEEE Comm. Mag., Vol. COM-22, No. 6, June 1984.
- [16] ~~W.R. Braun~~, "Comparison Between Variable and Fixed Dwell-Time PN Acquisition Algorithms", Proc. Int. Conf. Comm., Denver, CO, June 1981.
- [17] A.K. Elhakeem, G.S. Takhar and S.C. Gupta, "New Code Acquisition Techniques in Spread Spectrum Communications", IEEE Trans. Comm., Vol. COM-28, No. 2, Feb 1980.
- [18] S.A. Musa and W. Wasylkiwskyj, "Co-Channel Interference of Spread Spectrum Systems in a Multiple User Environment", IEEE Trans. Comm., Vol. COM-26, No. 10, October 1978
- [19] T. Masamura et al, "Differential Detection of MSK with Non-redundant Error Correction", IEEE Trans. Comm., Vol. COM-27 No. 6, June 1979.
- [20] C.E. Cook et al, "Spread Spectrum Communications", IEEE Press, New York, 1983.