PPM SEQUENCES WITH DESIRABLE CORRELATION PROPERTIES

A Thesis

by

MOE ZAW WIN

University Undergraduate Fellow, 1986-87

Approved as to style and content by:

Dr. C. N. Georghiades
(Chairman of Committee)

Dr. L. M. Canfield
(Honors Director)

Dr. J. W. Howze
(Head of Department)

April 1987

PPM SEQUENCES WITH DESIRABLE CORRELATION PROPERTIES

A Thesis

by

MOE ZAW WIN

University Undergraduate Fellow, 1986-87

Submitted to the College of Engineering of
Texas A&M University
in partial fulfillment of the requirement for the degree of

BACHELOR OF SCIENCE

April 1987

Major Subject: Electrical Engineering

ABSTRACT

PPM SEQUENCES WITH DESIRABLE CORRELATION PROPERTIES. (April 1987)

Moe Zaw Win, B.S., Texas A&M University

Chairman of Advisory Committee: Dr. C. N. Georghiades

The problem of designing Q-ary Pulse-Position Modulation (PPM) sequences with good correlation properties is investigated in three parts. Sequences with good autocorrelation property (GAP) are considered in the first part and sequences with good crosscorrelation property (GCP) are considered in the second part. In the third part, sequences that have both GAP and GCP are investigated. Theoretical bounds on sequences with GAP and GCP are presented and compared with computer results.

DEDICATION

To my mother Dr. Aye A. Tin-Sein, my aunt Dr. Mya M. Tin-Sein, my grandmother Dr. Khin Mya, and to the memory of my grandfather Dr. Tin-Sein, without whose love, sacrifice, and proper guidance, none of my goals could have been attained.

ACKNOWLEDGEMENTS

The author wishes to express his deep appreciation and sincere gratitude to Dr. C. N. Georghiades, his advisory committee chairman, for proper guidance, constructive criticism, encouragement, patience, and the time he has spent throughout this work. The author sincerely acknowledges his gratitude to Dr. J. D. Gibson, Dr. J. A. Fleming, and Dr. T. R. Fischer for introducing him to the field of communication and signal processing and their valuable advice; Dr. Al Boggess and Dr. P. Fred Dham for their tremendous help in Mathematics and Statistics, Dr. William Beasely and Professor Will Worley for their advice, and Mrs. Nancy Fletcher for her advice and help at Texas A&M University. The author wishes to thank Dr. Louise Canfield, the director of the University Honors Program, and Ms. Rebecca Furtado for their administrative support and genuine concern for the Universiy Undergraduate Fellows. The author also wishes to thank Dr. Sam Rodgers, Mr. John Duvall, and Mr. James Soch for their help and encouragement in Mathematics, and Mr. Homer Gifford and Mrs. Ann Faulkner for their help with English at Mountain View College. The author gratefully acknowledges his sincere appreciation to all his old teachers in Taunggyi, especially U Khin Maung, U Ko, U Phone Myint, U Kyaw Soe Myint, U Aung Min, U Shwe Maung, U Myat Mon Oo, U Chit Maung, U Soe Maung, U Kyaw Win Tin, U Myint Thu, Daw Khin Mu and Daw Sein Sein. The author also wishes to thank for the encouragement and help offered by his colleagues, especially Tricia Bohling, Karen Blake, Mahendra Advani, Pratiksh Parikh, Shaikh Haque, Khanh Nguyen, Art Flores, Maung Maung Myo, and Nyan Lin. Finally The author would like to thank Mrs. Roma Blaschke, Mr. Lester E. Blaschke, Jon Liljeblad, and Kjell Liljeblad for their friendship, love and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION AND MOTIVATION

Sets of sequences with one or both of the following properties have a large number of applications in system engineering:

I. Each sequence in the set is distinguishable from a time-shifted version of itself;

II. Each sequence in the set is distinguishable not only from every other sequence, but also from their time shifted versions.

The first property, which will be referred to as the "good" autocorrelation property (GAP), has important applications in a variety of areas such as ranging, radar, spread-spectrum, and synchronization [1,2]. The second, which will be referred to as the "good" crosscorrelation property (GCP), is important for applications such as multiple-terminal system identification, code-division multiple-access communications systems, simultaneous ranging of several targets, and matched-filter detection [2,3].

The motivation for this work stems from the previous work on Maximum-Likelihood (ML) synchronization and Minimum Mean-Square-Error (MMSE) synchronization for the optical direct-detection channel utilizing Onoff-Keying (OOK) [4]. However, this work is devoted to Q-ary Pulse-Position Modulation (PPM). Both PPM and OOK modulation formats are used in practice, but the former one is of current interest because it is proved that this choice is optimal [i.e. 5].

A. PULSE-POSITION MODULATION

Consider the time duration T of each PPM symbol with the length L. For Q-ary Pulse-Position Modulation, each baud is subdivided into Q slots, each of duration T'

Journal model is *IEEE Transactions on Communications.*

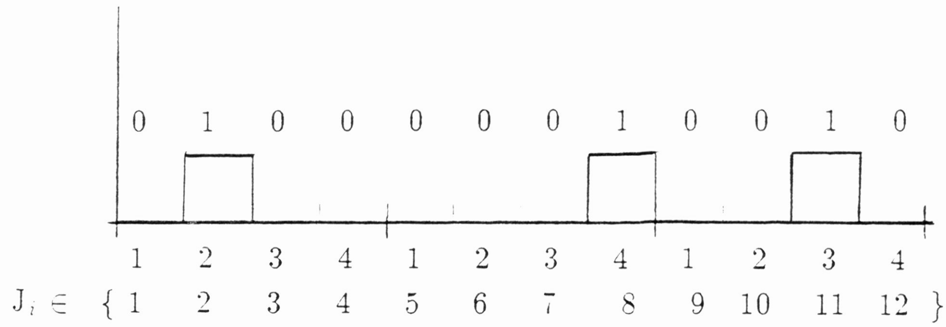
seconds, where $T' = \frac{T}{Q}$. In PPM, information is conveyed by the positioning of a single pulse in *one and only one* of Q subintervals (slots) dividing the symbol interval.

Let x_i, $i = 1, 2, ...L$, be an element of the set $\{1, 2, ..., Q\}$ of possible symbols defining the i^{th} symbol in a sequence. Let $J = \{j_i\}_{i=1}^{L}$ be the set of L indices denoting the position of the pulses in each of the L PPM symbols. It is clear that the x_i are related to the j_i by $j_i = (i-1)Q + x_i$, and that the set J for a given sequences completely describes the given PPM sequence. There are some advantages to denote PPM sequences by using j_i rather than x_i. For example, scrambled version j_i can still describe PPM sequences perfectly, whereas, scrambled version of x_i would give an error in determining the position of pulses in the original sequences.

An illustration for $Q = 4$ and $L = 3$ is given in Fig. 1. The time interval of T is then subdivided into four slots, each of duration $T' = T/4$ seconds. As we mentioned earlier, for every interval of T seconds, an optical pulse is placed in one and only one of the four slots and no pulse during the other seven slot times. Thus, each T-second interval contains 2 bits of information (since $2^2 = 4$). The energy content of the signal for an optical channel is directly proportional to the number photons observed in it. If each pulse contains an average energy of 2 detected photons, this particular PPM interval has power efficiency of 1 bit/detected photon. In general, Q-ary PPM contains X bits with a power efficiency of $\frac{X}{P}$ bits/detected photon, where X is define as $\log_2 Q$, and P is the number of photons (energy) that each pulse contains. It has been shown in [6] that optical signals can be used to transmit information over space channel with power efficiency up to 2.5 bits/detected photon.

B. SEQUENCES OF INTEREST

Maximum-Likelihood (ML) frame synchronizers for binary modulation have already



$J = \{J_i\}_{i=1}^{L}$ be the set of L index.

eg. Pulse-Position = (2,4,3)

J_i = (2,8,11)

Fig. 1: Example of PPM sequences for (Q, L)=(4,3).

been derived for an Additive White Gaussian Noise (AWGN) channel [7] and for OOK and PPM direct-detection optical communication systems [8,9]. Although the structure of the optimal synchronizers does not depend on the specific synchronization sequences used, the performance depends critically on the choice of "good sequences" [1]. Sequences that already exist are Gold Sequences [10], Kasami Sequences [11], Barker Sequences [12], and Neuman-Hofman Sequences[13].

In [7], Massey had used some of the above existing sequences and had evaluated the performance of ML synchronizer for AWGN channel with binary modulation. Georghiades [8,9] observed that there is no such "good sequences" available to satisfy the PPM constraint. In [8], it was assumed that symbol synchronization is present and in [9] it was assumed that only slot synchronization is present. PPM sequence design depends on whether symbol synchronization or only slot synchronization is present. For the former case, only correlations with the symbol shifts need to be considered since the evaluation of the likelihood statistics requires shifts by whole symbols. However, in the latter case, the evaluation of the likelihood statistic requires each slot shift (T' seconds), and hence the sequences must be designed to account for each slot shift.

CHAPTER II

MATHEMATICAL POINT OF VIEW

Consider a valid PPM sequence $\mathbf{S}$ and let Γ be the set of all PPM sequences for a given (Q, L) pair. There are a total of Q^L sequences in Γ. Define $S_k = 1$, when $k \in J = \{j_i\}_{i=1}^{L}$ for a given sequence $\mathbf{S}$, and $S_k = 0$ otherwise.

A. AUTOCORRELATION (Slot Synchronization Case)

The periodic autocorrelation of a given sequence $\mathbf{S}$ for each cyclic slot shift m, when only slot synchronization is present, is defined as:

$$A(\mathbf{S}, m) = \sum_{i=1}^{QL} S_{i \oplus m} S_i \qquad m = 1, ..., (QL - 1) \tag{1}$$

where $\oplus$ denotes the modulo-QL addition. For a given Q-ary PPM sequence, $m \in \Omega = \{1, 2, ..., (QL - 1)\}$ denotes the number of slot shifts of the sequences $\mathbf{S}$ and $j_i \in \{1, 2, ..., QL\}$, where the first slot in the sequences is slot 1, and the last is slot QL. Clearly, because of the PPM constraint, $j_k \geq j_i$ when $k \geq i$, with equality only when $k = i$. Equation (1) can be rewritten as

$$A(\mathbf{S}, m) = \sum_{i=1}^{L} S_{j_i \oplus m} S_{j_i} \qquad m = 1, ..., (QL - 1). \tag{2}$$

Note that each term of the right hand side of equation (2) satisfies the following property:

$$S_{j_i \oplus m} = 1 \qquad \text{if } j_i \oplus m \in J$$

$$S_{j_i \oplus m} = 0 \qquad \text{if } j_i \oplus m \notin J$$

$$S_{j_i} = 1 \qquad \text{always}$$

and consequently, equation (2) reduces to

$$A(\mathbf{S}, m) = \sum_{i=1}^{L} S_{j_i \oplus m}. \qquad m = 1, ..., (QL-1) \tag{3}$$

Furthermore, the peak-to-sidelobe autocorrelation distance (PTSAD) of a given sequence $\mathbf{S}$ for each cyclic slot shift m is given by:

$$\begin{aligned} D(\mathbf{S}, m) &= L - A(\mathbf{S}, m) \\ &= \sum_{i=1}^{L} (S_{j_i} - S_{j_i \oplus m}). \end{aligned} \tag{4}$$

For the slot synchronization case, if a sequence satisfies the following criterion:

$$D_{max}(Q, L) = \max_{\mathbf{S} \in \Gamma} D_{min}(\mathbf{S}), \tag{5}$$

where $D_{min}(\mathbf{S})$ is given by

$$D_{min}(\mathbf{S}) = \min_{m \in \Omega} D(\mathbf{S}, m), \tag{6}$$

it will be referred to as an *optimal sequence*. In [9], an optimal sequence was defined to be one that achieves a PTSAD of $(L-1)$. However, the definition of an optimal sequence for this work is more general and the sequences that achieve the maximum possible PTSAD, or satisfy the criterion given in (5) for a given (Q, L) pair, are optimal sequences.

The following theorem gives $D_{max}(Q, L)$ for the case when only slot synchronization is present [1].

Theorem 1: For a given (Q, L) pair, the maximum achievable PTSAD $D_{max}(Q, L)$ for the slot synchronization case is bounded according to

$$D_{max}(Q, L) \leq \frac{L^2(Q-1)}{(QL-1)}. \tag{7}$$

Let $\bar{D}$ be

$$\frac{L^2(Q-1)}{(QL-1)}, \tag{8}$$

which may or may not be an integer, but it is clear that $D_{max}(Q,L)$ in equation (7) is an integer. To compare the theoretical bound given in (7) with computer simulation results, $\bar{D}$ is replaced by $\lfloor D \rfloor$, the largest integer smaller than $\bar{D}$, since this results in a tighter bound. Consequently, equation (7) becomes

$$D_{max}(Q,L) \le \lfloor \frac{L^2(Q-1)}{(QL-1)} \rfloor. \tag{9}$$

At this point, it is worth pointing out some interesting implications of the above theorem. Suppose that PPM sequences with a constant $QL = C$ are desired such that the PTSAD is maximized. This situation may arise from the synchronizer complexity considerations when choices of Q is not critical. It is desirable to maximize $\bar{D}$ given in (8) assuming that the bound is tight. It is shown in [1] that a single maximum occurs between $L = 0$ and $L = C$ at $L = C/2$ and the corresponding achievable PTSAD for even C is

$$\bar{D}_{max} = \frac{(QL)^2}{4(QL-1)}. \tag{10}$$

If C is not even, then achievable PTSAD will be less than $\bar{D}_{max}$. However, C is always even in practice since Q is generally taken to be a power of two ($Q = 2^X$).

Another interesting implication of Theorem 1 is that a bound to the PTSAD per sequence symbol is

$$\frac{1}{L}\bar{D} = \frac{L(Q-1)}{(QL-1)} \tag{11}$$

for long sequences (large L). As the limit of L approach infinity,

$$\lim_{L\to\infty} \frac{1}{L}\bar{D} = \frac{(Q-1)}{Q} \tag{12}$$

which implies that $\bar{D}$, and hence $D_{max}(Q, L)$ when the bound given in theorem 1 is tight, increase linearly with L for long sequences, at a rate less than unity.

B. AUTOCORRELATION (Symbol Synchronization Case)

The problem of designing PPM sequences with GAP, when symbol synchronization is present, have been formulated as follows. At a first glance, it may seem to be a special case of the situation when slot synchronization is present. However, a closer look shows that optimal sequences, derived under the presence of slot synchronization only, are not necessarily optimal when symbol synchronization is present and vice versa [1]. Autocorrelation of a given sequence $\mathbf{S}$, when symbol synchronization present is developed parallel to slot synchronization case. Since the symbol shifts correspond to shifts by integer multiples of Q slots at a time, the periodic autocorrelation of a given sequence $\mathbf{S}$, when symbol synchronization is present, is given by

$$A(\mathbf{S}, m') = \sum_{i=1}^{L} S_{j_i \oplus m'} \ , \tag{13}$$

where $m' \in \omega' = \{Q, 2Q, ..., (L-1)Q\}$, and S_{j_i} is as previously defined.

Furthermore, the PTSAD of a given sequences $\mathbf{S}$ for each cyclic symbol shift m' is given by

$$\begin{aligned} d(\mathbf{S}, m') &= L - A(\mathbf{S}, m') \\ &= \sum_{i=1}^{L} (S_{j_i} - S_{j_i \oplus m'}). \end{aligned} \tag{14}$$

A different notation d for PTSAD is used here to distinguish between two cases when only slot synchronization is present and when symbol synchronization is present. Similar to slot synchronization, optimal sequence for the symbol synchronization case is defined. A sequence is said to be optimal if it satisfies the following criterion:

$$d_{max}(Q,L) = \max_{\mathbf{S}\in\Gamma} d_{min}(\mathbf{S}), \tag{15}$$

where $d_{min}(\mathbf{S})$ is given by

$$d_{min}(\mathbf{S}) = \min_{m'\in\omega} d(\mathbf{S}, m'). \tag{16}$$

The following theorem gives an achievable PTSAD $d_{max}(Q,L)$ when symbol synchronization is present [1].

<u>Theorem 2</u>: For a given (Q,L) pair, the maximum achievable PTSAD $d_{max}(Q,L)$ for the symbol synchronization case is bounded by

a)

$$d_{max}(Q,L) \leq \lfloor \frac{L^2(Q-1)}{Q(L-1)} \rfloor, \tag{17}$$

if(L/Q) is an integer, and by

b)

$$d_{max}(Q,L) \leq \lfloor L - \frac{u_L[2L - Q(u_L+1)]}{(L-1)} \rfloor, \tag{18}$$

if (L/Q) is not an integer, where $u_L = \lfloor L/Q \rfloor$, the largest integer smaller than (L/Q).

C. CROSSCORRELATION (Slot Synchronization Case)

Let $J = \{J_i\}_{i=1}^{L}$ and $K = \{k_i\}_{i=1}^{L}$ be a pair of sets, each of L indices, denoting the position of the sequence $\mathbf{S^j}$ and $\mathbf{S^k}$ respectively. It is clear that $j_i \in \{1,2,...,QL\}$ and $k_i \in \{1,2,...,QL\}$, where the first slot in each sequence is 1 and the last is QL. For a given Q-ary PPM sequence, $m \in \Omega = \{1,2,...,(QL-1)\}$ denotes the number of slot shifts of the sequences $\mathbf{S^k}$.

The crosscorrelation of given sequences $\mathbf{S^j}$ and $\mathbf{S^k}$ for each cyclic shift m, when slot synchronization is present, is defined as:

$$C(\mathbf{S^j},\mathbf{S^k},m) = \sum_{i=1}^{L} S^j_{j_i} S^k_{k_i \oplus m} \qquad m = 1, ..., (QL-1), \tag{19}$$

where $\oplus$ denotes the modulo-QL addition. Note that each term on the right hand side of equation (19) satisfies the property:

$$S^k_{k_i \oplus m} = 1 \qquad \text{if } k_i \oplus m \in K$$

$$S^k_{k_i \oplus m} = 0 \qquad \text{if } k_i \oplus m \notin K$$

$$S^j_{j_i} = 1 \qquad \text{always},$$

and consequently, equation (19) reduces to

$$C(\mathbf{S^j},\mathbf{S^k},m) = \sum_{i=1}^{L} S^j_{k_i \oplus m} \qquad m = 1, ..., (QL-1), \tag{20}$$

where

$$S^j_{k_i \oplus m} = 1 \qquad \text{if } k_i \oplus m \in J,$$

$$S^k_{k_i \oplus m} = 0 \qquad \text{if } k_i \oplus m \notin J.$$

Furthermore, for each cyclic slot shift m, the peak-to-sidelobe crosscorrelation distance (PTSCD) of sequences $\mathbf{S^j}$ and $\mathbf{S^k}$ is given by

$$\begin{aligned} D'(\mathbf{S^j},\mathbf{S^k},m) &= L - C(\mathbf{S^j},\mathbf{S^k},m) \\ &= \sum_{i=1}^{L} (S^j_{j_i} - S^j_{k_i \oplus m}). \end{aligned} \tag{21}$$

For the slot synchronization case, if a sequence satisfies the following criterion:

$$D'_{max}(Q,L) = \max_{\mathbf{S} \in \Gamma} D'_{min}(\mathbf{S^j},\mathbf{S^k}), \tag{22}$$

where $D'_{min}(\mathbf{S^j}, \mathbf{S^k})$ is given by

$$D'_{min}(\mathbf{S^j}, \mathbf{S^k}) = \min_{m \in \Omega} D'(\mathbf{S^j}, \mathbf{S^k}, m), \tag{23}$$

it will be referred to as an *optimal sequence.*

Theorem 3: For a given (Q, L) pair, the maximum achievable peak-to-sidelobe distance $D_{max}(Q, L)$ for crosscorrelation case is bounded according to

$$D_{max}(Q, L) \leq L(\frac{Q-1}{Q}). \tag{24}$$

D. CODE RATE

Code rate, denoted by CR is defined to be the ratio of the number of the transmitted bits using "good" sequences and the number of bits a system could have transmitted using all sequences in Γ for given (Q, L). This can be express as

$$CR = \frac{\log_2(\# \text{ of good sequences})}{L \ \log_2(Q)}. \tag{25}$$

CHAPTER III

ALGORITHMS

A. AUTOCORRELATION (Slot Synchronization Case)

Let Δ_1 be the set of all differences $|j_i - j_k|$ and Δ_2 be $(QL - |j_i - j_k|)$ for all $i \neq k$ $k = 1, 2, ..., L$. Define Δ to be $\Delta_1 \cup \Delta_2$. In the sequel, Δ will be referred to as the *difference set* of a given sequence $\mathbf{S}$. It is clear that all possible elements of Δ are the elements of $\Omega = \{1, 2, ..., (QL - 1)\}$, i.e. $\Delta \subset \Omega$.

Lemma 1: If $m \in \Omega$ is repeated n times in Δ, $(1 \leq n \leq L)$, for some PPM sequences $\mathbf{S}$, then $D(\mathbf{S}, m) = (L - n)$.

The proof of this Lemma 1 was shown easily in [9]. This Lemma implies that the minimum of $D(\mathbf{S}, m)$ over m corresponds to the value of m that repeats most in Δ, i.e., m has the largest n. This prevent the large numbers of calculations that would have been required to find $D(\mathbf{S})$ using (4) and to minimize over m. An illustration of the use of this Lemma is shown in Fig. 2 for (Q, L)=(4,3).

The following algorithm investigates the sequences with GAP when only slot synchronization is present. For a given (Q, L) pair, this algorithm generates a set of optimal sequences, the achievable maximum PTSAD, and the total number of optimal sequences.

GAPslot Algorithm

Step 1. Generate PPM sequence $\mathbf{S}$ for given (Q, L).

Set $D^-_{min}(\mathbf{S}) = 0$.

Step 2. Find the difference set Δ.

Step 3. Compute $D_{min}(\mathbf{S}) = L - n$ using Lemma 1.

$$(Q,L) = (4,3)$$
$$QL = 12$$
$$Pulse - Position = (2,4,3)$$
$$j_i = (2,8,11)$$
$$|j_i - j_k| = 6,9,3 \qquad i \neq k$$
$$(QL - |j_i - j_k|) = 6,3,9$$
$$\Delta = \{6,9,3,6,3,9\}$$
$$n = 2$$

$A_s(m) = 0$ $m = 1$	0100 0001 0010 0010 0000 1001
$A_s(m) = 0$ $m = 2$	0100 0001 0010 1001 0000 0100
$A_s(m) = 2$ $m = 3$	0100 0001 0010 0100 1000 0010
$A_s(m) = 0$ $m = 4$	0100 0001 0010 0010 0100 0001
$A_s(m) = 0$ $m = 5$	0100 0001 0010 1001 0010 0000
$A_s(m) = 2$ $m = 6$	0100 0001 0010 0100 1001 0000
$A_s(m) = 0$ $m = 7$	0100 0001 0010 0010 0100 1000
$A_s(m) = 0$ $m = 8$	0100 0001 0010 0001 0010 0100
$A_s(m) = 2$ $m = 9$	0100 0001 0010 0000 1001 0010
$A_s(m) = 0$ $m = 10$	0100 0001 0010 0000 0100 1001
$A_s(m) = 0$ $m = 11$	0100 0001 0010 1000 0010 0100

Fig. 2: Example for the Use of Lemma 1.

Step 4. If $D_{min}(\mathbf{S}) = D^{-}_{min}(\mathbf{S})$,

Save the Sequence.

Step 5. If $D_{min}(\mathbf{S}) > D^{-}_{min}(\mathbf{S})$,

$D^{-}_{min}(\mathbf{S}) = D_{min}(\mathbf{S})$.

Delete the previously saved sequences and save the new sequence.

Step 6. Generate new PPM sequence **S** for given (Q, L).

Go to Step 2 until all sequences in Γ are checked.

The following Lemma and its proof, given in [1], suggest an alternate algorithm to find sequences with GAP when slot synchronization is present.

Lemma 2: Let **S** be a PPM sequence with some $D_{min}(\mathbf{S})$, and $\mathbf{S}^l$ be a cyclic slot shift version of **S** by l slots. Then, if $\mathbf{S}^l$ satisfies the PPM constraint, it is $D_{min}(\mathbf{S}) = D_{min}(\mathbf{S}^l)$

This Lemma implies that **S** is an optimal sequence iff $\mathbf{S}^l$ is an optimal sequence, which suggests to find an algorithm to search for a shifted version $\mathbf{S}^l$ of the original sequence **S** for any $l \in \Omega = \{1, 2, ..., (QL-1)\}$ in Γ. This would save a large number of calculations since it is only necessary to find $D_{min}(\mathbf{S})$ for the sequence **S** and $\mathbf{S}^l$ for any $l \in \Omega$ having the same minimum PTSAD, $D_{min}(\mathbf{S}) = D_{min}(\mathbf{S}^l)$, by Lemma 2.

Lemma 3: Let $D(\mathbf{S}, m)$ be the PTSAD of a given PPM sequence **S** with some cyclic slot shift m. Then $D(\mathbf{S}, m) = D(\mathbf{S}, QL-m)$

Proof: Let some cyclic slot shift, $m \in \Omega$, repeat n times in Δ, where $\Delta = \Delta_1 + \Delta_2$. Then

$$\sum_{i=1}^{L} S_{j_i \oplus m} = n = n_1 + n_2, \tag{26}$$

where n_1 and n_2 are the numbers of times m repeats in Δ_1 and Δ_2, respectively. Clearly, m repeats n_1 times in Δ_1 iff $QL - m$ repeats n_1 times in Δ_2. Likewise, m

repeats n_2 times in Δ_2 iff $QL - m$ repeats n_2 times in Δ_1. Therefore, m repeats n times in Δ, and $(QL - m)$ repeats n times in Δ, and consequently,

$$\sum_{i=1}^{L} S_{j_i \oplus m} = \sum_{i=1}^{L} S_{j_i \oplus (QL-m)} \tag{27}$$

$$D(\mathbf{S}, m) = D(\mathbf{S}, QL - m). \tag{28}$$

This proves the Lemma 3.

B. AUTOCORRELATION (Symbol Synchronization Case)

Notice the symbol shift $m' \in \omega' = \{Q, 2Q, ..., (L-1)Q\}$, and slot shift $m \in \Omega = \{1, 2, ..., (QL-1)\}$. It is clear that $\Delta \subset \Omega$, $\omega' \subset \Omega$, and all the differences must be in the set of ω' when symbol synchronization is present. Let δ', the difference set for symbol synchronization case, be the intersection between Δ and ω'.

Lemma 4: If $m' \in \omega'$ is repeated n times in δ', $(1 \leq n' \leq L)$, for some sequences $\mathbf{S}$, then $d(\mathbf{S}, m') = (L - n')$.

The following algorithm investigates the sequences with GAP when symbol synchronization is present. This algorithm gives a set of optimal sequences, the maximum achievable PTSAD, and the total number of optimal sequence for the given (Q, L) pair.

GAPsymbol Algorithm

Step 1. Generate PPM sequence $\mathbf{S}$ for given (Q, L).

Set $d^{-}_{min}(\mathbf{S}) = 0$.

Step 2. Find the difference set δ'.

Step 3. Compute $d_{min}(\mathbf{S}) = L - n'$ using Lemma 4.

Step 4. If $d_{min}(\mathbf{S}) = d^{-}_{min}(\mathbf{S})$, Save the sequence.

Step 5. If $d_{min}(\mathbf{S}) > d^{-}_{min}(\mathbf{S})$,

$d^{-}_{min}(\mathbf{S}) = d_{min}(\mathbf{S})$.

Delete the previously saved sequences and save the new sequence.

Step 6. Generate new PPM sequence $\mathbf{S}$ for given (Q, L).

Go to Step 2 until all sequences in Γ are checked.

A special case of Lemma 2 is when l is an integer multiple of PPM alphabet size Q. The implication of Lemma 2 for the symbol synchronization case is that $\mathbf{S}$ is an optimal sequence iff $\mathbf{S}^l$ is an optimal sequence for any $l \in \omega' = \{Q, 2Q, ..., (L-1)Q\}$. Again, it is worth pointing out that only $d_{min}(\mathbf{S})$ need to be found for sequence $\mathbf{S}$ and $\mathbf{S}^l$ for any $l \in \omega'$ if the algorithm to find $\mathbf{S}^{iQ}$, for any integer i, is available. Lemma 2 further implies that the total number of optimal sequences is divisible by L since there are $(L-1)$ symbols shifts that yield distinct sequences.

Lemma 5: Let $D(\mathbf{S}, m')$ be the PTSAD of a given PPM sequence $\mathbf{S}$ with some cyclic symbol shift m'. Then $D(\mathbf{S}, m') = D(\mathbf{S}, QL - m')$

Proof: The proof follows easily from Lemma 3. Let some cyclic symbol shift, $m' \in \omega'$, repeats n' times in δ', where $\delta' = \delta'_1 + \delta'_2$. Then

$$\sum_{i=1}^{L} S_{j_i \oplus m'} = n' = n'_1 + n'_2, \tag{29}$$

where n'_1 and n'_2 are the numbers of times m' repeats in δ'_1 and Δ'_2 respectively. Clearly, m' repeats n'_1 times in δ'_1 iff $QL - m'$ repeats n'_1 times in δ'_2. Likewise, m' repeats n'_2 times in δ_2 iff $QL - m'$ repeats n'_2 times in δ'_1. Therefore, m' repeats n' times in δ', and $(QL - m')$ repeats n' times in δ', and consequently,

$$\sum_{i=1}^{L} S_{j_i \oplus m'} = \sum_{i=1}^{L} S_{j_i \oplus (QL-m')} \tag{30}$$

and

$$D(\mathbf{S}, m') = D(\mathbf{S}, QL - m). \tag{31}$$

This proved the Lemma 5.

C. CROSSCORRELATION (Slot Synchronization Case)

Define Δ', the difference for slot synchronization case, of a given sequence $\mathbf{S^j}$ and $\mathbf{S^k}$ to be the set $\{j_i - k_l\}$ if $(j_i - k_l)$ is positive, and $\{QL + (j_i - k_l)\}$ if $(j_i - k_l)$ is negative, for all $i = 1, 2, ..., L$ and $k = 1, 2, ..., L$.

Lemma 6: If $m \in \Omega$ is repeated n times in Δ', $(1 \leq n \leq L)$, for PPM sequence $\mathbf{S^a}$ and $\mathbf{S^b}$, then $D'(\mathbf{S^a}, \mathbf{S^b}, m) = (L - n)$.

The following algorithm is to investigate a set of sequences with GCP when slot synchronization is present. This algorithm results in a set of good sequences and the number of good sequences for given PTSCD and (Q, L) pair.

GCPslot Algorithm

Step 1. Generate PPM sequence $\mathbf{S}$ for given (Q, L) and save it.
Set $D'^{-}_{min}(\mathbf{S^j}, \mathbf{S^k})$.

Step 2. Generate new PPM sequence $\mathbf{S}$ for given (Q, L).

Step 3. Find the difference set Δ' between new sequence $\mathbf{S^j}$ and previously saved sequence $\mathbf{S^k}$.

Step 4. Compute $D'_{min}(\mathbf{S^j}, \mathbf{S^k}) = L - n$ using Lemma 6.

Step 5. If $D'_{min}(\mathbf{S^j}, \mathbf{S^k}) < D'^{-}_{min}(\mathbf{S^j}, \mathbf{S^k})$,
Go to Step 2 until all sequences in Γ are checked.

Step 6. Repeat step 3, 4 and 5 with all the previously saved sequences.

Save the sequence

Step 7. Go to step 2 until all sequences in Γ are checked.

CHAPTER IV

RESULTS

Algorithms decribed in chapter III have been implemented using VAX FORTRAN on the Texas A&M University Engineering Computer Services (ECS) VAX 8800/8650 Cluster. The main algorithms GAPslot, GAPsymbol, GCPslot, and GACPslot are listed Appendix A. The subroutines necessary for these main programs are listed in Appendix B.

A. GOOD AUTOCORRELATION SEQUENCES

Sequences with GAP are generated by exhaustive computer search. Periodic autocorrelation of a given sequence **S** for some cyclic shift is computed as the sum over all slots of the product of the sequence **S** and shifted **S** at each slot.

Typical autocorrelation function of any cyclic slot shift m, when only slot synchronization is present, for (Q, L)=(5,6) and (Q, L)=(5,5) are shown in Fig. 3 and Fig. 4, respectively. The peak of this function is the autocorrelation at m=0, and there is a corresponding sidelobe for each cyclic slot shift m. Sidelobe for some cyclic slot shift m is simply the autocorrelation at that particular shift. The PTSAD for some cyclic slot shift m, when only slot synchronization is present, is the difference between the autocorrelation functions at zero shift and that particular slot shift m.

For (Q, L)=(5,6), minimum PTSAD achieved by the computer is (peak – maximum sidelobe) = (5 – 1) = 4, and theoretical bound is

$$\lfloor\frac{L^2(Q-1)}{(QL-1)}\rfloor = \lfloor\frac{6^2(5-1)}{(30-1)}\rfloor = 4.$$

For (Q, L)=(5,5), minimum PTSAD achieved by the computer is (peak – maximum sidelobe = (5 – 1) = 4, and theoretical bound is

$$\lfloor \frac{L^2(Q-1)}{(QL-1)} \rfloor = \lfloor \frac{5^2(5-1)}{(25-1)} \rfloor = 4.$$

The above two cases illustrate the fact that the computer results achieved the theoretical bounds, given by Theorem 1, with equality. Furthermore, the autocorrelation function is symmetric at $\pm QL/2$ in Fig. 3 and Fig. 4, which agrees with $D(\mathbf{S}, m) = D(\mathbf{S}, QL-m)$, given in Lemma 3. For various (Q, L) pair, Table I compares the theoretical bound given in Theorem 1 with computer results for the case when only slot synchronization is present.

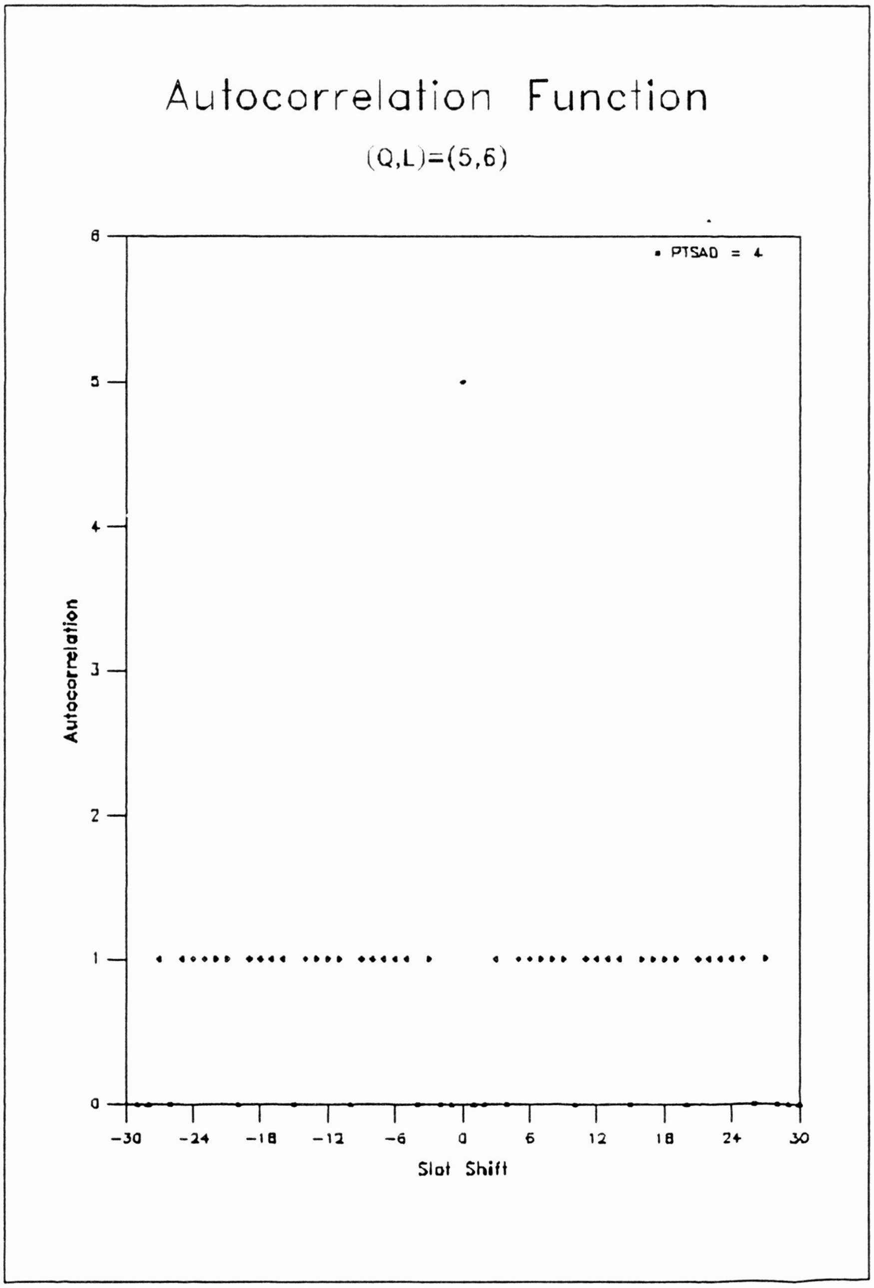


Fig. 3: Typical Autocorrelation Function Vs. Slot Shift for (Q, L)=(5,6).

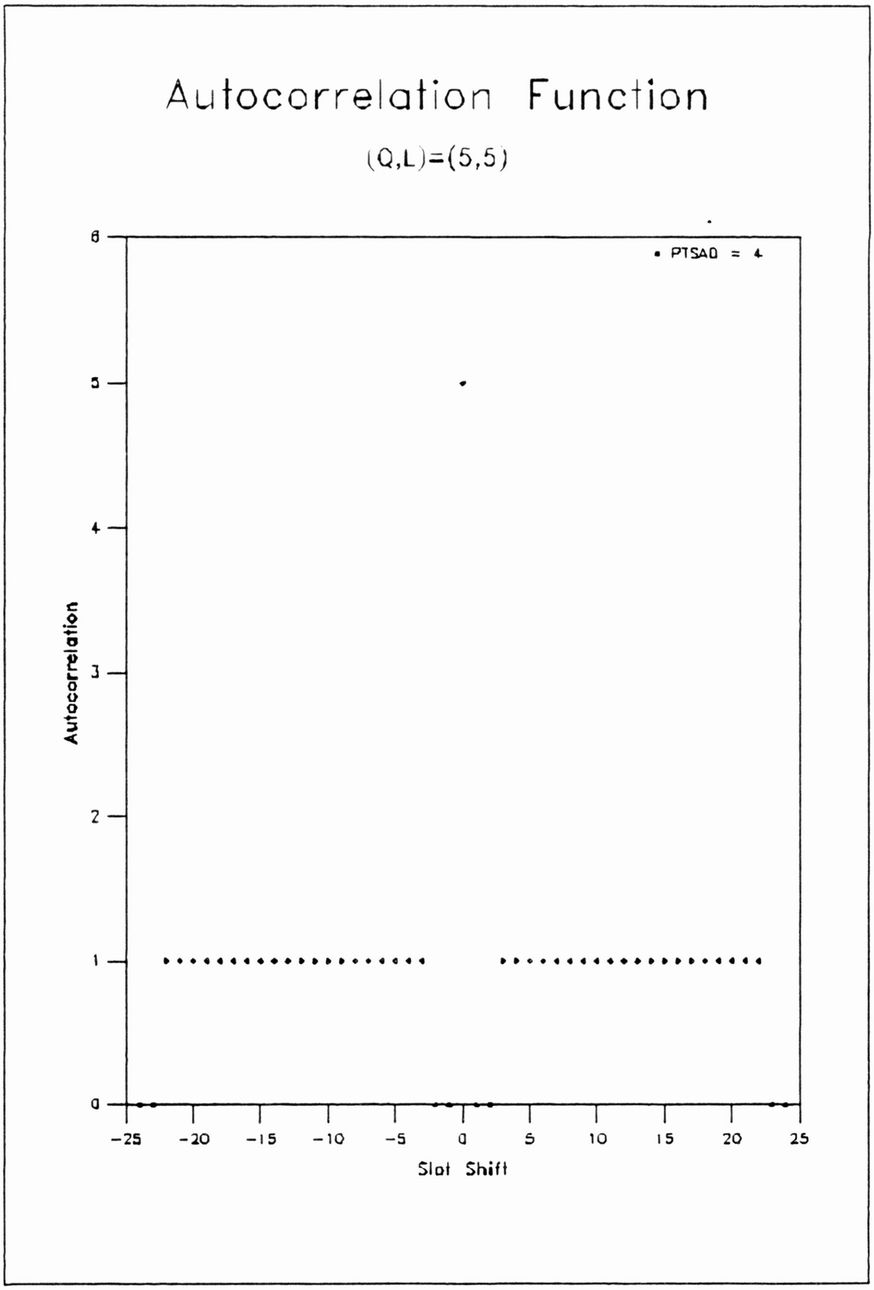


Fig. 4: Typical Autocorrelation Function Vs. Slot Shift for (Q, L)=(5,5).

TABLE I
Comparative Analysis of PTSAD for Slot Synchronization Case between Computer Results and Theroretical Bounds

(Q,L)	Computer Results $D_{max}(Q,L)$	Theoretical Bounds	# of Sequences with GAP
(4,4)	3	3	8
(4,5)	3	3	5400
(4,6)	4	4	516
(4,7)	5	5	140
(4,8)	5	6	12688
(4,9)	6	6	10620
(4,10)	7	7	2060
(8,3)	2	2	348
(8,4)	3	3	1328
(8,5)	4	4	2760
(8,6)	5	5	1176
(8,7)	6	6	126
(8,8)	6	7	2861312
(8,9)	7	7	6437574
(2,2)	1	1	2
(3,3)	2	2	18
(4,4)	3	3	8
(5,5)	4	4	40
(6,6)	4	5	20814
(7,7)	5	6	254709
(8,8)	6	7	2861312
(6,4)	3	3	296
(8,7)	6	6	126
(15,2)	1	1	210
(10,3)	2	2	720
(6,5)	4	4	180
(3,10)	6	6	1840
(2,15)	7	7	60

Typical autocorrelation function of any cyclic symbol m', when only symbol synchronization is present, is shown in Fig. 5 and Fig. 6 for (Q,L)=(4,6) and (Q,L)=(4,10) respectively. The peak of this function is the autocorrelation at m'=0, and there is a corresponding sidelobe for each cyclic symbol shift m'. The Sidelobe, for some cyclic shift m, is simply the autocorrelation at that particular shift. The PTSAD for some cyclic symbol shift m', when only slot synchronization is present, is the difference between the autocorrelation functions at zero shift, and that particular symbol shift m'.

For (Q,L)=(4,6), minimum PTSAD achieved by the computer is (peak – maximum sidelobe) = (6 – 1) = 5, and theoretical bound is

$$\lfloor L - \frac{u_L[2L - Q(u_L+1)]}{(L-1)} \rfloor = \lfloor 6 - \frac{1[12 - 4(1+1)]}{(6-1)} \rfloor = 5.$$

For (Q,L)=(4,10), minimum PTSAD achieved by the computer is (peak – maximum sidelobe) = (10 – 2) = 8, and theoretical bound is

$$\lfloor L - \frac{u_L[2L - Q(u_L+1)]}{(L-1)} \rfloor = \lfloor 10 - \frac{2[20 - 4(2+1)]}{(10-1)} \rfloor = 8.$$

This shows that the computer results achieved theoretical bounds given in Theorem 2 with equality. Furthermore, the autocorrelation function is symmetric at $\pm QL/2$ which agrees with the fact that $D(\mathbf{S}, m') = D(\mathbf{S}, QL - m')$, which is given in Lemma 5. For various (Q,L) pairs, the theoretical bound given in Theorem 2 are compared with the computer results in Table II, for the case when PPM symbol synchronization is present.

It is clear from Table I and Table II, that for most of the (Q,L) search, the theoretical bounds given in Theorem 1 and 2 are achieved with equality.

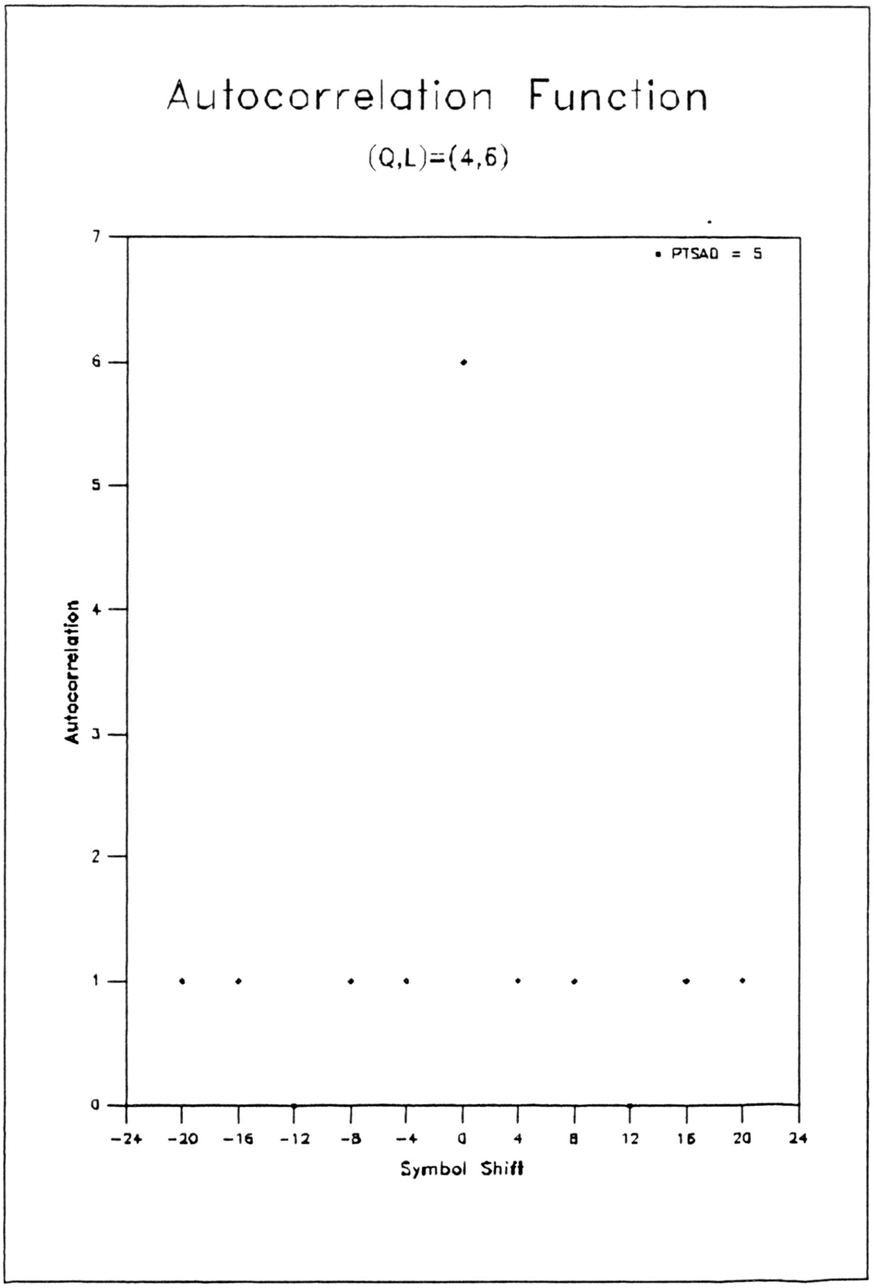


Fig. 5: Typical Autocorrelation Function Vs. Symbol Shift for (Q, L)=(4,6).

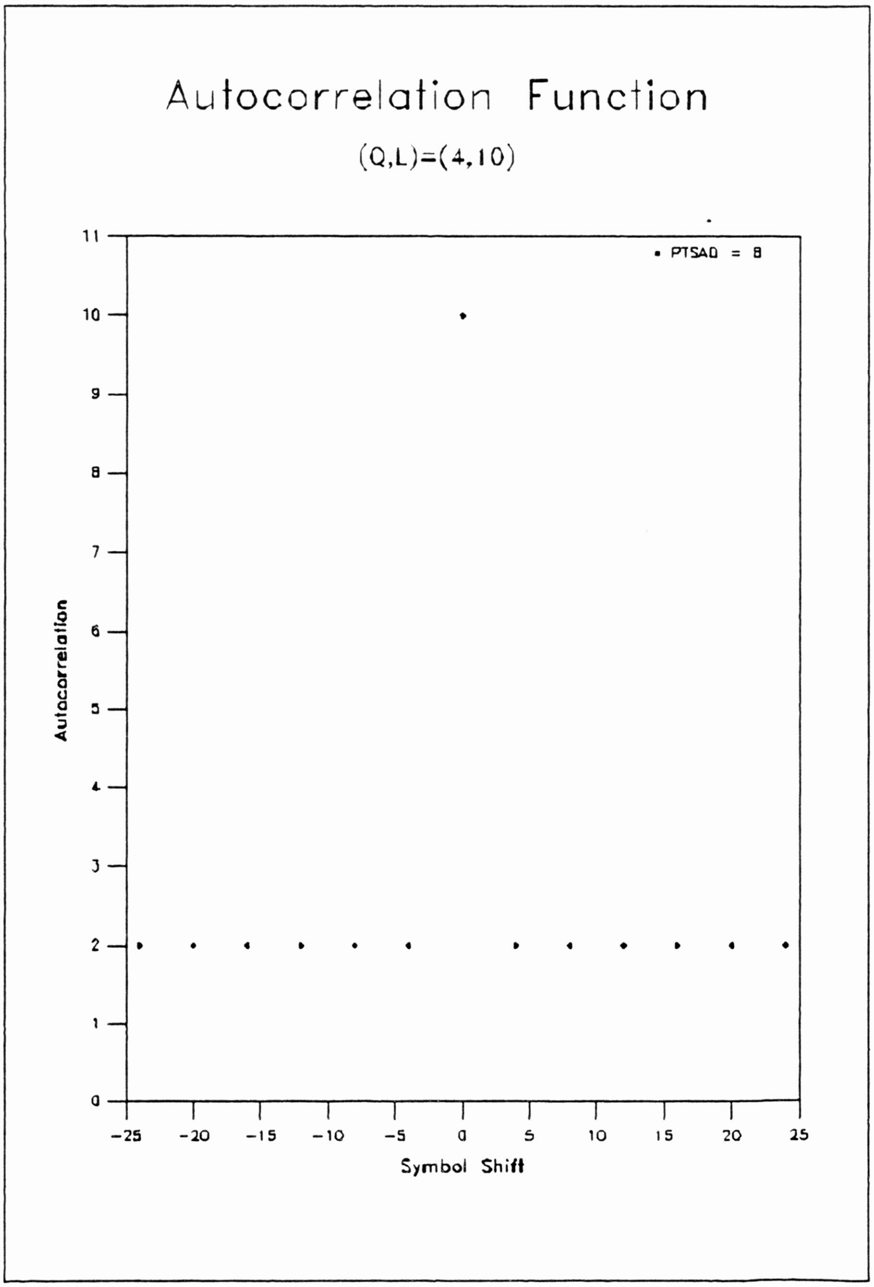


Fig. 6: Typical Autocorrelation Function Vs. Symbol Shift for (Q, L)=(4,10).

TABLE II
Comparative Analysis of PTSAD for Symbol Synchronization Case between Computer Results and Theroretical Bounds

(Q,L)	Computer Results $d_{max}(Q,L)$	Theoretical Bounds	# of Sequences with GAP
(4,2)	2	2	12
(4,3)	3	3	24
(4,5)	4	4	360
(4,6)	5	5	288
(6,2)	2	2	30
(6,3)	3	3	120
(6,4)	4	4	360
(6,5)	5	5	720
(6,7)	6	6	78120
(8,2)	2	2	56
(8,3)	3	3	336
(8,4)	4	4	1680
(8,5)	5	5	6720
(8,6)	6	6	20160
(2,2)	2	2	2
(3,3)	3	3	6
(4,4)	4	4	24
(5,5)	5	5	120
(6,6)	6	6	720
(2,4)	2	2	12
(2,6)	2	3	54
(2,10)	4	5	590
(2,15)	8	8	60
(3,4)	3	3	24
(3,7)	5	5	294
(4,10)	8	8	12000
(5,6)	5	5	2880

B. GOOD CROSSCORRELATION SEQUENCES

Periodic crosscorrelation of two given sequences $\mathbf{S^j}$ and $\mathbf{S^k}$, for some cyclic shift, is computed as the sum over all slots of the product of sequence $\mathbf{S^j}$ and shifted $\mathbf{S^k}$ at each slot.

Typical crosscorrelation function of any cyclic slot shift m for (Q, L)=(5,5), when only slot synchronization is present, is shown in Fig. 7. The peak of this function is defined to be the length of the sequences L. The sidelobe, for some cyclic slot shift m, is simply the crosscorrelation at that particular shift. Unlike the autocorrelation case, the crosscorrelation function at $m = 0$ is also considered to be the sidelobe for zero shift. The PTSCD for some cyclic slot shift m, when only slot synchronization is present, is the difference between L and the crosscorrelation functions at that particular slot shift m.

For (Q, L)=(5,5), minimum PTSCD achieved by the computer is (L – maximum sidelobe) = (5 – 1) = 4, and theoretical bound is

$$\lfloor L\left(\frac{Q-1}{Q}\right)\rfloor = \lfloor 5\left(\frac{5-1}{5}\right)\rfloor = 4.$$

This illustrates that the computer results achieved the theoretical bounds given by Theorem 3 with equality.

Comparison between the PTSCD achieved by a computer search and theoretical bound given by Theorem 3 were made in Table III. This table is for the case when only slot synchronization is present. Although the case with symbol synchronization is not presented here, it can be obtained by using the same algorithm (GCPslot), with the exception that m is a symbol shift.

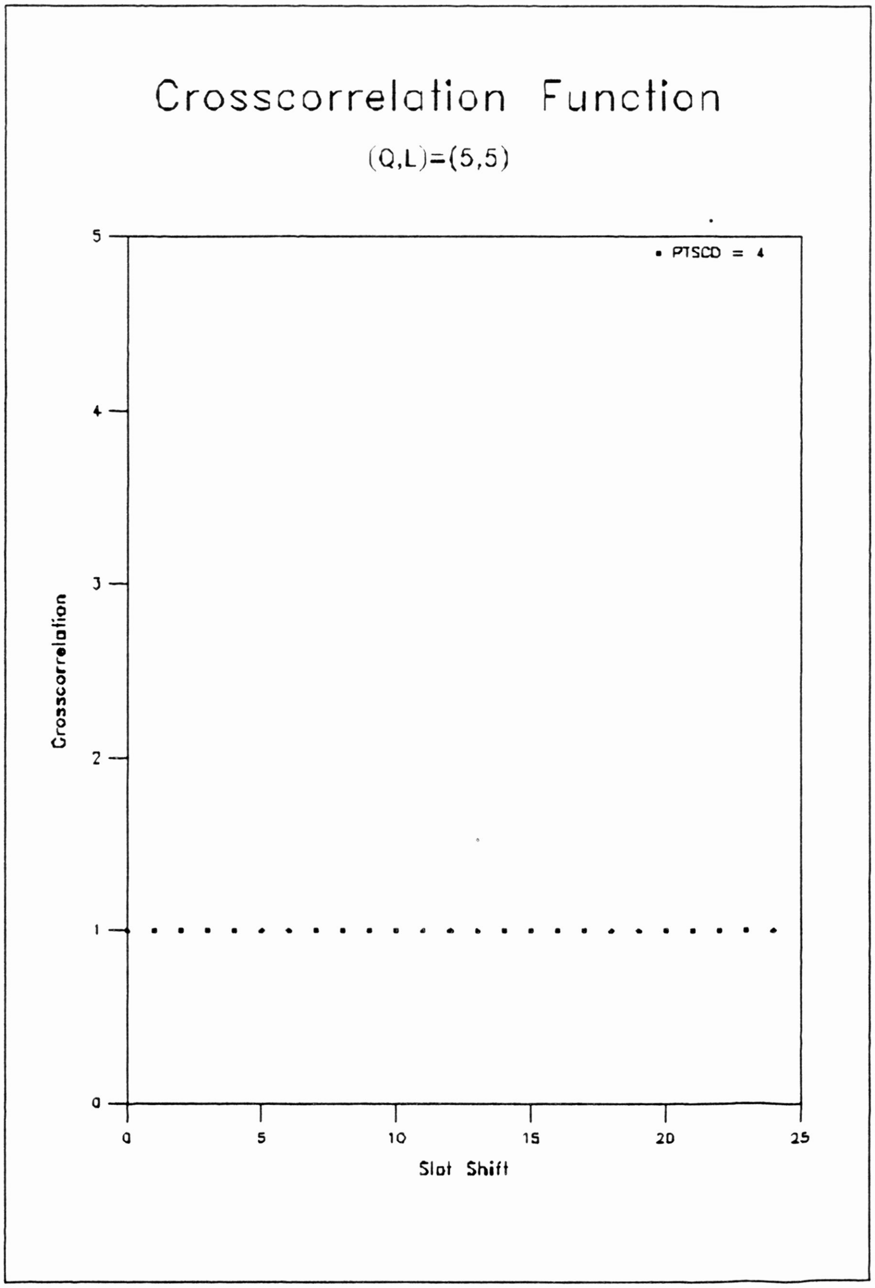


Fig. 7: Typical Crosscorrelation Function Vs. Slot Shift for (Q, L)=(5,5).

Notice these two sequences have only GCP.

TABLE III
Comparative Analysis of PTSCD for Slot Synchronization Case between Computer Results and Theroretical Bounds

(Q,L)	Computer Results $D'_{max}(Q,L)$	Theoretical Bounds	# of Sequences with GCP
(4,2)	1	1	4
(4,3)	2	2	3
(4,5)	3	3	3
(4,6)	4	4	3
(4,7)	5	5	2
(4,8)	6	6	2
(6,2)	1	1	6
(6,3)	2	2	3
(6,4)	3	3	2
(6,5)	4	4	3
(6,7)	5	5	4
(6,8)	6	6	4
(8,2)	1	1	8
(8,3)	2	2	4
(8,4)	3	3	4
(8,5)	4	4	4
(8,6)	5	5	2
(8,7)	6	6	3
(2,2)	1	1	2
(3,3)	2	2	2
(4,4)	3	3	2
(5,5)	4	4	2
(6,6)	5	5	2
(7,7)	6	6	2
(8,8)	7	7	2
(2,10)	5	5	2
(4,10)	7	7	2

Fig. 8 shows the crosscorrelation function of any given slot shift for (Q, L)=(5,5). These two sequences have both GAP and GCP. As expected, the sidelobes of this crosscorrelation function are higher than the cases where the two sequences have only GCP. This is illustrated in Fig. 7. Note that there is a trade-off between GAP and GCP.

Typical code rate, as a function of PTSCD, is shown in Fig. 9 for (Q, L)=(4,10). Notice that code rate decreases as PTSCD increases. Fig. 10 shows the code rate of sequences that have both GAP and GCP as a function of PTSAD and PTSCD, for (Q, L)=(4,10). Notice the trade-offs between the code rate and PTSAD as well as the code rate and PTSCD conform with our expectations.

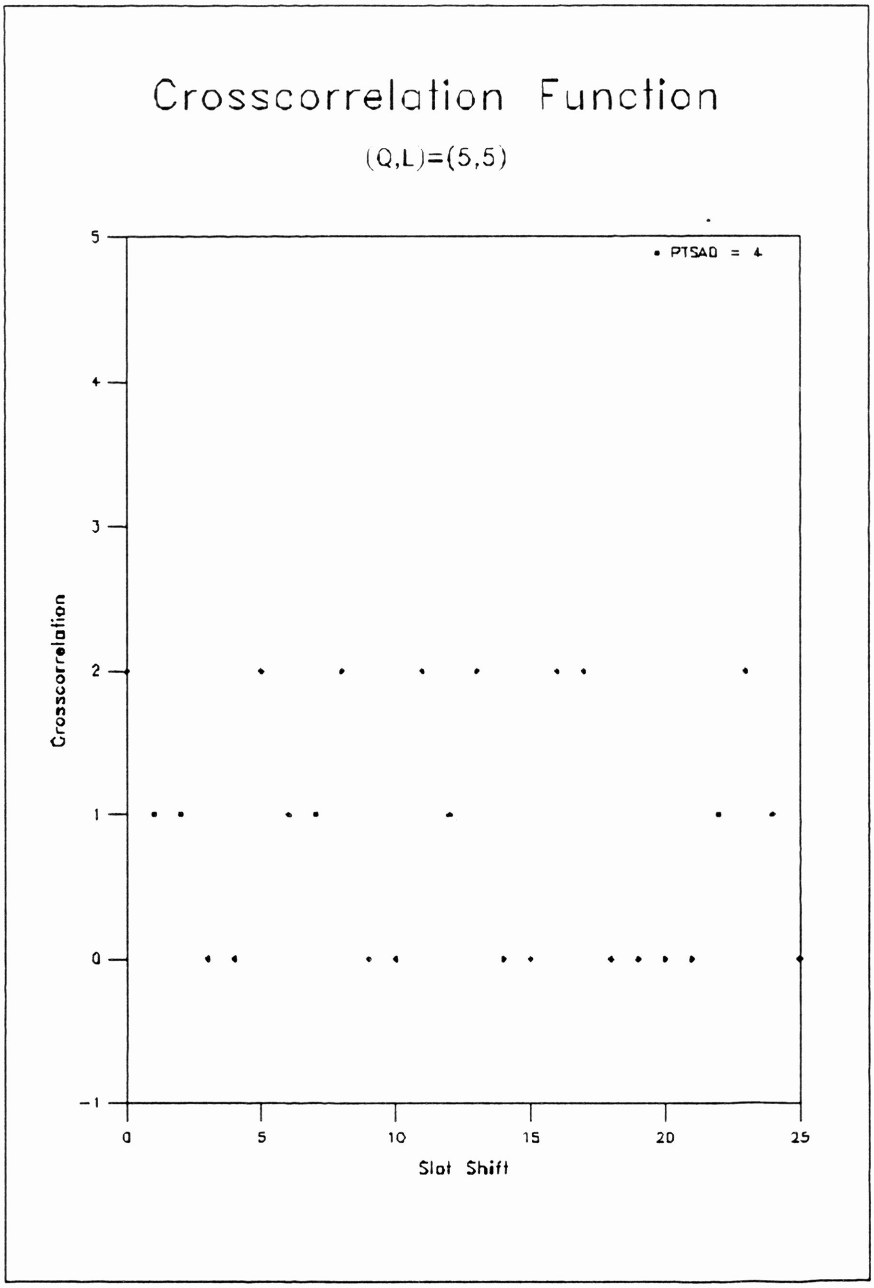


Fig. 8: Typical Crosscorrelation Function Vs. Slot Shift for (Q,L)=(5,5). Notice these this two sequences have both GAP as well as GCP.

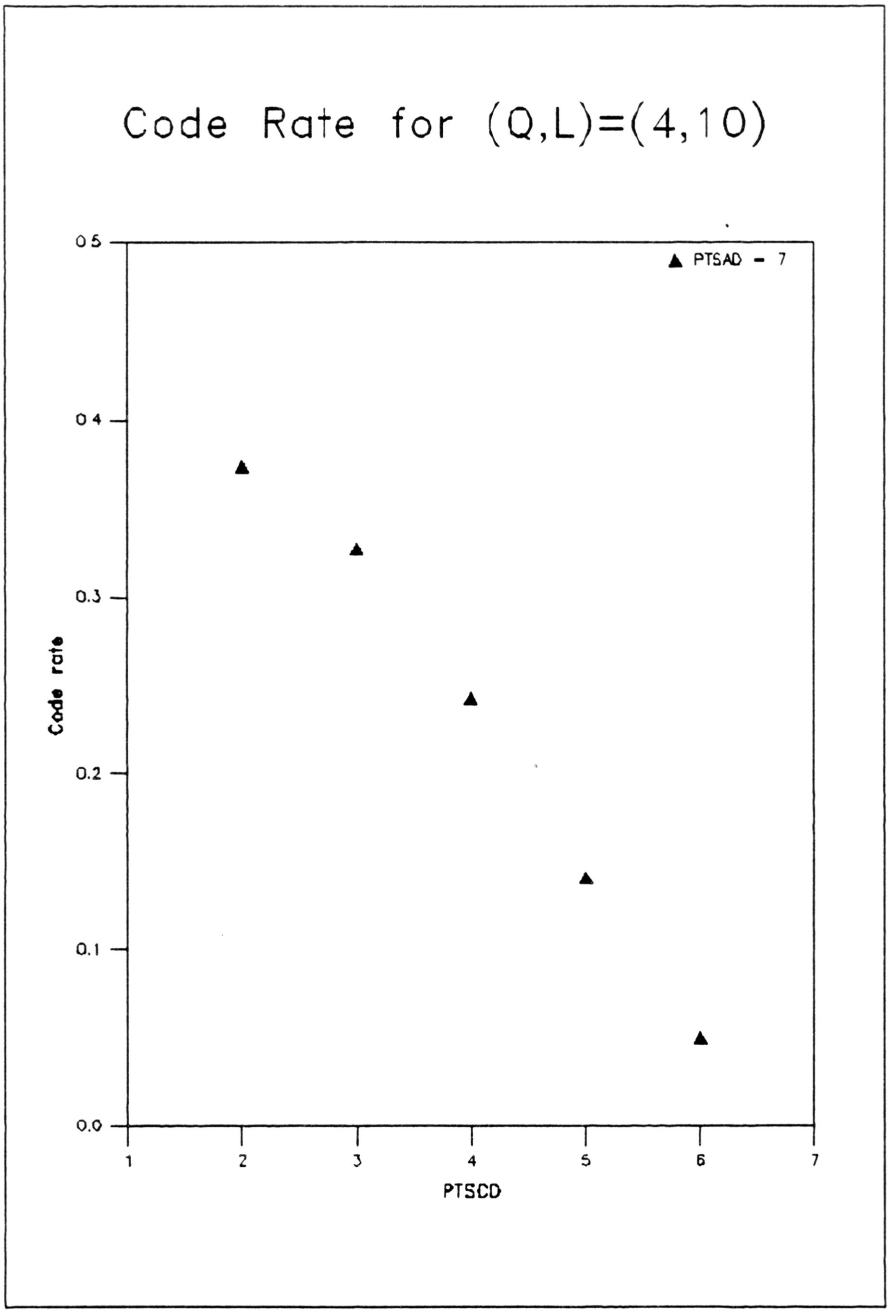


Fig. 9: Typical Code Rate Vs. PTSCD for (Q, L)=(4,10).

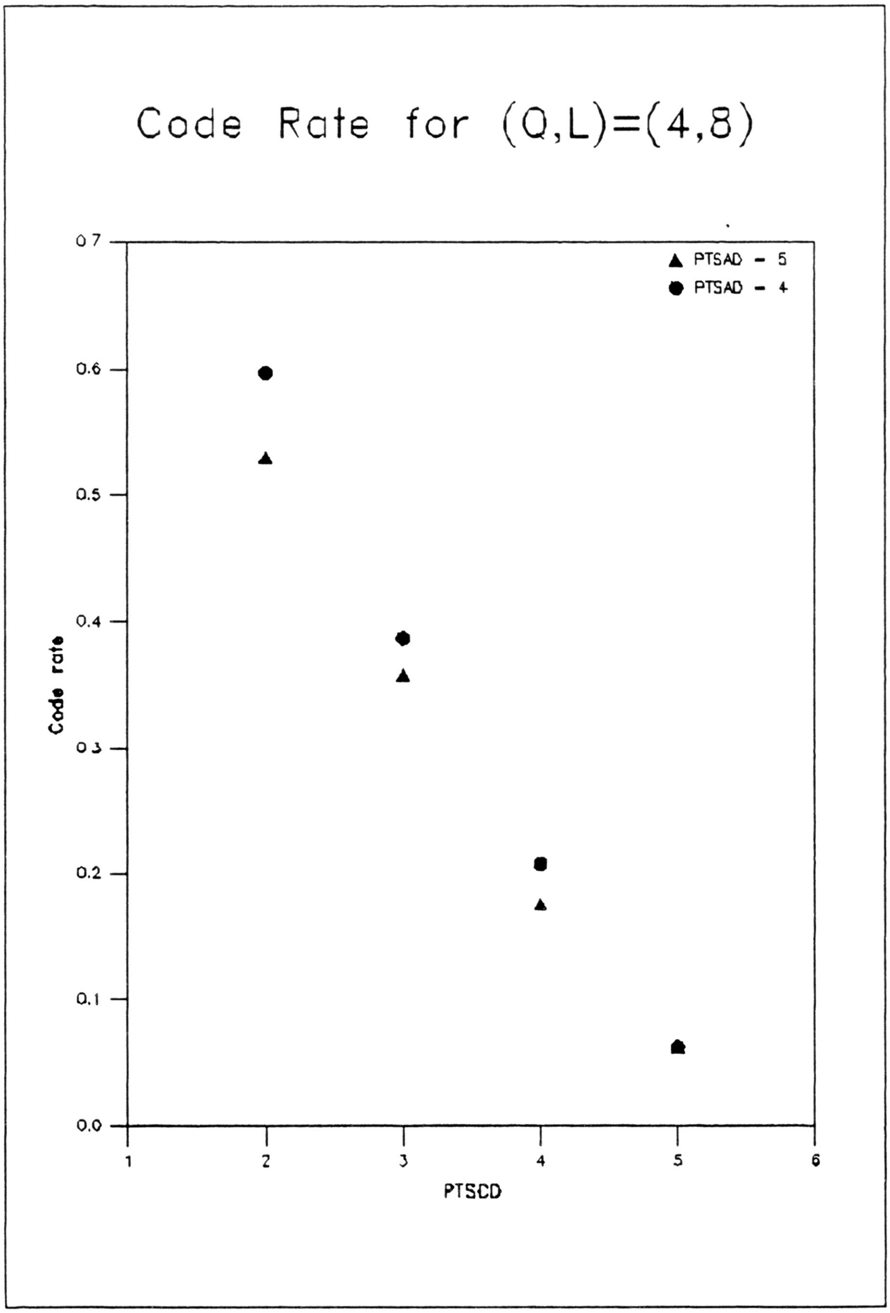


Fig. 10: Typical Code Rate Vs. PTSAD and PTSCD for (Q, L)=(8,4).

CHAPTER V

CONCLUSION

The problem of designing PPM sequences with GAP has been formulated for two different cases: when only slot synchronization is present, and when PPM symbol synchronization is present. The design of PPM sequences with GCP has also been formulated for the case when only slot synchronization is present. Comparative analysis between the computer results and the theoretical bounds for both PTSAD and PTSCD were carried out. Through computer search, the bounds given in the theorems were observed to be tight and achieved with equality for most of the (Q, L) values. Moreover, PPM sequences with both GAP and GCP were investigated.

There are some interesting questions that still remained unanswered. For a given (Q, L), there is no way of determining the number of good PPM sequences with specific peak-to-sidelobe distance. As mentioned earlier, a sequence $\mathbf{S^j}$ must be provided to the algorithm to search for the other sequences with GCP. The computer results show that the same number of sequence with GCP is achieved, regardless of the initial sequence $\mathbf{S^j}$, although a different set of sequences were obtained for different $\mathbf{S^j}$. However, it is still needed to be shown analytically that the total number of good correlation sequences is the same, regardless of the initial sequence $\mathbf{S^a}$.

REFERENCES

[1] C. N. Georghiades, "On PPM Sequences with Good Autocorrelation Properties," To appear in the IEEE Trans. Inform. Theory.

[2] Dilip V. Sarwate and Michael B. Pursley, "Crosscorrelation Properties of Pseudo-random and Related Sequences," *Proc. IEEE* , vol. 68, pp. 593-619, May 1980.

[3] L. R. Welch, "Lower Bounds on the Maximum Cross Correlation of Signals," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 397-399, May 1974.

[4] M. Z. Win, "Optimal Symbol Synchronization," *Research Report*, TEES Report Series, Telecommunication and Control Systems Lab., Texas A&M Univ., 1986.

[5] D. L. Snyder and C. N. Georghiades, "Design of Coding and Modulation for Power Efficient Use of a Bandlimited Optical Channel," *IEEE Trans. Commun.*, vol. COM-31, pp. 560-565, April 1983.

[6] J. R. Lesh, J. Katz, H. H. Tan and D. Zwillinger, "2.5-Bits/Detected Photon Demonstration Program: Description, Analysis, and Phase I Results," in *TDA Progress Report, 42-66*, pp. 115-132, Jet Propulsion Laboratory, Pasadena, Calif., Dec. 1981.

[7] J. L. Massey, "Optimum Frame Synchronization," *IEEE Trans. Commun.*, vol. COM-20, pp. 115-119, April 1972.

[8] C. N. Georghiades and D. L. Snyder, "Locating Data Frame in Direct-Detection Optical Communication Systems," *IEEE Trans. Commun.*, vol. COM-32, pp. 118-123, Feb. 1984.

[9] C. N. Georghiades, "Joint Baud and Frame Synchronization in Direct-Detection Optical Communications," *IEEE Trans. Commun.*, vol. COM-33, pp. 357-360, April 1985.

[10] R. Gold, "Optimal Binary Sequences for Spread Spectrum Multiplexing," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 619-621, 1969.

[11] T. Kasami, "Weight Distribution Formula for Some Class of Cyclic Codes," *Research Report*, Coordinated Science Lab., Univ. Illinois, Urbana., 1966.

[12] R. H. Barker, "Group Synchronization of Binary Digital Systems," in *Communication Theory*. W. Jackson, ED. London, England: Butterworth, 1953, chap. , pp. 273-287.

[13] F. Neuman and L. Hofman, "New Pulse Sequences with Desirable Correlation Properties," in *National Telementary Conference.*, Washington, D.C., pp. 272-282, April 1971.

SUPPLEMENTAL SOURCES CONSULTED

[1] C. N. Georghiades, private communications, 1986-87.

APPENDIX A
ALGORITHMS

```
!                                                                       !
!-----------------------------------------------------------------------!
! Title                                                                 !
        program         GAPslot
!-----------------------------------------------------------------------!
! Abstract                                                              !
!                                                                       !
!       This algorithm is to search the PPM sequences with good        !
! autocorrelation property (GAP)  when only slot synchronization       !
! is present. This program generates the a set of optimal sequences,  !
! maximum achievable peak-to-sidelobe autocorrelation distance         !
! (PTSAD), and the total numbers of optimal sequences.                 !
!                                                                       !
!       For more information, refer to author's undergraduate honors   !
! thesis which can be found in the libary of Texas A&M University.     !
! It is listed as "PPM Sequences with Desirable Correlation            !
! Properties" by Moe Z. Win, University Undergraduate Fellow, 1986-87. !
!                                                                       !
!-----------------------------------------------------------------------!
! History                                                               !
!                                                                       !
!       Author: M. Z. Win    Date: Jan/18/87      Place: TAMU(TCSL)     !
!-----------------------------------------------------------------------!
! Constants                                                             !
!-----------------------------------------------------------------------!
! Variables                                                             !
                implicit        none
!- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -!
! Passed                                                                !
!- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -!
! Internal                                                              !
!                                                                       !
        integer*4 i, k
        integer*4 islots, ibauds, ilength, idelta_length
        integer*4 idsm, imaxdsm, iflag
        integer*4 imaxrepeat, numautoseq
!                                                                       !
        integer*4 iseq( 15 ), ji( 15 ), idelta( 400 )
        integer*4 iautoseq( 60000,15 )
!                                                                       !
!-----------------------------------------------------------------------!
! Routines Called                                                       !
!                                                                       !
!       NUM_GEN                                                         !
!       FIND_DELTA                                                      !
!       MAX_REPEAT                                                      !
!       STORE_AUTO                                                      !
!                                                                       !
!-----------------------------------------------------------------------!
!234567890123456789012345678901234567890123456789012345678901234567890!
!-----------------------------------------------------------------------!
! Code                                                                  !
!-----------------------------------------------------------------------!
!                                                                       !

      open (unit=10, file='autoslot.seq',status='new')
      print *,'slots=?  length = ?'
      read (*,*) islots,ilength
      write (10,*) '     slot synchronization     '
      write (10,*) '     ====================     '
```

```
      write (10,*) '
      write (10,111) islots,ilength
111   format (1h ,'q = ',i3,/,' l = ',i3)

!------------------------------------------------------------------------!
         ibauds = islots * ilength
         idelta_length = ilength * (ilength-1)
         iseq(1) = 0
         imaxdsm = 0
         numautoseq    =0
      do 10 i  = 2,ilength
         iseq(i) = 1
10    continue

100     CALL NUM_GEN ( iseq, ji , ilength, islots, iflag )
         if ( iflag .eq. 1 ) goto 8800

        CALL FIND_DELTA ( ji, idelta, ibauds, ilength,
     .                    idelta_length )

        CALL MAX_REPEAT ( imaxrepeat, idelta, idelta_length )
!                                                                        !
!------------------------------------------------------------------------!
! This part of program is to find the values of d(s,M) maximize over s.!
!------------------------------------------------------------------------!
!                                                                        !

         idsm = ilength - imaxrepeat
         if (idsm .eq. imaxdsm) then
            numautoseq = numautoseq + 1
            CALL STORE_AUTO ( iseq, iautoseq, numautoseq, ilength )
         endif
         if (idsm .gt. imaxdsm) then
            imaxdsm = idsm
            numautoseq = 1
            CALL STORE_AUTO ( iseq, iautoseq, numautoseq, ilength )
         endif
      goto 100
!                                                                        !
!------------------------------------------------------------------------!
! This part of program is to write the "good" sequences from 1 to numgs!
!------------------------------------------------------------------------!
!                                                                        !
8800  write (10,222) imaxdsm, numautoseq
222   format (1h ,'Maximum distance= ',i5,/,
     .              ' Number of good sequence = ',i8)

      do i = 1,min(numautoseq,50)
         write (10,333) (iautoseq(i,k), k = 1,ilength)
333      format(<ilength>(i3,x))
      end do
      stop
      end
!                                                                        !
!------------------------------------------------------------------------!
!                                                                        !
```

```
!                                                                        !
!------------------------------------------------------------------------!
! Title                                                                  !
!                                                                        !
      program GAPsymbol
!------------------------------------------------------------------------!
! Abstract                                                               !
!                                                                        !
!       This algorithm is to search the PPM sequences with good          !
! autocorrelation property (GAP)  when symbol synchronization            !
! is present. This program generates the a set of optimal sequences,     !
! maximum achievable peak-to-sidelobe autocorrelation distance           !
! (PTSAD), and the total numbers of optimal sequences.                   !
!                                                                        !
!       For more information, refer to author's undergraduate honors     !
! thesis which can be found in the libary of Texas A&M University.       !
! It is listed as "PPM Sequences with Desirable Correlation              !
! Properties" by Moe Z. Win, University Undergraduate Fellow, 1986-87.   !
!                                                                        !
! ID(K) = Kth number of the SEQUENCE                                     !
! IGS(K) = Kth number of the "good" sequence                             !
! JIPM   = J                                                             !
!                                                                        !
!------------------------------------------------------------------------!
! History                                                                !
!                                                                        !
!       Author: M. Z. Win     Date: Oct/10/86      Place: TAMU (TCSL)    !
!------------------------------------------------------------------------!
! Constants                                                              !
!------------------------------------------------------------------------!
! Variables                                                              !
!                 implicit         none                                  !
!- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - !
! Passed                                                                 !
!                                                                        !
!                                                                        !
!- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - !
! Internal                                                               !
!                                                                        !
       INTEGER*4 ID(100),IGS(5000,100),JI(100),JIPM(100)
       INTEGER*4 IDELTA(10000),ICOUNT(10000)
!                                                                        !
!------------------------------------------------------------------------!
! Routines Called                                                        !
!                                                                        !
!                                                                        !
!------------------------------------------------------------------------!
!2345678901234567890123456789012345678901234567890123456789012345678901!
!------------------------------------------------------------------------!
! Code                                                                   !
!------------------------------------------------------------------------!
!                                                                        !
      OPEN (UNIT=10, FILE='AUTOSYMBOL.SEQ',STATUS='NEW')
      PRINT *,'SLOTS=?  LENGTH = ?'
      READ (*,*) ISLOTS,ILENGTH
      WRITE (10,*) '      SYMBOL SYNCHRONIZATION     '
      WRITE (10,*) '      ======================     '
      WRITE (10,*) '                                 '
      WRITE (10,111) ISLOTS,ILENGTH
 111  FORMAT (1H ,'Q = ',I3,/,' L = ',I3)
```

```
        IBAUDS = ISLOTS * ILENGTH
        IDELTA_LENGTH = ILENGTH * (ILENGTH-1)
        ID(1) = 0
        IMAXDSM = 0
        NOGS    =0
     DO 10 I  = 2,ILENGTH
        ID(I) = 1
10   CONTINUE
100  ID(1) = ID(1) + 1
        DO 200 I = 1,ILENGTH
              IF ( ID(I) .GT. ISLOTS ) THEN
              ID(I) = 1
              ID(I+1) = ID(I+1) + 1
              IF (ID(ILENGTH) .GT. ISLOTS) GOTO 8000
        ENDIF
200     CONTINUE
!                                                                        !
!------------------------------------------------------------------------!
!                                                                        !
!THIS PORTION GENERATES THE SETS OF ALL  { Ji } WHERE i = 1 to Ilength  !
!                                                                        !
!------------------------------------------------------------------------!
!                                                                        !
           DO 250 I = 1,ILENGTH
              JI(I) = ISLOTS*(I-1) + ID(I)
250        CONTINUE
!                                                                        !
!------------------------------------------------------------------------!
!                                                                        !
! THIS PORTION OF THE PROGRAMME IS TO FIND DELTA, THE SET OF ALL THE    !
! NUMBERS |Ji - Jk|, and QL - |Ji - Jk|                                  !
!                                                                        !
!------------------------------------------------------------------------!
!                                                                        !
              IPOINTER = 0
              IDIFF    = IDELTA_LENGTH / 2
        DO 300 IINCREMENT = 1, ILENGTH-1
           DO 400 ISUBINCREMENT = 1, ILENGTH-1
              IF ((ISUBINCREMENT + IINCREMENT) .EQ. (ILENGTH + 1) )
   *             GOTO 300
              IPOINTER = IPOINTER + 1
              IDIFF    = IDIFF + 1
              IDELTA (IPOINTER) = JI (ISUBINCREMENT +
   *                             IINCREMENT) - JI (ISUBINCREMENT)
              IDELTA (IDIFF)    = IBAUDS - IDELTA (IPOINTER)
400        CONTINUE
300     CONTINUE
!                                                                        !
!------------------------------------------------------------------------!
!                                                                        !
!                         SYMBOL SYNCHRONIZATION                         !
!                                                                        !
! THIS PART OF PROGRAM IS FIND TO THE MAXIMUM NUMBER OF REPEATED m'.    !
!                                                                        !
!------------------------------------------------------------------------!
!                                                                        !
        DO 700 I = 1, IDELTA_LENGTH
           ICOUNT ( IDELTA(I) ) = 0
700     CONTINUE
           IMAXREPEAT = 0
```

```
        DO 800 I = 1, IDELTA_LENGTH
C           RREMAINDER =  MOD ( IDELTA (I) , ISLOTS )
           IF ( (MOD ( IDELTA (I) , ISLOTS) ).EQ. 0 ) THEN
              ICOUNT ( IDELTA(I) ) = ICOUNT ( IDELTA(I)) + 1
              IMAXREPEAT = MAX0 ( ICOUNT (IDELTA(I)) , IMAXREPEAT )
           ENDIF
800     CONTINUE
!                                                                        !
!------------------------------------------------------------------------!
!                                                                        !
! THIS PART OF PROGRAM IS TO FIND THE VALUES OF D(S,m') MAXIMIZE OVER S!
!                                                                        !
!------------------------------------------------------------------------!
!                                                                        !
        IDSM = ILENGTH - IMAXREPEAT
         IF (IDSM .EQ. IMAXDSM) THEN
            NOGS = NOGS + 1
            IF (NOGS .GT. 100) GOTO 7000
            GOTO 2000
         ENDIF
         IF (IDSM .GT. IMAXDSM) THEN
            IMAXDSM = IDSM
            NOGS = 1
            GOTO 2000
         ENDIF
      GOTO 100
!                                                                        !
!------------------------------------------------------------------------!
!                                                                        !
!  THIS PART OF PROGRAM IS TO KEEP THE "GOOD" SEQUENCES FROM 1 TO NOGS !
!                                                                        !
!------------------------------------------------------------------------!
!                                                                        !
2000     DO 3000 K = 1,ILENGTH
               IGS(NOGS,K) = ID(K)
3000     CONTINUE
7000 GOTO 100
8000 WRITE (10,222) IMAXDSM, NOGS
222  FORMAT (1H ,'Maximum distance= ',I5,/,
     c              ' Number of good sequence = ',I8)
      DO 4000 I = 1,MIN(NOGS,50)
            WRITE (10,333) (IGS(I,K), K = 1,ILENGTH)
333         FORMAT(<ILENGTH>(I3,X))
4000 CONTINUE
      STOP
      END
!                                                                        !
!------------------------------------------------------------------------!
!                                                                        !
```

```
!                                                                         !
!-------------------------------------------------------------------------!
! Title                                                                   !
        program          GCPslot
!-------------------------------------------------------------------------!
! Abstract                                                                !
!                                                                         !
!       This algorithm is to search the PPM sequences with good           !
! crosscorrelation property (GCP)  when only slot synchronization         !
! is present. This program generates the a set of optimal sequences,      !
! maximum achievable peak-to-sidelobe crosscorrelation distance           !
! (PTSCD), and the total numbers of optimal sequences.                    !
!                                                                         !
!       For more information, refer to author's undergraduate honors      !
! thesis which can be found in the libary of Texas A&M University.        !
! It is listed as "PPM Sequences with Desirable Correlation               !
! Properties" by Moe Z. Win, University Undergraduate Fellow, 1986-87.    !
!                                                                         !
!-------------------------------------------------------------------------!
! History                                                                 !
!                                                                         !
!       Author: M. Z. Win    Date: Feb/18/87      Place: TAMU(TCSL)       !
!-------------------------------------------------------------------------!
! Constants                                                               !
!-------------------------------------------------------------------------!
! Variables                                                               !
                implicit        none
!- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -!
! Passed                                                                  !
!- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -!
! Internal                                                                !

      integer*4 i, j, k, l
      integer*4 islots, ibauds, ilength
      integer*4 idsm, imaxdis, iflag
      integer*4 imaxrepeat, nogs

      integer*4 iseq(20),  ji( 20 ),  idelta (1000)
      integer*4 igjis( 250,20 ), igs( 250,100 )
!                                                                         !
!-------------------------------------------------------------------------!
! Routines Called                                                         !
!                                                                         !
!       NUM_GEN                                                           !
!       MAX_REPEAT                                                        !
!       STORE_CROSS                                                       !
!                                                                         !
!-------------------------------------------------------------------------!
!234567890123456789012345678901234567890123456789012345678901234567890 1!
!-------------------------------------------------------------------------!
! Code                                                                    !
!-------------------------------------------------------------------------!
!                                                                         !
      open (unit=10, file='crossslot.seq',status='new')
      print *,'slots=?  length = ? max_distance=? '
      read (*,*) islots,ilength,imaxdis
      print *,'Enter the sequence ? '
        do i = 1, ilength
           read (*,*) iseq(i)
        end do
```

```
      write (10,*) '      cross correlation         '
      write (10,*) '      ===================       '
      write (10,*) '                                '
      write (10,111) islots,ilength
111   format (1h ,'   q = ',i3,/,'    l = ',i3)
         ibauds = islots * ilength
         nogs = 0
!------------------------------------------------------------------------!

      do i = 1,ilength
         ji ( i ) = islots * ( i-1 ) + iseq ( i )
      end do

      CALL STORE_CROSS ( iseq, ji, igs, igjis, ilength, nogs )

         iseq(1) = 0

      do 100 i  = 2,ilength
         iseq(i) = 1
100   continue

9900  CALL NUM_GEN ( iseq, ji, ilength, islots, iflag )
      if (iflag .eq. 1) goto 8800

      do 200 i = 1,nogs
             l = 0

             do 300 j = 1,ilength

                    do 400 k = 1,ilength
                           l = l+1
                           idelta(l) = igjis(i,j) - ji(k)
                           if (idelta(l) .lt. 0 )
     *                        idelta(l) = idelta(l) + ibauds
400                 continue

300          continue

             CALL MAX_REPEAT (imaxrepeat,idelta,l)

             idsm = ilength - imaxrepeat
             if (idsm .lt. imaxdis) goto 9900

200   continue

      CALL STORE_CROSS ( iseq, ji, igs, igjis, ilength, nogs)

      go to 9900

8800  write (10,222) imaxdis, nogs
222   format (1h ,'maximum distance= ',i5,/,
     .              ' number of good sequence = ',i8)

      do i = 1,min(nogs,50)
            write (10,333) (igs(i,k), k = 1,ilength)
333         format(<ilength>(i3,x))
      end do

      stop
      end
```

```
!                                                                        !
!------------------------------------------------------------------------!
! Title                                                                  !
        program         GACPslot
!------------------------------------------------------------------------!
! Abstract                                                               !
!                                                                        !
!     This algorithm is to search the sequence with Desireable           !
! Correlations properties (both GAP and GCP ) for given slots            !
! and length ( Q,L ) pair.                                               !
!                                                                        !
! GAP = Good Autocorrelation Property                                    !
! GCP = Good Crosscorrelation Property                                   !
!                                                                        !
!     This program generates the a set of optimal sequences,             !
! maximum achievable peak-to-sidelobe autocorrelation distance           !
! (PTSAD), and the total numbers of optimal sequences.                   !
!                                                                        !
!     For more information, refer to author's undergraduate thesis
! titled "PPM Sequences with Desirable Autocorrelation Properties."
!       For more information, refer to author's undergraduate honors     !
! thesis which can be found in the libary of Texas A&M University.       !
! It is listed as "PPM Sequences with Desirable Correlation              !
! Properties" by Moe Z. Win, University Undergraduate Fellow, 1986-87    !
!                                                                        !
!------------------------------------------------------------------------!
! History                                                                !
!                                                                        !
!       Author: M. Z. Win    Date: Feb/18/87     Place: TAMU(TCSL)       !
!------------------------------------------------------------------------!
! Constants                                                              !
!------------------------------------------------------------------------!
! Variables                                                              !
                implicit        none
!- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - !
! Passed                                                                 !
!                                                                        !
!               none                                                     !
!- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - !
! Internal                                                               !
!                                                                        !
        integer*4 i, j, k, l
        integer*4 imore, ipointer
        integer*4 islots, ibauds, ilength, idelta_length
        integer*4 idsm, imaxdsm, imaxdis, iflag
        integer*4 imaxrepeat, numautoseq, numseq
!                                                                        !
        integer*4 iseq( 15 ), ji( 15 ), idelta( 250 )
        integer*4 iautoji( 60000,15 ), icrossji( 2000,15 )
!                                                                        !
!------------------------------------------------------------------------!
! Routines Called                                                        !
!                                                                        !
!       NUM_GEN                                                          !
!       FIND_DELTA                                                       !
!       MAX_REPEAT                                                       !
!       STORE_AUTO                                                       !
!       STORE_SEQ                                                        !
!                                                                        !
!------------------------------------------------------------------------!
```

```
!234567890123456789012345678901234567890123456789012345678901234567890!
!----------------------------------------------------------------------!
! Code                                                                 !
!----------------------------------------------------------------------!
!                                                                      !
      open (unit=10, file='autocross.seq;',status='new')
      print *,'slots=?  length = ? '
      accept *, islots,ilength

         ibauds = islots * ilength
         idelta_length = ilength * (ilength-1)
!                                                                      !
!----------------------------------------------------------------------!
!----------------------------------------------------------------------!
!                  AUTOCORRELATION CRITERION                           !
!----------------------------------------------------------------------!
!----------------------------------------------------------------------!
!                                                                      !
        iseq(1) = 0
        imaxdsm = 0
        numautoseq    =0

        do 10 i  = 2,ilength
           iseq(i) = 1
 10     continue

        print *, '                          '
        print *, ' Enter Maximum distance for Auto correlation '
        accept *, imaxdsm

1100    CALL NUM_GEN ( iseq, ji , ilength, islots, iflag )
         if ( iflag .eq. 1 ) goto 2200

        CALL FIND_DELTA ( ji, idelta, ibauds, ilength,
     .                    idelta_length )

        CALL MAX_REPEAT ( imaxrepeat, idelta, idelta_length )
!                                                                      !
!----------------------------------------------------------------------!
! This part of program is to find the values of D(S,m) maximize over s.!
!----------------------------------------------------------------------!
!                                                                      !
        idsm = ilength - imaxrepeat

         if ( ( idsm .eq. imaxdsm ) .or. ( idsm .gt. imaxdsm ) ) then

            numautoseq = numautoseq + 1
            CALL STORE_AUTO ( ji, iautoji, numautoseq, ilength )

         endif
!                                                                      !
!----------------------------------------------------------------------!
! This part would find the best autocorrelation sequence (maximum      !
! distances) which have maximum paeak-to-sidelobe autocorrelation      !
! distance (PTSAD).                                                    !
!----------------------------------------------------------------------!
!                                                                      !
*
*        if (idsm .eq. imaxdsm) then
*
```

```
*             numautoseq = numautoseq + 1
*             CALL STORE_AUTO ( ji, iautoji, numautoseq, ilength )
*
*          endif
*
*          if (idsm .gt. imaxdsm) then
*
*              imaxdsm = idsm
*              numautoseq = 1
*              CALL STORE_AUTO ( ji, iautoji, numautoseq, ilength )
*
*          endif
!
!-----------------------------------------------------------------------!
!                                                                       !
      goto 1100
!                                                                       !
!-----------------------------------------------------------------------!
! Input/Output wanning statements.                                      !
!-----------------------------------------------------------------------!
!                                                                       !
2200    print *, '                                              '
        write ( 6,111) numautoseq
 111    format ( '  Number of good Auto correlation sequence = ',i8 )

        if ( numautoseq .gt. 60000 ) then

           write ( 10,* )  '*** WARNNING::Autocorrelation sequences
     .has exceeded the array size. *** '
           print *,  '*** WARNNING::Autocorrelation sequences
     .has exceeded the array size. *** '

        end if
!                                                                       !
!-----------------------------------------------------------------------!
!-----------------------------------------------------------------------!
!                     CROSS CORRELATION CRITERION                       !
!-----------------------------------------------------------------------!
!-----------------------------------------------------------------------!
!                                                                       !
        print *, '                                                  '
5500    print *, '--------------------------------------------------'
        print *, '                                                  '
        print *, ' Enter Maximum distance for Cross correlation '
        accept *, imaxdis
        ipointer = 1
        numseq   = 0

        CALL STORE_SEQ ( icrossji, iautoji, ipointer, ilength, numseq )

3300    ipointer = ipointer + 1
        if ( ipointer .gt. numautoseq ) goto 4400

        do i = 1,numseq
           l = 0

           do j = 1,ilength

              do k = 1,ilength
                 l = l+1
```

```
            idelta(l) = icrossji ( i,j ) - iautoji( ipointer,k )
            if (idelta(l) .lt. 0 ) idelta(l) = idelta(l) + ibauds
         end do

      end do

      CALL MAX_REPEAT (imaxrepeat,idelta,l)

      idsm = ilength - imaxrepeat
      if ( idsm .lt. imaxdis ) goto 3300

   end do

   CALL STORE_SEQ ( icrossji, iautoji, ipointer, ilength, numseq )

   go to 3300

!                                                                    !
!--------------------------------------------------------------------!
! This part of program is to convert Ji to iseq and write            !
! the "good" sequences.                                              !
!--------------------------------------------------------------------!
!                                                                    !

4400   write ( 10,*)  '      Desirable Correlation Properties      '
       write ( 10,* ) '      ================================      '
       write ( 10,* ) '                                     '
       write ( 10,222 ) islots,ilength
222    format ( 1h ,' q = ',i3,/,'  l = ',i3)
       write ( 10,333 ) imaxdsm, imaxdis, numautoseq, numseq
333    format ( 1h ,'Autocorrelation maximum distance =    ',i5,/,
     .         ' Cross correlation maximum distance = ',i5,/,
     .         ' Number of good Autocorrelation sequence = ',i10,/,
     .         ' Number of good Cross correlation sequence = ',i8 )
       write ( 10,* ) '                             '

       do i = 1,min(numseq,50)

          do j = 1,ilength
             iseq ( j ) = icrossji ( i,j ) - islots * ( j - 1 )
          end do

          write (10,444) ( iseq ( k ), k = 1, ilength )
444       format ( < ilength > ( i3,x ) )

       end do

       write ( 10,555 )
555    format ( '1', '                               ' )

       print *, '                               '
       write ( 6,666) numseq
666    format ( ' Number of good Cross correlation sequence = ',i8 )

       print *, ' More Cross correlation sequence for different distance ??? '
       print *, '        YES = 1 '
       print *, '        NO  = 0 '

       accept *, imore
       if ( imore .eq. 1 ) go to 5500
```

```
      if ( ( imore .ne. 1 ) .and. ( imore .eq. 0 ) ) go to 6600

6600  print *, '                                              '
      print *, '______________________________________________'
      stop
      end
!                                                                   !
!___________________________________________________________________!
!                                                                   !
```

APPENDIX B

SUBROUTINES

```
! This is a test program                                              !
!---------------------------------------------------------------------!
!                                                                     !
!       dimension iseq (100), ji (100)
!       type*, ' ISLOTS,ILENGTH '
!       accept*, islots, ilength
!       iseq ( 1 ) = 0
!       do i = 2,ilength
!          iseq (i) = 1
!       end do
!                                                                     !
!10     call numgen ( iseq, ji, ilength, islots, iflag )
!       if ( iflag .eq. 1 ) go to 20
!      write (6,*) (iseq(i), i=1,ilength)
!      write (6,*) (ji(i), i=1,ilength)
!      write (6,*) '-------------------------------------'
!      go to 10
!
!20     stop
!       end
!                                                                     !
!---------------------------------------------------------------------!
! Title                                                               !
        subroutine NUM_GEN ( iseq, ji, ilength, islots, iflag )

!---------------------------------------------------------------------!
! Abstract                                                            !
!---------------------------------------------------------------------!
! Abstract                                                            !
!                                                                     !
!       This subroutine is to generate all possible numbners (Xi)     !
! and maps into Ji for given slots & length (Q,L).                    !
!                                                                     !
!       For more information, refer to author's undergraduate honors  !
! thesis which can be found in the libary of Texas A&M University.    !
! It is listed as "PPM Sequences with Desirable Correlation           !
! Properties" by Moe Z. Win, University Undergraduate Fellow, 1986-87. !
!                                                                     !
!---------------------------------------------------------------------!
! History                                                             !
!                                                                     !
!       Author: M. Z. Win    Date: Jan/01/87     Place: TAMU(TCSL)    !
!---------------------------------------------------------------------!
! Constants                                                           !
!---------------------------------------------------------------------!
! Variables                                                           !
                implicit        none
!- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -!
! Passed                                                              !
!                                                                     !
        integer*4 ilength, islots, iflag
        integer*4 iseq( 15 ), ji( 15 )
!                                                                     !
!- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -!
! Internal                                                            !
!                                                                     !
        integer*4 i
!                                                                     !
!---------------------------------------------------------------------!
! Routines Called                                                     !
```

```
!                                                                          !
!                                                                          !
!--------------------------------------------------------------------------!
!234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901!
!--------------------------------------------------------------------------!
! Code                                                                     !
!--------------------------------------------------------------------------!
!                                                                          !
        iflag = 0
        iseq(1) = iseq(1) + 1

        do i = 1,ilength

               if ( iseq(i) .gt. islots ) then
                  iseq(i) = 1
                  iseq(i+1) = iseq(i+1) + 1

                  if (iseq(ilength) .gt. islots) then
                     iflag = 1
                     goto 8800
                  endif

               endif

        end do
!                                                                          !
!--------------------------------------------------------------------------!
! This portion generates the sets of all  { Ji } where i = 1 to ilength!
!--------------------------------------------------------------------------!
!                                                                          !
        do i = 1,ilength
               ji(i) = islots*(i-1) + iseq(i)
        end do

 8800 return
      end
!                                                                          !
!--------------------------------------------------------------------------!
!                                                                          !
```

```
!                                                                          !
!--------------------------------------------------------------------------!
! Title                                                                    !
        subroutine FIND_DELTA ( ji, idelta,
     .                          ibauds, ilength,idelta_length )
!--------------------------------------------------------------------------!
! Abstract                                                                 !
!                                                                          !
!        This subroutine is to find delta, the set of all the              !
!  numbers |Ji - Jk|, and QL - |Ji - Jk|.                                  !
!                                                                          !
!       For more information, refer to author's undergraduate honors       !
! thesis which can be found in the libary of Texas A&M University.         !
! It is listed as "PPM Sequences with Desirable Correlation                !
! Properties" by Moe Z. Win, University Undergraduate Fellow, 1986-87.     !
!                                                                          !
!--------------------------------------------------------------------------!
! History                                                                  !
!                                                                          !
!       Author: M. Z. Win    Date: Feb/18/87      Place: TAMU(TCSL)        !
!--------------------------------------------------------------------------!
! Constants                                                                !
!--------------------------------------------------------------------------!
! Variables                                                                !
                implicit        none
!- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - !
! Passed                                                                   !
!                                                                          !
        integer*4       ibauds, ilength, idelta_length
        integer*4       ji( 15 ), idelta( 250 )

!                                                                          !
!- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - !
! Internal                                                                 !
!                                                                          !
        integer*4       ipointer, idiff
        integer         iincrement, isubincrement
!                                                                          !
!--------------------------------------------------------------------------!
! Routines Called                                                          !
!                                                                          !
!                                                                          !
!--------------------------------------------------------------------------!
!234567890123456789012345678901234567890123456789012345678901234567890123!
!--------------------------------------------------------------------------!
! Code                                                                     !
!--------------------------------------------------------------------------!
!                                                                          !
        ipointer = 0
        idiff  = idelta_length / 2

        do 300 iincrement = 1, ilength-1

            do isubincrement = 1, ilength-1

                if ( (isubincrement + iincrement) .eq. (ilength + 1) )
     .              goto 300
                ipointer = ipointer + 1
                idiff    = idiff + 1
                idelta (ipointer) = ji (isubincrement +
```

```
                              iincrement) - ji (isubincrement)
              idelta (idiff)    = ibauds - idelta (ipointer)

            end do

300     continue

        return
        end
!                                                                    !
!--------------------------------------------------------------------!
!                                                                    !
```

```
! This program is to test the subroutine                                   !
!--------------------------------------------------------------------------!
!
!       dimension idelta ( 100 )
!10      type*, ' any five numbers '
!       accept*, (idelta( i ), i=1,5)
!       call maxrepeat ( imaxrepeat, idelta, 5 )
!       type*,   imaxrepeat
!       go to 10
!       end
!                                                                          !
!--------------------------------------------------------------------------!
! Title                                                                    !
!                                                                          !
        subroutine MAX_REPEAT ( imaxrepeat, idelta, idelta_length )
!--------------------------------------------------------------------------!
! Abstract                                                                 !
!                                                                          !
! subroutine to find the maximum number of repeated M.                     !
! l = the length of idelta                                                 !
!                                                                          !
!       For more information, refer to author's undergraduate honors       !
! thesis which can be found in the libary of Texas A&M University.         !
! It is listed as "PPM Sequences with Desirable Correlation                !
! Properties" by Moe Z. Win, University Undergraduate Fellow, 1986-87.     !
!                                                                          !
!--------------------------------------------------------------------------!
! History                                                                  !
!       Author: M. Z. Win    Date: Jan/07/87      Place: TAMU(TCSL)        !
!                                                                          !
!--------------------------------------------------------------------------!
! Constants                                                                !
!--------------------------------------------------------------------------!
! Variables                                                                !
                implicit        none
!- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - !
! Passed                                                                   !
!                                                                          !
        integer*4       imaxrepeat, idelta_length
        integer*4       idelta( idelta_length )
!                                                                          !
!- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - !
! Internal                                                                 !
!                                                                          !
        integer*4       i, icount( 0:100 )
!                                                                          !
!--------------------------------------------------------------------------!
! Routines Called                                                          !
!                                                                          !
!                       none                                               !
!--------------------------------------------------------------------------!
!2345678901234567890123456789012345678901234567890123456789012345678901!
!--------------------------------------------------------------------------!
! Code                                                                     !
!--------------------------------------------------------------------------!
!                                                                          !
        do i = 1, idelta_length
             icount ( idelta(i) ) = 0
        end do
```

```
        imaxrepeat = 0

        do i = 1, idelta_length
             icount ( idelta(i) ) = icount ( idelta(i)) + 1
             imaxrepeat = max0 ( icount (idelta(i)) , imaxrepeat )
        end do

        return
        end
!                                                                       !
!-----------------------------------------------------------------------!
!                                                                       !
```

```
!                                                                        !
!------------------------------------------------------------------------!
! Title                                                                  !
        subroutine STORE_AUTO ( ji, iautoji, numautoseq, ilength )
!------------------------------------------------------------------------!
! Abstract                                                               !
!                                                                        !
! This subroutine to store the sequences, which have "good"              !
! autocorrelation propertie (GAP).                                       !
!                                                                        !
!      For more information, refer to author's undergraduate honors      !
! thesis which can be found in the libary of Texas A&M University.       !
! It is listed as "PPM Sequences with Desirable Correlation              !
! Properties" by Moe Z. Win, University Undergraduate Fellow, 1986-87.   !
!                                                                        !
!------------------------------------------------------------------------!
! History                                                                !
!                                                                        !
!       Author: M. Z. Win     Date: Jan/10/87      Place: TAMU(TCSL)     !
!------------------------------------------------------------------------!
! Constants                                                              !
!------------------------------------------------------------------------!
! Variables                                                              !
                implicit        none
!- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - !
! Passed                                                                 !
!                                                                        !

        integer*4       numautoseq, ilength
!                                                                        !
        integer*4       ji( 15 )
        integer*4       iautoji( 60000, 15 )
!                                                                        !
!- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - !
! Internal                                                               !
!                                                                        !
        integer*4       i
!                                                                        !
!------------------------------------------------------------------------!
! Routines Called                                                        !
!------------------------------------------------------------------------!
!234567890123456789012345678901234567890123456789012345678901234567890 1!
!------------------------------------------------------------------------!
! Code                                                                   !
!------------------------------------------------------------------------!
!                                                                        !

        do i = 1, ilength

           iautoji ( numautoseq,i ) = ji( i )

        end do

        return
        end
!                                                                        !
!------------------------------------------------------------------------!
!                                                                        !
```

```
!                                                                          !
!--------------------------------------------------------------------------!
! Title                                                                    !
        subroutine STORE_CROSS( iseq,ji, igs,igjis, ilength,nogs)
!--------------------------------------------------------------------------!
! Abstract                                                                 !
! subroutine to keep the "good" sequences with desirable cross             !
! correlation properties (GCP) from 1 to nogs.                             !
!                                                                          !
!       For more information, refer to author's undergraduate honors       !
! thesis which can be found in the libary of Texas A&M University.         !
! It is listed as "PPM Sequences with Desirable Correlation                !
! Properties" by Moe Z. Win, University Undergraduate Fellow, 1986-87.     !
!                                                                          !
!--------------------------------------------------------------------------!
! History                                                                  !
!                                                                          !
!       Author: M. Z. Win     Date: Jan/15/87      Place: TAMU(TCSL)       !
!--------------------------------------------------------------------------!
! Constants                                                                !
!--------------------------------------------------------------------------!
! Variables                                                                !
                implicit        none
!- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - !
! Passed                                                                   !
!                                                                          !
        integer*4       igs(250,20),igjis(250,20)
        integer*4       iseq(100),ji(100)
        integer*4       nogs, ilength
!- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - !
! Internal                                                                 !
        integer*4       k
!--------------------------------------------------------------------------!
! Routines Called                                                          !
!                                                                          !
!       none                                                               !
!--------------------------------------------------------------------------!
!234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901!
!--------------------------------------------------------------------------!
! Code                                                                     !
!--------------------------------------------------------------------------!
!                                                                          !
      nogs = nogs + 1

      do k = 1,ilength
              igs(nogs,k) = iseq(k)
              igjis(nogs,k) = ji(k)
      end do

      return
      end
!                                                                          !
!--------------------------------------------------------------------------!
!                                                                          !
```

```
!                                                                         !
!-------------------------------------------------------------------------!
! Title                                                                   !
!                                                                         !
      subroutine STORE_SEQ ( icrossji, iautoji,
     .                       ipointer, ilength, numseq )
!-------------------------------------------------------------------------!
! Abstract                                                                !
!                                                                         !
!       This subroutine is to store Ji, which has both GAP and GCP        !
! The ji will be converted back to sequences in the main program.         !
! This is done so to save memory.                                         !
!                                                                         !
!       For more information, refer to author's undergraduate honors      !
! thesis which can be found in the libary of Texas A&M University.        !
! It is listed as "PPM Sequences with Desirable Correlation               !
! Properties" by Moe Z. Win, University Undergraduate Fellow, 1986-87.    !
!                                                                         !
!-------------------------------------------------------------------------!
! History                                                                 !
!                                                                         !
!       Author: M. Z. Win     Date: Feb/25/87       Place: TAMU(TCSL)     !
!-------------------------------------------------------------------------!
! Constants                                                               !
!-------------------------------------------------------------------------!
! Variables                                                               !
                implicit        none
!- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -!
! Passed                                                                  !
!                                                                         !
      integer*4       ipointer, ilength, numseq
!                                                                         !
      integer*4       icrossji (2000, 15 )
      integer*4       iautoji ( 60000, 15 )
!                                                                         !
!- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -!
! Internal                                                                !
!                                                                         !
      integer*4       i
!                                                                         !
!-------------------------------------------------------------------------!
! Routines Called                                                         !
!                                                                         !
!               none                                                      !
!-------------------------------------------------------------------------!
!234567890123456789012345678901234567890123456789012345678901234567890!
!-------------------------------------------------------------------------!
! Code                                                                    !
!-------------------------------------------------------------------------!
!                                                                         !
      numseq = numseq + 1

      do i = 1, ilength
         icrossji ( numseq, i ) = iautoji ( ipointer, i )

      end do

      return
      end
!                                                                         !
```