THE UNIVERSITY OF MANITOBA

SOME COMBINATORIAL PROPERTIES OF COMPLEMENTARY SEQUENCES

by

TERRANCE H. ANDRES

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE (M.Sc.)

DEPARTMENT OF COMPUTER SCIENCE

WINNIPEG, MANITOBA

May 1977

"SOME COMBINATORIAL PROPERTIES OF COMPLEMENTARY SEQUENCES"

by

TERRANCE H. ANDRES

A dissertation submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

TABLE OF CONTENTS

TABLE OF ILLUSTRATIONS

CHAPTER 1: INTRODUCTION

1.1 Definitions

Two sequences of 1's and -1's are said to be complementary if the number of like pairs having a separation of k elements in the two sequences equals the number of unlike pairs so separated, for k= 1,2, ... Complementary sequences have combinatorial properties of considerable interest, partially due to their applications to physical problems in more than one field. As a consequence, the theory of these sequences has passed through two distinct generations , as different researchers have bent their studies toward different ends. In the process, complementary sequences have been generalized in several ways. In this work, these developments are reviewed, and generalized in the form of self-complementary sequences of orthogonal vectors (SCoSOV's).

Complementary sequences in their original form can be defined more explicitly with the aid of correlation functions. Given a pair of finite sequences, $A = (a_i)$ and $B = (b_i)$, for i=1(1)L, the correlation function of A with B for a spacing k is defined to be

$$R_{AB}(k) = a_1 b_{1+k} + a_2 b_{2+k} + \ldots + a_{L-k} b_L .$$

This expression can be more concisely written as

$$R_{AB}(k) = \sum_{i=-\infty}^{\infty} a_i b_{i+k} ,$$

if the assertion is made that $a_i = b_i = 0$ for i<1 and for i>L. Note that

$$R_{AB}(k) = R_{BA}(-k)$$

using this form of the summation. The autocorrelation of A with itself for a spacing k is, by analogy,

$$R_{AA}(k) = R_{AA}(-k) = \sum_{i=-\infty}^{\infty} a_i \, a_{i+k} .$$

With the aid of these functions, a pair of sequences, $A = (a_i)$ and $B = (b_i)$ will be said to be complementary if

$$R_{AA}(k) + R_{BB}(k) = 0,$$

for k not equal to 0. That is,

$$\sum_{i=-\infty}^{\infty} (a_i \, a_{i+k} + b_i \, b_{i+k}) = 0,$$

for k not equal to 0, if the domains of the sequences are again suitably extended. Henceforth, any subscripted variable bearing a subscript outside its domain of definition, whether a vector or a scalar, will be assumed to have the appropriate zero value.

1.2 History and Applications

The known properties of complementary sequences were all given by Marcel Golay in his landmark paper of 1961 [5]. This paper was , in fact, his third paper on the subject, the first two [3, 4] being in the field of optics. Golay used complementary sequences in the late 1940's in the design of multi-slit spectrometers. Whereas a single-slit spectrometer has a long narrow aperture, through which a light source shines , the multi-slit device has a series of such slits, parallel and carefully spaced. Light shining through such apertures is diffracted, by an amount determined by its frequency and by the width of the slits, to produce a beam-splitting effect, similar in some respects to the refractive dispersion of light in a prism. However the image of each slit on a screen or recording device is complicated by the appearance of light and dark bands, caused by reinforcement and cancellation of light waves. For very narrow slits, these fringes will be wide and dim. For wider slits, the bands will be narrow, but brighter. In the many-slit spectrometer, several slits are used to enhance efficiency, but narrow slits can be used to permit greater dispersion.

Complementary series were first used in the design of the slit arrangement of such a spectrometer, as shown in figure 1, which was taken from Golay's 1961 paper. The slits were equally spaced, with some open or uncovered, and some closed, or sealed, in a pattern specified by two complementary sequences, A and B. The detector was placed behind a similarly-slitted screen, so that the autocorrelation function was automatically calculated by summing the radiation incident on the screen apertures. This design permitted efficient use of the light from the source, and the selection of just one frequency range for examination through effective cancellation of all others.

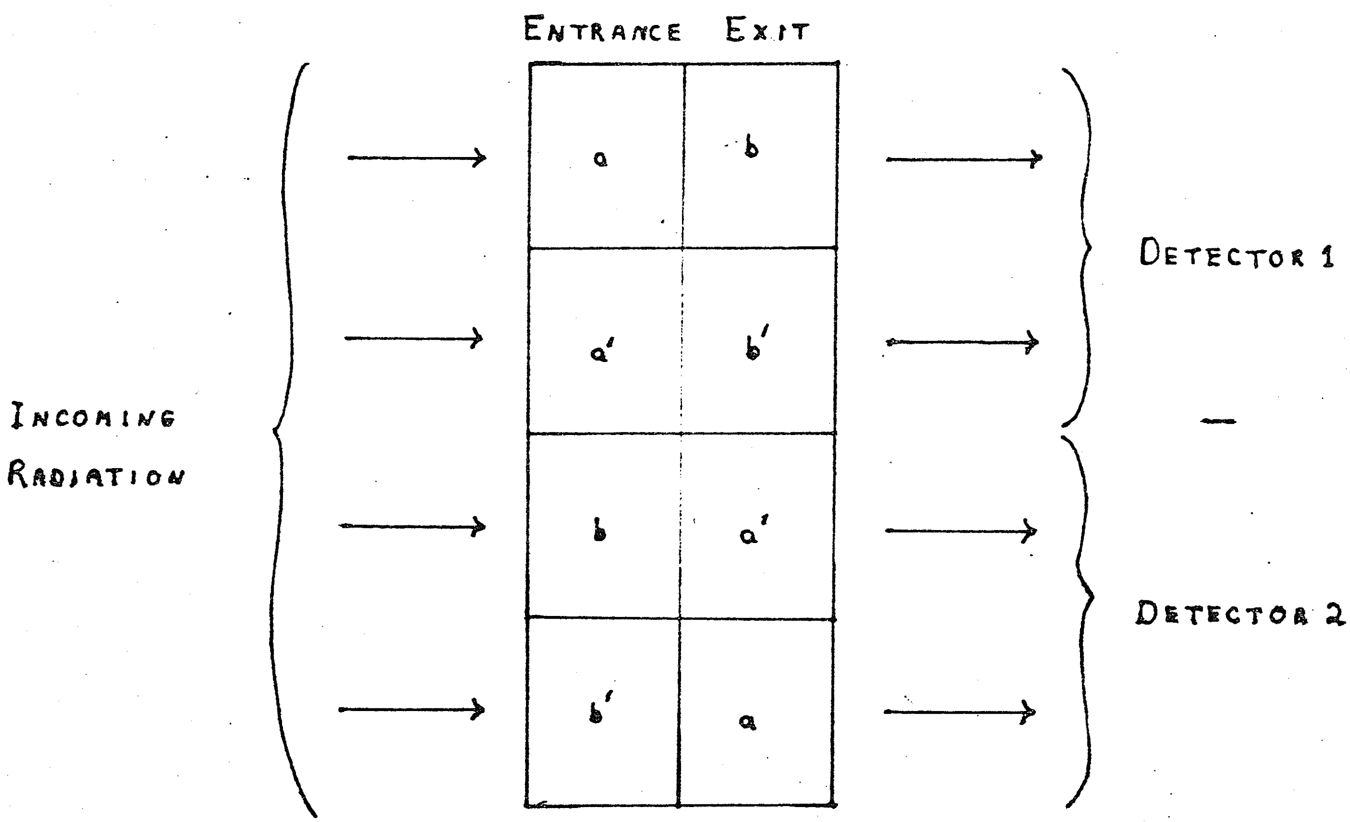


FIG. 1 - MULTI-SLIT SPECTROMETER

The interests of Golay in the combinatorial aspects of complementary series apparently extended beyond their usage in spectroscopy, for his 1961 paper refers only briefly to optics. Primarily, he discusses combinatorial properties of complementary series , as well as methods of synthesis of longer sequences from certain short basis series of lengths 2 and 10 . Having given a necessary condition (which will be stated later as Theorem 2.7) for the existence of complementary series, he also used a counter-example of length 18 to show that the condition was not sufficient. In a later note [6], Golay was also the first to publish an example of complementary series of length 26. It is interesting to note that although extensive computer searches have been made by myself and others [8], the last new complementary pair discovered was found by Golay in a "by hand" search.

In other fields besides optics, physical phenomena can often be represented by correlation functions, and the utility of complementary series rests on this fact. Correlations appear in communications theory as an intrinsic aspect of the separation of signals from noise [24]. It is not surprising then to find that the theory of radar, born in the 1940's, should have found applications for complementary sequences in the 1950's and 1960's. The use of special codings for transmitted radar signals was explored to some extent in the frequently referenced paper by Siebert [14]. Then, at the same time that Golay was publishing his paper , George Welti [23] produced a paper showing how certain codings could be successfully used in pulsed radar for range detection. Richard Turyn later established the isomorphism between the sequences of Golay and the codes of Welti [20].

Communications theory, to which radar studies belong, covers many smaller disciplines, but stripped to the essentials, the problem of extracting signals from noise has the same characteristics, whether the medium be radar or surface acoustic waves on crystals, or something even more exotic.

The basic objectives are commonly twofold. First, a signal which has been transmitted must be recognized, usually after power dissipation during transmission, and in the presence of noise. Secondly, the time of arrival of the signal must be determined. Meeting the first objective is hampered by a limitation on the maximum power output of the transmitter. (Otherwise, the signal strength could be stepped up until it blotted out all interference.) The second aim is limited by the frequency bandwidth of the transmitted signal. (This is the principle that makes laser ranging much more accurate than radar ranging, because light waves have a higher frequency than radar waves.) Conflict arises in attempts to satisfy both criteria, because the signal which packs the most energy will have a continuous power output, while a spike output will be easiest to accurately measure in the time domain. A compromise can be achieved by using a receiver which continuously correlates the incoming signal with the form of the transmitted signal. In the discrete-coding case with which we are concerned, the signal consists of a long pulse containing many shorter pulses, separated by possible phase reversals. In effect, the result is a time-varying sequence of 1's and -1's. If the receiver is matched to the transmitter, the general form of the received correlation function will appear as in figure 2, which shows a pseudo-random code. The key features are the central peak and the smaller side-peaks(side-lobes). If clever coding can keep the ratio of side-lobe height to central peak height as low as possible, then the energy transmitted will be concentrated into the main peak. Of course, to transmit information, the whole pattern must be repeated for each bit of the message. Even this feature turns out not to be a disadvantage in the world of surface acoustic wave (S.A.W.) devices.

Many codes have been constructed for the purpose of making the side-lobe to centre peak ratio as low as possible. These include the Barker codes [1], illustrated in figure 3, which feature side-lobes of constant minimal amplitude, and of constant sign, the Welti codes [23], which as we have said, are isomorphic to complementary series, but

a) 1 - 1 1 - - 1 - 1 1 1 - 1 - - -

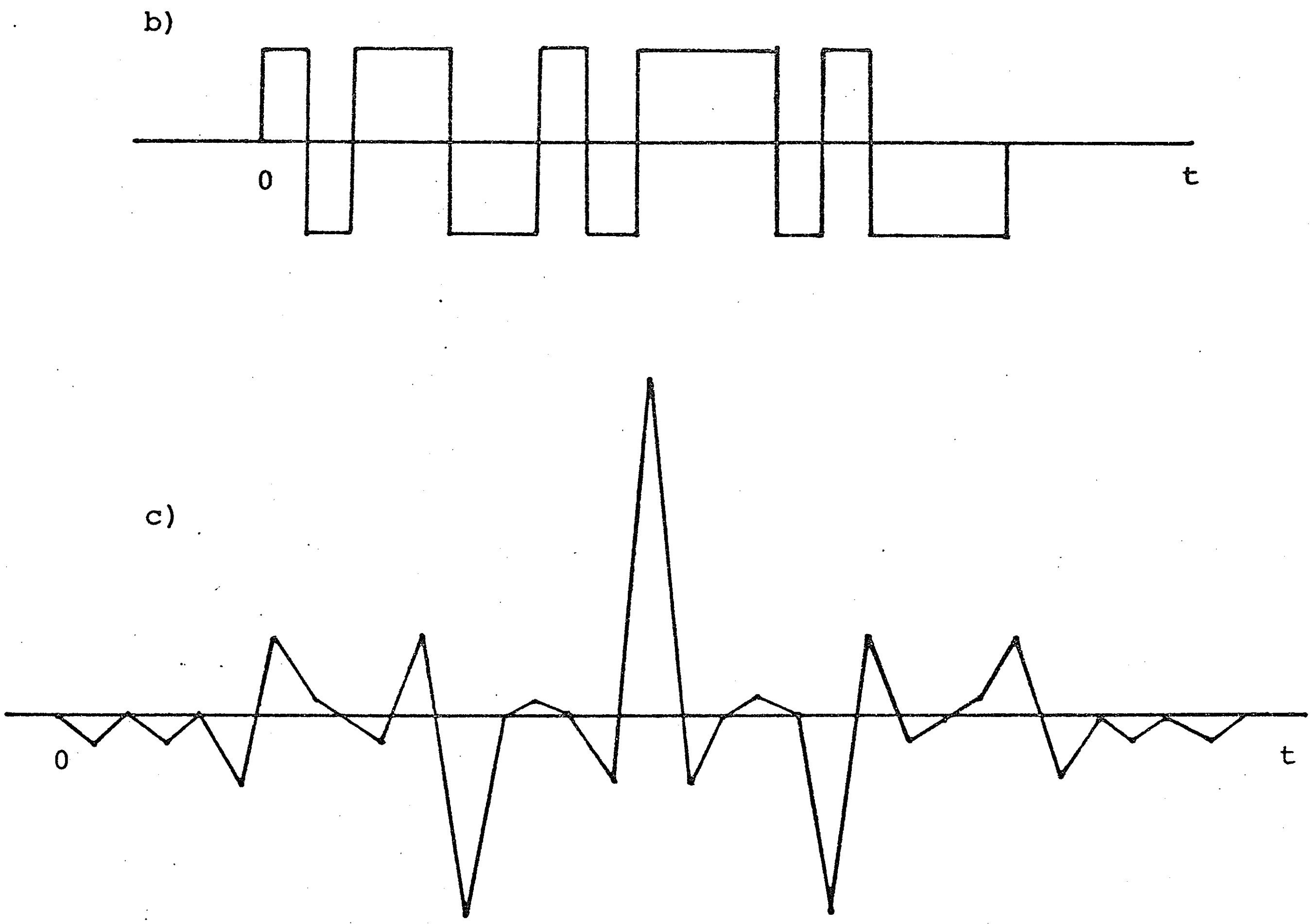


FIG. 2 - A PSEUDO-RANDOM CODE: a) A CODE OF LENGTH 16; b) THE CODE AS A FUNCTION OF TIME; c) THE AUTO-CORRELATION FUNCTION.

a) 1 1 1 - - 1 -

b)

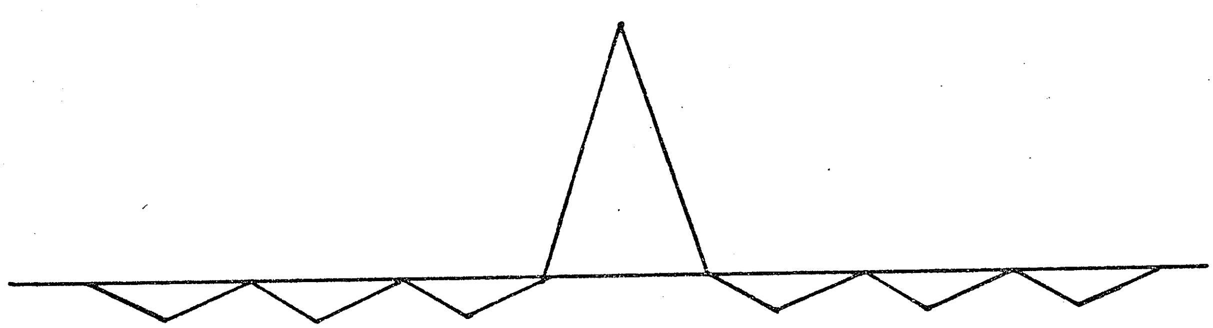


FIG. 3 - A BARKER CODE: a) THE CODE, OF LENGTH 7

b) THE AUTO-CORRELATION FUNCTION

which are not binary; the Frank codes [2], which are also not binary, etc. Any such codes must necessarily have non-zero side-lobes, caused by the arrival of the first sub-pulse, which is correlated in the receiver with the last sub-pulse of the transmitted pattern.

Using complementary series, however, an ideal situation can be (theoretically) achieved where no side-lobes, but only the central peak, exist. In exchange the use of complementary series extracts the penalty of using two channels. Each signal received must be independently correlated with the corresponding transmitter signal, and the two channel outputs can be summed to produce the net output. This procedure is illustrated in figure 4. The central peak will correspond to a zero-shift correlation, and as such , it will have a magnitude of 2L.

In practice, only if the channels are matched extremely well will the theoretical predictions be reached. In range-finding radar applications, for instance, signals could be transmitted at two different frequencies (requiring expensive duplication of equipment). The transmitters , receivers, and amplifiers would then have to be similar in response, and stable over long periods of time. Even then, spurious results might be obtained if the target responded differently at the two frequencies. Alternatively, the two signals could alternate on the same

a) A = 1 1 1 - - 1 - -
   B = 1 1 1 - 1 - 1 1

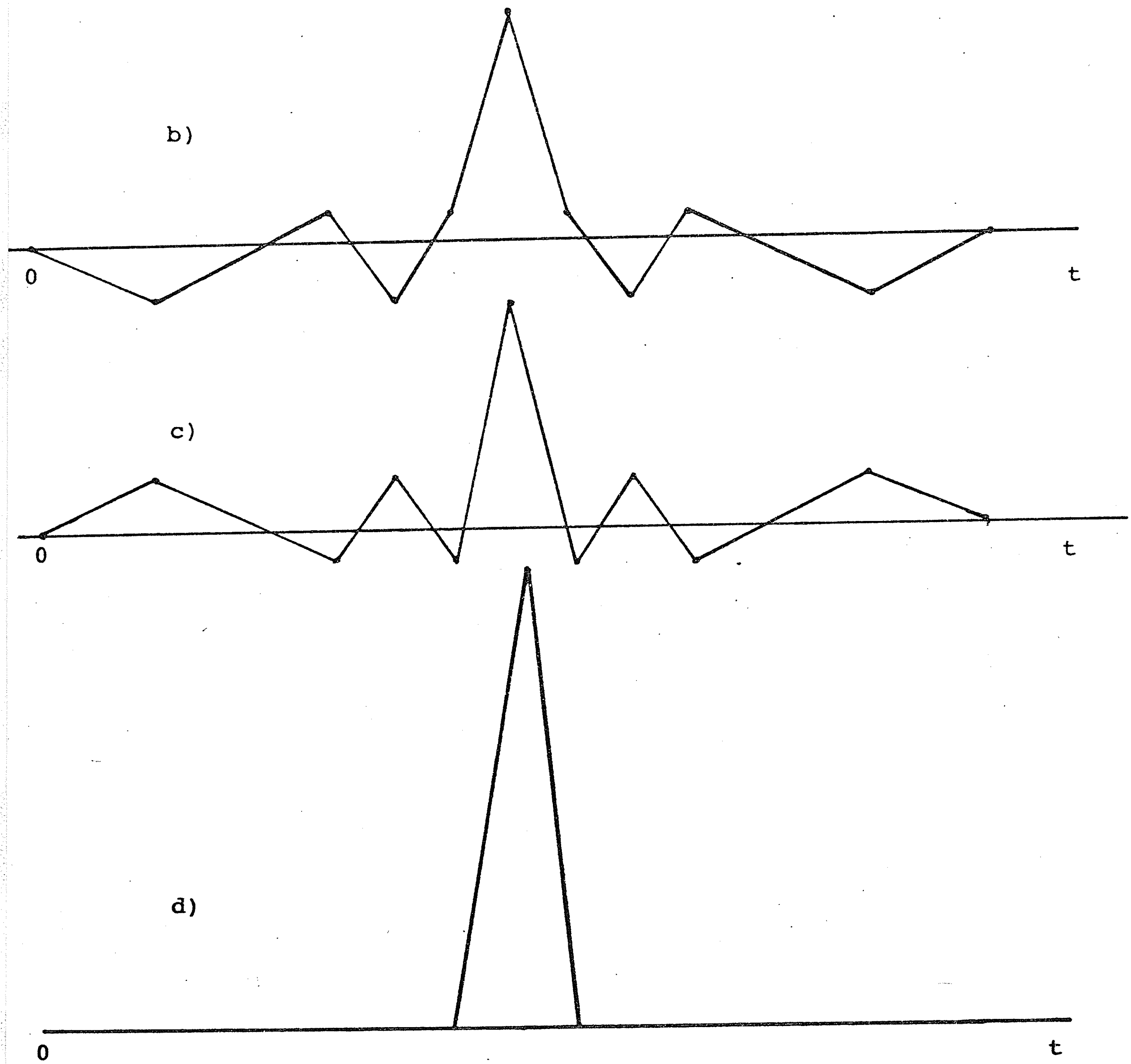


FIG. 4 - A PAIR OF COMPLEMENTARY SEQUENCES: a) SEQUENCES A AND B; b) AUTO-CORRELATION FUNCTION OF A; c) AUTO-CORRELATION FUNCTION OF B; AND d) SUM OF b) AND c)

equipment, if one of the signals could be delayed to permit the necessary summing when the second signal arrived. That technique would prove unsatisfactory should the echo change appreciably in the time required for one pulse. The difficulties involved in creating two channels in radar usage might very well outweigh the advantages of using complementary series, except in very high signal density situations.

Surface-acoustic wave (S.A.W.) devices [9] have proved much more amenable to the application of complementary series. These devices utilize the piezo-electric properties of certain crystals to convert electronic signals to and from Rayleigh waves, which propagate across the crystal at a speed much less than the speed of light. This slow velocity makes S.A.W. delay lines very compact. Furthermore, various signal-modifying and filtering functions can be performed simply through the design of the transducers on the crystals. It is however in delay lines that Golay sequences have been widely used [15, 16, 18, 21].

The first favourable circumstance involves the ease with which two identical channels can be created in a S.A.W. device. The transmitters consist of two transducers lying side-by-side on the surface of the crystal. The transducers appear as in figure 5. Each one consists of a sequence of

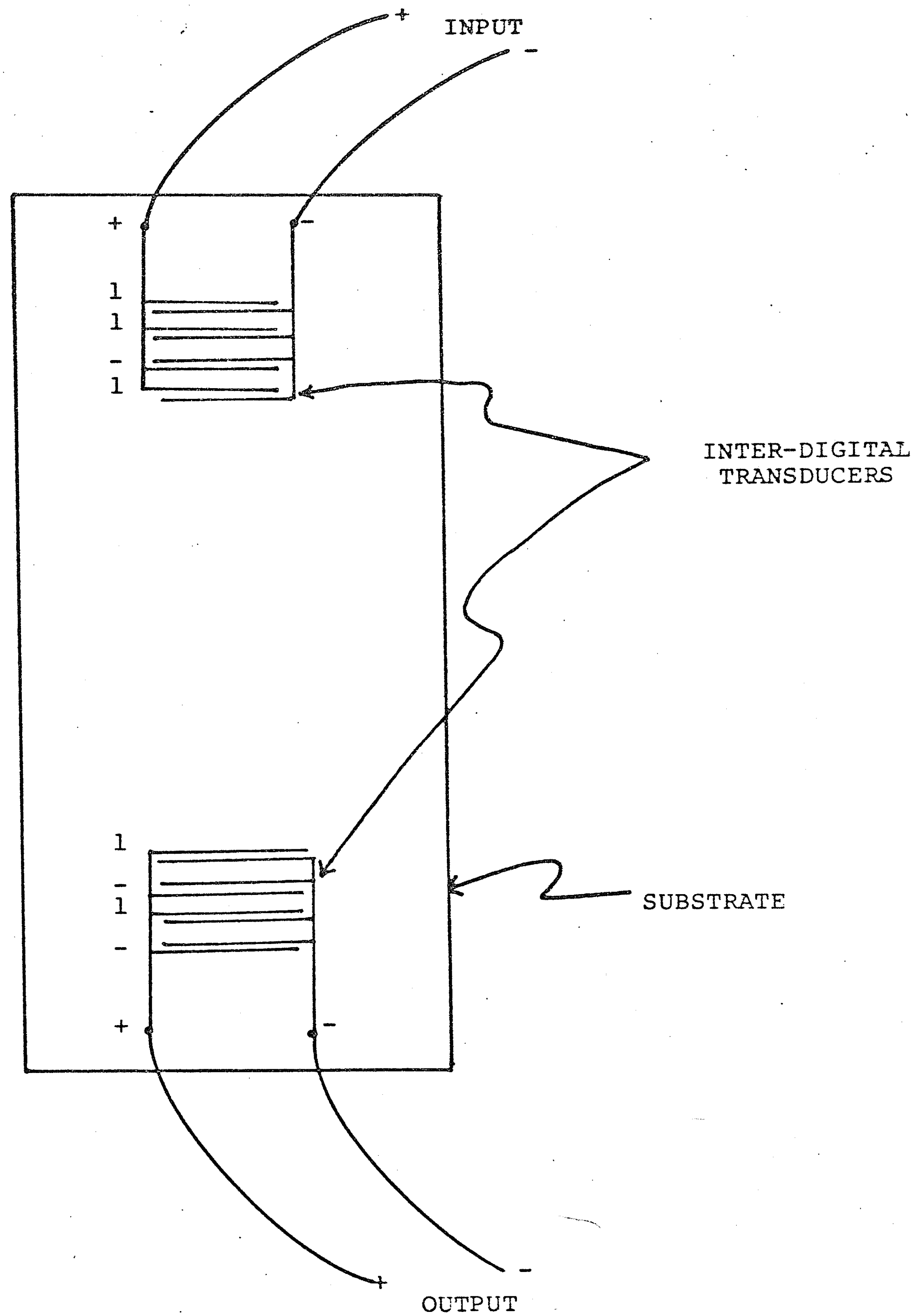


FIG. A S.A.W. DEVICE

metallic digit pairs deposited on the crystal material. In each case, one of the digits is grounded, and the other is given the input signal. A piezo-electric crystal will always change shape in the presence of an electric field, and create an electric field when subjected to physical stress. Therefore, when a signal is applied to one of the inter-digital pairs, it will create a surface wave which will propagate at right angles to the digits. The pairs themselves are separated by gaps sufficiently large to prevent interference. An identical set of inter-digital transducers acts as a receiver some distance along in the path of the acoustic waves. The correlation functions are automatically and continuously formed, and summed by connecting the output leads of the two sets of receiving electrodes. The revolution in micro-circuit technology since the mid 1960's has ensured that to all intents and purposes the two sets of transducers can be made identical.

Of course, as in radar applications, the duplication of anything involves additional expense. In this case, the area on the crystal used for acoustic pathways costs the most. The nature of this cost has been considerably reduced since C. C. Tseng published a paper [18] showing that two acoustic pathways could simultaneously carry two independent signals in a non-interfering manner, provided that these signals were orthogonal. This development started the search for orthogonal sets of complementary series, known as orthogonal mates. Figure 6 shows how two acoustic pathways could be efficiently shared by two independent, but non-interacting, signal lines.

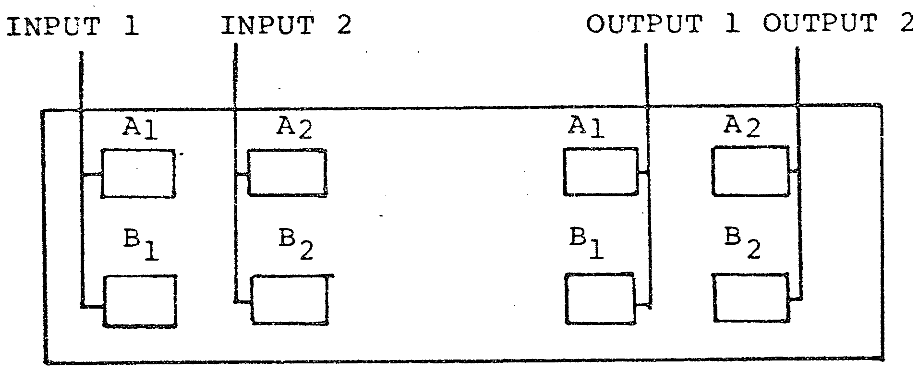
Mathematically, this non-interfering quality can be expressed in the following way. Two pairs of complementary series (A,B) and (C,D) are orthogonal if

$$R_{AC}(k) + R_{BD}(k) = 0,$$

for $k = \ldots -2, -1, 0, 1, 2, \ldots$.

In his doctoral dissertation, Bernard Schweitzer has shown how n sequences can form a complementary system, which he calls a code, and how n such codes can form a mutually orthogonal set, which he calls a complementary code set. He proceeds to show how such complementary code sets (CCS) can be synthesized from certain primitive code sets, using various transformations devised by him. For binary sequences, the primitive elements in his chains of synthesis are Hadamard matrices [22], the properties of which are well known.

Using other orthogonal matrices, Schweitzer was able to construct real-valued CCS. These were found to have certain properties not shared by the binary CCS. In fact, the nature of this generalization is so broad that the real codes share very little in common with the binary codes. SCOSoV's were designed to fill this large gap, by allowing a)

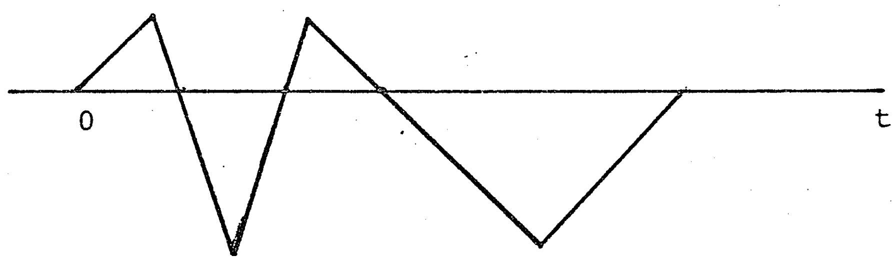


b)

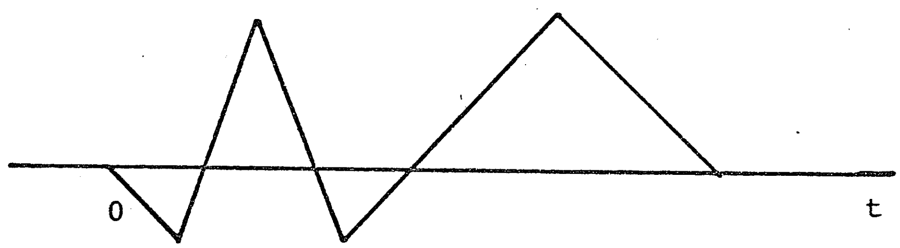
A_1: 1 1 - 1 $\qquad$ A_2: 1 - - -

B_1: 1 1 1 - $\qquad$ B_2: 1 - 1 1

c)



d)



e)

FIGURE 6 - ORTHOGONAL COMPLEMENTARY SEQUENCES

a) MULTIPLYING S.A.W. DEVICE; b) ORTHOGONAL COMPLEMENTARY PAIRS, (A_1, B_1), (A_2, B_2); c) $R_{A_1A_2}(t)$; d) $R_{B_1B_2}(t)$; e) $R_{A_1A_2}(t)+R_{B_1B_2}($

real values to occur, while retaining much of the discrete nature of the original complementary series. In general, these form a much more restricted class than Schweitzer's real CCS.

1.3 Outline

In Chapter 2 are given the known properties of complementary series as they were known to Golay. These properties include restrictions on the lengths of complementary series, as well as transformations which leave the complementary property invariant, and restrictions on the number and arrangements of 1's in complementary series. In general, these properties do not greatly restrict the number of sequences which must be examined in an exhaustive search for complementary series (c.f. [10]).

Since different techniqes must be used to render tractable the problem of searching for complementary sequences, Chapter 3 is devoted to the search for such methods. It will be shown that from a partial description of a possible solution, complementary series, if they exist, can be found through a well-defined sequence of operations involving matrix reduction in the binary field (BF). This field contains the numbers (0,1), along with the operations of addition and multiplication modulo 2. The application of this procedure is given as a set of computer programs in

Appendix 1. These programs have shown that no complementary series exist of lengths 34, 36, or 50.

In Chapter 4, various properties will be derived for self-complementary sequences of orthogonal vectors (SCoSOV's). Some of these sequences are isomorphic to complementary series, but due to their generality, even sequences of functions can be accomodated within the definition. In effect they abstract the complementary property from the sequences in which it appears. They have the further advantage that SCoSOV's are subject to the search techniques of Chapter 3. Several methods of synthesis of SCoSOV's will be given, as well as a table of some known SCoSOV's.

Chapter 5 will contain a brief summary of the work of Bernard Schweitzer. His doctoral dissertation [13] concerned itself with orthogonal codes and complementary code sets. In this chapter the connection between complementary sequences and Hadamard matrices will be developed. Certain theorems of Schweitzer will also be given, without proofs.

The remaining chapter, Chapter 6, will be a review and summary of the previous chapters. It will also contain a brief mention of research work which remains to be done in this field.

CHAPTER 2: COMPLEMENTARY SEQUENCES

2.1 Synopsis

The study of complementary series was originally restricted to properties of two sequences, A and B, of ones (1's) and minus ones (-'s) with the complementary property

(2.1) $R_{AA}(k) + R_{BB}(k) = 2L\delta_{0K}$,

where L is the length of the sequences, and δ_{0K} is the Kronecker delta, which has the value 1 if its indices are alike, and 0 otherwise. Most of the following properties of such sequences were known to Golay when he wrote his 1961 paper [5]. The first few are quite trivial and will be given with minimal explanation.

Theorem 2.6, however, is very important, although it will not be proved until Chapter 3. It will be used to introduce the concept of a quad of elements in a pair of complementary series, a concept which will be indispensible in Chapter 3.

Theorem 2.7 is the sole existing criterion by which one may reject a sequence of length L as unable to support a pair of complementary series.

The remainder of the material in the chapter gives some information about the distribution of 1's and -'s in the sequences themselves. This discussion comes in two parts, the first dealing with the number of 1's which appear in certain subsequences of the two primary sequences, and the second which deals with the frequency of occurrence of certain short patterns of 1's and -'s.

2.2 Basic Properties

The first few theorems here do not require much explanation.

Theorem 2.1: Two complementary series A and B have the same length.

Proof: Suppose without loss of generality that the lengths of the sequences were L and M respectively, with L < M. Then

$$R_{AA}\ (M-1) = 0,$$

but $$R_{BB}\ (M-1) = \pm 1.$$

Therefore,

$$R_{AA}\ (M-1) + R_{BB}\ (M-1) \neq 0,$$

which contradicts our original assumption.

Theorem 2.2: Interchanging A and B does not affect the complementary property.

Proof: $R_{AA}(k) + R_{BB}(k) = R_{BB}(k) + R_{AA}(k) = 2L\,\delta_{0k}$.

Theorem 2.3: Reversing A and/or B does not affect the complementary property.

Proof: One need only notice that if A′ is A reversed,

$$R_{A'A'}(k) = R_{AA}(k).$$

Theorem 2.4: Negating A and/or B does not affect the complementary property.

Proof: Negating all elements does not change any of the products in the correlations. Therefore, (2.1) is still satisfied after the negation.

Theorem 2.5: Negating every second element of A and B does not affect the complementary property.

Proof: Here there are two cases. For even k, negating every second element in A leaves $R_{AA}(k)$ unchanged, since the new signs will invariably cancel, and the same situation holds for $R_{BB}(k)$ as well. For odd k, $R_{AA}(k)$ becomes negated, since one factor of each term has had its sign changed. Again the same result pertains as well to $R_{BB}(k)$. Therefore, (2.1) is negated, but remains 0.

Notice that due to the previous theorems, it does not matter which set of elements is negated and which is left unchanged in either sequence.

Together, Theorems 2.2 - 2.5 imply that one pair of complementary series can be used to generate many others, which may or may not all be distinct. Two sequence pairs, (A,B) and (C,D), will be termed isomorphic if (A,B) can be obtained from (C,D) by any sequence of the operations mentioned in Theorems 2.2 - 2.5.

The next theorem greatly reduces the number of sequence pairs which are potentially complementary.

Theorem 2.6: If $A = (a_i)$ and $B = (b_i)$ are complementary series, then

(2.2) $a_i \; a_{L+1-i} \; b_i \; b_{L+1-i} = -1,$

for i=1(1)L.

Proof: This theorem can be proved by induction on k in (2.1). However, the proof will be given in Chapter 3, where a different method will be used.

Corollary: The length L of any pair of complementary series is even.

Proof: In the theorem take i = (L+1)/2, when L is odd. By the theorem, the product of two real squares would equal -1, an obvious contradiction. Therefore, L must be even.

This corollary can be used to prove the following very important result.

Theorem 2.7: If (a_i), i=1(1)L, and (b_i), i=1(1)L, are complementary series of length L, then

(2.3) $L = 2 \times (u^2 + v^2)$,

for some integers u and v.

Proof: Consider the identity

(2.4) $(R_{AA}(k) + R_{BB}(k)) = 2L.$

Inside the summation, every pair of elements from A occur exactly once together as a term, and similarly for the elements of B. The net positive result comes from the squared terms, all the rest cancelling out. Due to this cancellation, one can double all the non-square terms, and write

(2.5) $(\sum_{i=1}^{L} a_i)^2 + (\sum_{i=1}^{L} b_i)^2 = 2L.$

However, since L must be even by the corollary of the previous theorem, both $\sum_{i=1}^{L} a_i$ and $\sum_{i=1}^{L} b_i$ must be even, since each term is either 1 or -1. Let $u = 0.5 \times (\sum_{i=1}^{L} a_i)$ and $v = 0.5 \times (\sum_{i=1}^{L} b_i)$ to get

$$(2u)^2 + (2v)^2 = 2L,$$

and hence

$$2 \times (u^2 + v^2) = L.$$

This result proves the theorem, and also assigns a meaning to u and v, namely one-half the sum of their respective series. Once L is known, the absolute values of these sums can be computed. From these numbers, the number of 1's in a pair of complementary series will be known to belong to a set of very limited size. Theorem 2.8 follows at once.

Theorem 2.8: If $A = (a_i)$ and $B = (b_i)$ are complementary sequences of length L, then without loss of generality it can be assumed that A has $m = u + (L/2)$ 1's and B has $n = v + (L/2)$ 1's, where u and v are positive integers which satisfy equation (2.3). It can further be assumed that n is not greater than m.

Proof: First note that if the sum of the series formed from either sequence is negative, it can be made positive through negation of that particular sequence (Theorem 2.4). Then if m and n are the number of 1's in the first and second sequences respectively, the following equations hold:

$$m - (L-m) = 2u,$$

and

$$n - (L-n) = 2v.$$

The unique solutions to these equations are

$$m = u + (L/2),$$

and

$$n = v + (L/2).$$

Finally, the order of the two sequences may be interchanged if necessary to guarantee that n does not exceed m.

Further information can be obtained about the distribution of the 1's from Theorem 2.5. In particular, the number of 1's in the subsequences with even subscripts can be determined. For convenience, the following notation is introduced.

(2.6) $A(i,j) = (a_i \ , \ a_{i+j} \ , \ a_{i+2j} \ , \ \ldots \ ),$

and $B(i,j) = (b_i \ , \ b_{i+j} \ , \ b_{i+2j} \ , \ \ldots \ ).$

Since the number of 1's in such subsequences is of interest, the following notation will also be introduced:

$m(i,j)$ = the number of 1's in $A(i,j)$,

and $n(i,j)$ = the number of 1's in $B(i,j)$.

A theorem analogous to Theorem 2.8 can now be proved.

Theorem 2.9: If A and B are complementary sequences of length L, then one of the two following systems of equations will hold, where without loss of generality it can be assumed that $m(1,2) \leq m(2,2)$ and $n(1,2) \leq n(2,2)$.

$$(2.7) \quad \begin{array}{lcl} m(1,2) = L/4 & & m(1,2) = (L/4) + (u-v)/2 \\ m(2,2) = (L/4) + u & \text{or} & m(2,2) = (L/4) + (u+v)/2 \\ n(1,2) = L/4 & & n(1,2) = (L/4) + (u-v)/2 \\ n(2,2) = (L/4) + v & & n(2,2) = (L/4) + (u+v)/2 \end{array}$$

Here u and v are non-negative integers which satisfy (2.3) .

Proof: By Theorem 2.3 isomorphic forms of A and B have $m(1,2) < m(2,2)$ and $n(1,2) < n(2,2)$, so those conditions may be considered met. But then by Theorem 2.5 the terms of $A(1,2)$ and of $B(1,2)$ can be negated to obtain sequences C and D which are complementary and isomorphic to A and B. The number of 1's appearing in C and D will still be $u+(L/2)$ and $v+(L/2)$, as in Theorem 2.8, but now in no particular order. This statement must be true, because the number of 1's in C, for instance, is given by the expression

$$\begin{aligned} n(C) &= m(2,2) + (L/2) - m(1,2) \\ &= (L/2) + (m(2,2)-m(1,2)) \\ &> L/2. \end{aligned}$$

similarly,

$$\begin{aligned} n(D) &= (L/2) + (n(2,2)-n(1,2)) \\ &> L/2. \end{aligned}$$

Therefore, either

$$\begin{aligned} m(2,2)-m(1,2) &= u \qquad \text{or} \qquad m(2,2)-m(1,2) = v \\ n(2,2)-n(1,2) &= v \qquad \text{or} \qquad n(2,2)-n(1,2) = u. \end{aligned}$$

Recall that

$$\begin{aligned} m(1,1) &= m(1,2) + m(2,2) = u + (L/2) \\ n(1,1) &= n(1,2) + n(2,2) = v + (L/2). \end{aligned}$$

There are therefore two parallel sets of four equations in four unknowns, which have the two parallel solutions given in the statement of the theorem.

Obviously, the first solution can hold only if L is congruent to 0 modulo 4. In most cases where one is seeking unknown complementary series, the length is congruent to 2 modulo 4 (exception: L=36).

As an example of the use of the last two theorems, let L=34. the sequences A and B can then without loss of generality be manipulated so that m(1,1) = 21 and n(1,1) = 18. Furthermore, it can be asserted that m(1,2) = 10 and m(2,0) = 11, and n(1,0) = 7 and n(2,2) = 11. Unfortunately, although 17 is the sum of two squares, and although all these numbers can be computed, no complementary sequences of length 34 exist. Accordingly, these theorems must be considered as interesting observations, rather than immediately useful results.

Can the reasoning behind Theorems 2.8 and 2.9 be carried further? To a certain extent, the answer is yes. For instance, consider the six sequences A(1,3), A(2,3), A(3,3), and B(1,3), B(2,3), B(3,3), which may be named s_1, s_2, s_3, t_1, t_2, t_3. A little insight reveals that

$$\sum_{i=1}^{3} (R_{s_i s_i}(k) + R_{t_i t_i}(k)) = R_{AA}(3k) + R_{BB}(3k) = 2L\,\delta_{0k} .$$

A system has been created of six complementary sequences, not necessarily all of the same length. Conditions on the number of 1's in these sequence can be drawn in the same manner as with the original A and B. That is,

$$\sum_{i=1}^{3} \sum_{k=-\infty}^{\infty} (R_{s_i s_i}(k)+R_{t_i t_i}(k)) = 2L,$$

which can also be written,

$$\sum_{i=1}^{3} \left(\left(\sum_{K=-\infty}^{\infty} s_{iK} \right)^2 + \left(\sum_{K=-\infty}^{\infty} t_{iK} \right)^2 \right) = 2L,$$

which reveals 2L to be the sum of 6 integer squares. This condition is clearly weaker than Theorem 2.7. In fact, a theorem of Fermat [12] states that any positive integer can be expressed as the sum of only four integer squares.

Very extensive arguments based on results of this nature have been used [10] to prove that no complementary pairs of length 18 exist. Unfortunately, this proof is very detailed. Perhaps an algorithm could be devised for computer implementation which would check that the results of this nature which can be obtained for all possible subsequences A(i,j) and B(i,j) would be possible for a given L. The degree of complexity which such an examination would entail can be seen in the following example.

Take again, as a guinea pig, the case where L = 34. The new criterion requires that 68 be the sum of six squares. Furthermore , since the subsequences are of lengths 11 (in four cases) and 12, four of the six must be odd. Moreover, taken in groups, the restrictions

$$\sum_{i=1}^{3} \left(\sum_{K=-\infty}^{\infty} s_{iK} \right) = 21,$$

and

$$\sum_{i=1}^{3} \left(\sum_{K=-\infty}^{\infty} t_{iK} \right) = 18$$

must hold. The following decompositions of 68 satisfy all these properties.

$$
\begin{aligned}
68 &= 49 + 16 + 1 + 1 + 1 \\
&= 49 + 9 + 4 + 4 + 1 + 1 \\
&= 49 + 9 + 9 + 1 \\
&= 64 + 1 + 1 + 1 + 1 \\
&= 36 + 25 + 4 + 1 + 1 + 1 \\
&= 36 + 9 + 9 + 9 + 4 + 1 \\
&= 25 + 25 + 9 + 9 \\
&= 25 + 25 + 9 + 4 + 4 + 1 \\
&= 25 + 16 + 16 + 9 + 1 + 1 \\
&= 25 + 16 + 9 + 9 + 9
\end{aligned}
$$

The multiplicity of possibilities renders this approach unappetizing , for such decompositions must be performed for many different subsequences of the originals before any useful results could be obtained. It can be shown that similar decompositions for SCoSOV's involve less complexity than these, even for the case isomorphic to this one. Consequently, a variation of this method may yet be used to discuss the existence of complementary series too large to be found by search.

2.3 Frequencies of Certain Bit Patterns

The last section used the combinatorial property, for many spacings k, to generate some information about the total number of 1's appearing in a pair of complementary sequences. Some quite different results can be obtained by studying the effects of specific spacings on short portions of these sequences. As of this writing, it has not been found possible to combine the two sets of information in any meaningful manner.

The most obvious results come from equation (2.1) using a spacing of k = 1. If the complementary sequences A and B are of length L, then each autocorrelation with k = 1 will have L - 1 terms, for a total of 2L - 2 terms in the entire equation. Half of these must be positive and half negative. Therefore there are exactly L-1 pairs of adjacent elements in the two sequences which differ (as well as L-1 pairs which are the same). If a block of elements in a sequence of 1's and -'s is defined to be a set of contiguous elements of the sequence, all of the same kind, then the following theorem can be deduced immediately.

Theorem 2.10: If A and B are two complementary sequences of length L, then exactly L+1 distinct blocks must exist in the two sequences.

Proof: Every block in either of the sequences must have two ends. Every pair of unlike adjacent elements in either sequence marks the ends of two blocks. Furthermore, the four ends of A and B each mark the end of one block. The number of block endings is thus 4+2L-2 = 2L+2. But then there must have been L+1 blocks.

The necessity of introducing the four ends of A and B in the proof of this theorem provided a slight complication. If the discussion is extended to larger spacings k, then this complication becomes no longer slight, but instead renders many of the numbers involved indefinite. In order to make the results as concrete as possible, therefore, the concept of cyclic complementary sequences will be introduced here.

Definition: A and B are said to be cyclic complementary sequences of length L if

$$(2.8)\qquad C_{AA}(k)+C_{BB}(k) = \Big(\sum_{i=1}^{L}{}' a_i\, a_{i+k}\Big)+\Big(\sum_{i=1}^{L}{}' b_i\, b_{i+k}\Big) = 2L\,\delta_{0k},$$

for $k=0(1)L-1$, where the apostrophes indicate that all the subscripts are taken modulo L (i.e. The subscripts are mapped onto the set (1 2 3 ... L-1 L)). The function which has been denoted here as $C_{AA}(k)$ is a cyclic autocorrelation.

Cyclic complementary series can be used to obtain useful properties about complementary series because of the following statement.

Theorem 2.11: Every pair of complementary sequences is also a pair of cyclic complementary sequences.

Proof: First verify the easy identity

$$(2.9) \qquad C_{AB}(k) = \sum_{i=1}^{L}{}' \, a_i \, b_{i+k}$$
$$= \sum_{i=1}^{L-k} a_i \, b_{i+k} + \sum_{i=1}^{k} a_{L+i-k} \, b_i$$
$$= R_{AB}(k) + R_{BA}(L-k),$$

for k=0(1)L-1. At once it is clear that

$$C_{AA}(k)+C_{BB}(k)=(R_{AA}(k)+R_{AA}(L-k))+(R\ (k)+R\ (L-k))$$
$$= L \quad + L$$
$$= 2L$$

for the complementary series A and B.

It will now be possible to prove several interesting facts about cyclic complementary sequences, and immediately extend them to complementary series. The notion cyclic block will occur frequently in this connection. A cyclic block is either a block, or the set of elements formed by taking the block which ends a sequence and the block which begins the same sequence, should they happen to contain the same kind of elements. A cyclic block then is just a block in a sequence which has been curved into a closed ring.

First, a restatement of Theorem 2.11 in cyclic form will be given.

Theorem 2.12: If A and B are cyclic complementary sequences of length L, then they contain exactly L cyclic blocks.

Proof: By Theorem 2.6, the endpoints of the two sequences A and B include three elements of one type, and one of the other. Therefore, one of the sequences A and B must be disconnected in the middle of a cyclic block. The other has two ends of two cyclic blocks at its ends. Therefore, by analogy with the proof of Theorem 2.11, a count of 2L ends of cyclic blocks has been made, implying the existence of L cyclic blocks.

Next, consider the implications of equation (2.1) with a spacing k=2. It will prove convenient to consider all the possible forms that three consecutive elements of the cyclic complementary sequences can take. For simplicity, these forms will be independent of the sign of the first of the three elements. Thus, without loss of generality, all the length 3 groupings can be represented in the symbolic notation 111, 11-, 1-1, 1--. It will be convenient to represent the number of the class 1-1, for instance, by the notation n(1-1). The four variables n(111), n(11-), n(1-1), n(1--) have a unique value which can be found through the solution of four linear equations. Of the many such equations which can be stated, the following four, which are independent, will be used.

$$
\begin{aligned}
& n(111) + n(11-) - n(1-1) - n(1--) = 0 \\
(2.10)\quad & n(111) - n(11-) + n(1-1) - n(1--) = 0 \\
& n(111) - n(11-) - n(1-1) + n(1--) = 0 \\
& n(111) + n(11-) + n(1-1) + n(1--) = 2L
\end{aligned}
$$

These equations have been derived from the following considerations: to satisfy equation (2.1) for a spacing $k=1$, the number of groupings which have the first two elements the same must equal the number of groupings which have the first two elements unlike, and similarly for the last two elements. Furthermore, to satisfy equation (2.1) for a spacing $k=2$, the number of groupings which have first and last elements alike must equal the number in which those two elements differ. Finally, since each element starts exactly one grouping of length 3, there must be exactly 2L of such groupings.

The unique solution to this system of equations is easily found to be $n(111) = n(11-) = n(1-1) = n(1--) = L/2$. From the third of these numbers the following theorem is drawn.

Theorem 2.13: The number of cyclic blocks of length 1 in a pair of cyclic complementary sequences A and B, of length L, is exactly $L/2$.

Proof: Each cyclic block of length 1 will occur exactly once as the middle element of a grouping of length 3. The number of such groupings, $n(1-1)$, is the number of such blocks, namely $L/2$.

Of course, the same reasoning will generate a system of equations in the 8 variables n(1111), n(111-), n(11-1), n(11--), n(1-11), n(1-1-), n(1--1), n(1---). In matrix form, one such system is:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} n(1111) \\ n(111-) \\ n(11-1) \\ n(11--) \\ n(1-11) \\ n(1-1-) \\ n(1--1) \\ n(1---) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2L \end{bmatrix}$$

This system is underdetermined. The lack of equations does not represent the need to find more conditions, but a genuine lack of restraints in the system itself. This fact is exemplified by the following non-isomorphic pairs of complementary series for L=8.

1111-11-	111--1--
11---1-1	111-1-11

In the former, n(1111) = 1, and in the latter n(1111) = 2. Any attempt to find another independent equation which linked our eight variables would then be doomed to failure, since it would imply a unique solution for n(1111) in this case.

Of course, this counterexample does not deny the possibility of extra restraints in other particular cases, nor does it prohibit general restraints in the form of inequalities or non-linear equations, which could have multiple solutions. In fact, it will be shown that certain particular solutions to the general solution can be disqualified at once.

The general solution, again in matrix form, is:

$$(2.11)\qquad \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} n(1111) \\ n(111-) \\ n(11-1) \\ n(11--) \\ n(1-11) \\ n(1-1-) \\ n(1--1) \\ n(1---) \end{bmatrix} = \begin{bmatrix} 0.5L \\ 0 \\ 0 \\ 0.5L \\ 0 \\ 0.5L \\ 0.5L \end{bmatrix}$$

The form of these equations does not uniquely specify the number of cyclic blocks of length 2 or of length 3 or more. It does, however, imply the following rather interesting relations.

Theorem 2.14: In the cyclic complementary sequences A and B, the number of pairs of adjacent cyclic blocks of length one, the number of cyclic blocks of length 2, and the number of pairs of adjacent cyclic blocks of length at least 2 are all the same.

Proof: By (2.11) the values of our eight variables all fall into one of two classes, namely either n(1---) or 0.5L-n(1---). n(1-1-) counts the number of pairs of adjacent length-1 groups, n(11--) counts the number of pairs of adjacent cyclic block of length greater than 1, and n(1--1) counts the number of cyclic blocks of length 2 . These values all fall into the same class, and hence are equal.

Some of the possible solutions permitted by (2.11) can be eliminated at once. For instance, if n(1---) = 0, then n(1111) = 0.5L. This situation cannot occur, for there are not enough 1111 groupings to fill a whole sequence. At the start of any cyclic block long enough to support a 1111 grouping, a 1--- grouping must occur. This contradiction still occurs because our sets of equations would hold, not just for two cyclic complementary sequences, but for any number of cyclic complementary sequences which contained a total of 2L elements, and had a minimum sequence length of 4. The nature of this formulation of the problem unfortunately renders impractical the application of the condition that only two sequences are used.

Attempts to analyze the frequencies of occurrence of groups of length 5 have proved even less successful, since the number of variables doubles to 16, but only twelve equations are forthcoming.

If the conditions derived in the last two sections of this chapter could be applied together, then perhaps more stringent requirements could be determined for the existence of complementary sequences.

CHAPTER 3: GENERATION OF COMPLEMENTARY SEQUENCES

3.1 Introduction

Although a reasonable amount is known about the theoretical properties of complementary sequences, very few actual sequences have been found. Despite the fact that an infinite number of sequences can be generated by the methods outlined in Section 3.4, all these complementary sequences will have lengths which can be given by the formula $L = 2^{\alpha}\, 10^{\beta}\, 26^{\gamma}$. For practical applications, no more is needed. There remains, however, the tantalizing question "Do other complementary sequences exist?"

At present, the scoreboard appears as follows. The set of integers up to 100 which satisfy Theorem 2.7 is:

(2, 4, 8, 10, 16, 18, 20, 26, 32, 34, 36, 40, 50, 52, 58, 64, 68, 72, 74, 80, 82, 90, 98, 100)

Of these potential sequence lengths, members of the following set are known to support complementary sequences:

(2, 4, 8, 10, 16, 20, 26, 32, 40, 52, 64, 80, 100).

Of these, only 2, 10, and 26 had to be found from first principles; the rest followed automatically through theorems of Golay and Turyn, given here in Section 3.4.

The following lengths have been laboriously disqualified:

(18, 34, 36, 50).

The first was eliminated by hand by Golay, and then by Kruskal [10]. The rest were tested exhaustively by myself, using the computer programs of Appendix 1.

The remaining lengths still rest in the class of undecided cases. These are:

(58, 68, 72, 74, 82, 90, 98).

The sheer number of possibilities offers the greatest resistance to tackling these greater lengths. A naive exhaustive search can be ruled out at once; even for the known case L=34, the number of potential sequences is $2^{68} = 2.95 \times 10^{20}$. Using Theorem 2.6, we can of course improve this number to $2^{51} = 2.25 \times 10^{15}$. If a fast computer could test 1,000,000 sequences per second, the search would be finished in about 71 years of solid cpu time. By using the Theorems 2.2 - 2.5 to develop a specific form for the end bits of the sequences a further saving of a factor of 8 can be made. Using all these, and Theorem 2.7, Stephen Jauregui required 75 hours of computer time to exhaustively search all sequence pairs of length 26 [8].

An enormous improvement in capability can be made by using the more sophisticated techniques of Section 3.3, which reduce the number of sequences to be tested to $2^{L/2-1}$ $= 2^{16}$ = 65536. Of course, the testing itself takes longer for these sequences, because they are only partial descriptions of the potential complementary sequences they represent, but still the problem has been shrunk to a manageable size. The equivalent to Jauregui's search now takes about a minute of cpu time on a PDP 11/45 computer.

In order to attack the first presently unknown case, L = 58, still more powerful methods will probably have to be used. For this L, the number of possibilities jumps to 2^{28} = 2.68×10^{8} even for my algorithm. At the present rate of 30 sequences disqualified per second, about one-third of a year of CPU time would be required for an exhaustive search. This figure is contingent of course upon the hardware, the operating system, and the implementation of the algorithm, so that improvements in one or more of these areas might make an exhaustive search practicable.

3.2 Methods of Reduction of Systems of Multinomial Equations in Binary Variables

As a preparation for Section 3.3, certain preliminary results must be introduced and established. These deal with finding solutions to systems of multinomial equations in variables which can take on only the values 1 and -1. These methods will then be applied to equation (2.1) with k taking on values 0, 1, 2, ... L-1.

The term "binary variable" will be used to describe a variable which can assume only the values 1 and -1. An unspecified pair of complementary sequences of length L is composed of binary variables.

A multinomial equation in binary variables has the form (3.1) $\sum_{j=1}^{m} c_j \; a_j = 0$,
where c_j is an integral coefficient, and a_j is a binary variable (or the product of binary variables), for j = 1(1)n. The constant coefficients can be positive or negative. Any binary variable which appears will have an exponent of 1, since all even powers vanish as 1's, and all odd powers are equivalent to the first power. For example, the following three equations form a multinomial system, in which all the subscripted variables appearing are binary.

$$\begin{aligned} & a_1 \; a_L + b_1 \; b_L = 0 \\ (3.2) \quad & a_1 \; a_{L-1} + a_2 \; a_L + b_1 \; b_{L-1} + b_2 \; b_L = 0 \\ & a_1 \; a_{L-2} + a_2 \; a_{L-1} + a_3 \; a_L + b_1 \; b_{L-2} + b_2 \; b_{L-1} + b_3 \; b_L = 0 \end{aligned}$$

It would seem that this system has three equations in twelve unknowns. In fact, there are fifteen equations in twelve unknowns, for the following twelve equations have been implicitly assumed.

$$a_1 \; a_1 = 1, \; a_2 \; a_2 = 1, \; \ldots \; a_L \; a_L = 1$$

$$(3.3) \qquad b_1 \; b_1 \; = 1, \; b_2 \; b_2 \; = 1, \; \ldots \; b_L \; b_L \; = 1$$

Every system of multinomial equations can be considered over-determined with the introduction of one implicit equation for every variable used. For that reason solutions may exist only rarely.

The following theorem can be used to eliminate up to one variable for each equation in the system.

Theorem 3.1: Every multinomial equation

$$(3.4) \qquad \sum_{j=1}^{m} c_j \; a_j \; = 0$$

has an associated equation

$$(3.5) \qquad \prod_{i=1}^{m} \left[a_i \frac{c_i}{|c_i|}\right]^{c_i} = (-1)^{\sum_{j=1}^{m} |c_j|/2}$$

which must hold whenever the original equation does.

Proof: A term in (3.4) with the coefficient 5 can be considered to be the sum of five 1's or five -1's . With this reasoning, (3.4) can be considered to be a sum with $\sum_{j=1}^{m} c_j$ 1's and -1's, which will be called the binary terms of the sum. Half of these binary terms must be positive, and half negative to make the sum 0. Therefore, the product of all of these binary terms, which is the left-hand side of (3.5), must be given by the right-hand side of (3.5).

The following corollary, a restatement of Theorem 2.6, can now be proved.

Corollary: (Theorem 2.6) If $A = (a_i)$ and $B = (b_i)$ are complementary sequences of length L, then

$$a_i \; a_{L+1-i} \; b_i \; b_{L+1-i} = -1,$$

for i=1(1)L.

Proof: (by induction) When i = 1, the statement of the result may be verified by simply applying Theorem 3.1 to equation (2.1) with k = L-1 .

Assume then that the result holds for i = 1, 2, 3, ... j . To prove it for i = j + 1, consider equation (2.1) with k = L-j-1:

$$\begin{aligned} R_{AA}(L-j-1)+R_{BB}(L-j-1) &= a_1 \, a_{L-j} + \ldots + a_{j+1} \, a_L \\ &\quad + b_1 \, b_{L-j} + \ldots + b_{j+1} \, b_L \\ &= 0. \end{aligned}$$

Applying Theorem 3.1 yields

$$\begin{aligned} \prod_{i=1}^{j+1} a_i \, b_i \prod_{i=L-j}^{L} a_i \, b_i &= \prod_{i=1}^{j+1} a_i \, b_i \, a_{L+1-i} \, b_{L+1-i} \\ &= (-1)^j \, a_{j+1} \, b_{j+1} \, a_{L-j} \, b_{L-j} \\ &= (-1)^{j+1} \end{aligned}$$

Therefore,

$$a_{j+1} a_{L-j} \; b_{j+1} b_{L-j} = -1.$$

Hence, since the result holds for i = 1, and since the result will hold for i= j+1 if it holds for i = 1(1)j, j<L, the corollary must be true for i = 1(1)L.

Theorem 3.1 can be used to reduce the number of variables in a system. Each associated equation provides a convenient way of representing any variable on the left hand side in terms of the rest (simply by multiplying both sides of the equation by the selected variable). Unfortunately, the associated equations often reduce to the tautology 1 = 1. This circumstance is the catch which prevents Theorem 3.1 from being applied repeatedly.

In certain special cases, other deductions can be made through peculiarities in the set of coefficients. The following two theorems provide examples.

Theorem 3.2: If the multinomial equation

$$\sum_{j=1}^{n} c_j \; a_j = 0$$

has a coefficient $|c_j| = 0.5 \sum_{i=1}^{m} |c_i|$ for some j, then

$$a_i = -(c_i / |c_i|)(c_j / |c_j|) a_j ,$$

for i not equal to j.

Proof: As in the proof of Theorem 3.1, consider the left-hand side to be a sum of $\sum_{i=1}^{m} |c_i|$ binary terms. Due to the size of our one coefficient c_j, half of the terms already have the same sign. The rest of the terms must be of the opposite sign. The sign of a binary term x is given by the following expression.

$$s(x) = (x/ |x|)$$

Therefore,

$$(c_i a_i / |c_i a_i|) = -(c_j a_j / |c_j a_j|),$$

for i not equal to j,

and $a_i = -(c_i / |c_i|)(c_j / |c_j|) a_j$.

Theorem 3.3: If a multinomial equation

$$\sum_{j=1}^{m} c_j a_j = 0$$

has two coefficients c_i and c_j such that

$$|c_i| = |c_j| > L/4$$

then

$$a_i = -(c_i / c_j) a_j .$$

Proof: The two indicated terms must cancel, since together they supply over half of the binary terms.

Hence, $c_i a_i = - c_j a_j$,

and $a_i = -(c_i / c_j) a_j$.

In some equations, other special relationships among the coefficients may lead to equalities which can be used to eliminate one or more variables from the remaining equations. Eventually however, all such opportunities must be used up. If the system has not been completely reduced, then at that point another approach must be taken.

Most methods of solving non-linear systems of real equations will not work with multinomial systems of binary variables, because one is forced to assume either that the system is over-determined, or that the variables are discrete. However, as if to soften the blow, the method of trial- and-error can be applied, at least in theory, to binary systems. Therefore, when an impasse is reached with substitution, the situation can be saved by trying all possible sets of values for the remaining variables. However, this procedure will be time-consuming, since if t variables remain, then 2^t possible solution sets exist. As an intermediate option, it may sometimes be possible to assign values to only some of the remaining variables, thereby creating another reducible system. In general it is not always clear how this may be done, but for the equations which will arise in Section 3.3, this alternative will prove viable.

The strategy being outlined can be improved still more. Since the representation of a multinomial equation tends to be awkward and difficult to work with, a way has been found to effectively use Theorem 3.1, without actually storing a representation of the entire equation system.

A matrix of 0's and 1's, ideal for representation on a binary computer, is stored instead. This matrix keeps track of the effects of changing the values of the variables of the system. Accordingly, it is called a matrix of change.

The use of a matrix of change forces the introduction of a new concept. The residual of a multinomial equation will be the value of the left-hand side expression of the equation for a specific set of values of the variables. The process of solution of a multinomial system consists of forcing these residuals to be zero through successive modifications of the variables. The matrix of change will represent the changes of these residuals corresponding to changes in the variables.

The algorithm presented here relies upon modular arithmetic. The residuals will be forced to become zero congruent to a specific modulus, usually 2 or 4. If every multinomial of the system has an even number of binary terms, then the residuals must already be congruent to 0, mod 2. In that case, whenever the value of a variable is changed from 1 to -1, or back, every binary term in which it occurs will change its value by 2. If the variable occurs in an even number of binary terms in an equation, then the residual of the equation will change by some multiple of 4 when the variable is flipped. Otherwise, the residual will change by 2, modulo 4. The vector of change of a variable records whether that variable will change the value of each residual of the system, modulo 4. A 1 in the i'th position of the vector of change of the variable x implies that switching x will change the residual of equation i by 2, mod 4. A 0 implies that no such change will occur.

These results are recorded in the following theorem.

Theorem 3.4: The J'th component of the vector of change of x is congruent modulo 2 to $n(J;x)$, the number of binary terms containing x in the J'th multinomial expression.

Any set of variables will also have a vector of change, as shown by the following theorem and its corollary.

Theorem 3.5: A pair of variables, x and y, will have a vector of change found by taking the term-by-term sum modulo 2 of the vectors of change of x and y.

Proof: Without loss of generality, consider only the i'th component of the vector of change of the pair. It should be 1 if a change of sign of both x and y changes the value of the i'th residual, modulo 4. By Theorem 3.4, only if an odd number of terms change sign can this situation occur. The number of terms which change sign is given by the following expression:

$$n(J;x,y) = n(J;x) + n(J;y) - 2n(J;xy),$$

where $n(J;x,y)$ is the number of terms in which x or y, but not both, occur, $n(J;x)$ and $n(J;y)$ are the numbers of terms in which x and y appear, respectively, and $n(J;xy)$ is the number of terms in which both x and y appear. Obviously,

$n(j;x,y)$ is odd only if either $n(j;x)$ or $n(j;y)$ is odd, but not both. Therefore,

$$n(j;p) = n(j;x) + n(j;y) \pmod{2},$$

and since $n(j;x)$ and $n(j;y)$ are congruent respectively to the j'th elements of the respective vectors of change, the theorem holds.

Corollary: The vector of change of a set of variables S has an i'th component given by the sum modulo 2 of the i'th components of the vectors of change of all the variables in S, for $i=1(1)m$, where m is the number of equations in the system.

Proof: This corollary can be proven by induction on the number of elements in S.

The vectors of change of all the variables in a system of equations can be made the columns of a 0-1 matrix, the matrix of change of the system. For instance, the matrix of change of the example (3.2) used above is:

$$\begin{matrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{matrix}$$

where the order of the columns is obtained from the following ordering of the variables:

$$a_1 , a_2 , \ldots \; a_L , b_1 , b_2 , \ldots \; b_L$$

The matrix of change becomes the augmented matrix of change if another column is added to it containing one-half the values modulo 4 of the residuals corresponding to a trial solution. Assume that in the example a trial solution of all 1's is used. then the augmented matrix is:

$$\begin{matrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & & 1 \end{matrix}$$

A matrix of this sort can be reduced to congruent forms through the usual type of row operations, in this case over the binary field. Only two row operations are required: interchanging two rows, and adding one row to another, mod 2. With these, the matrix can be transformed into a row-echelon form .

Theorem 3.6: Through successive steps of interchanging two rows, or of adding one row to another modulo 2, the augmented matrix A, of a system with n equations in v variables, can be transformed into a row-echelon form which satisfies the following conditions.

1) If the pivotal element (the first 1 to appear from the left) in the i'th row occurs in the j'th column, then all the rows 1 to i-1 must have some 1, in columns 1 to j-1, for i=2(1)n.

2) If the pivotal element in the i'th row occurs in the j'th column, then no other 1 will occur in the j'th column, for i=1(1)n.

Proof: The two conditions can be made to apply to each column in turn, starting with the first. In order to set the first column into the appropriate form, first check for 1's in that column. If none exist then the conditions are satisfied. If some 1 occurs in column 1, move it into the first row, if necessary, through an interchange. Then use this pivotal element to eliminate all other 1's in the first column by adding row 1 to any other row which contains such a 1.

Proceeding inductively, suppose that the conditions are satisfied for all rows up to row i, and for all columns up to column j. The conditions can then be enforced on column j+1. First search rows i+1 to n of column j+1 for 1's. If none occur, then column j+1 already satisfies the requirements. If a 1 occurs in some row, it may be moved by interchange into row i+1, without disturbing the conditions on the first i rows and j columns. Then every other 1 in column j+1 may be eliminated by addition of row i+1 to the appropriate rows. This procedure will not violate the conditions in rows 1 to i, nor in columns 1 to j, since by condition 2 and the inductive hypothesis, row i+1 cannot have a 1 to the left of column j+1. Therefore, the conditions can be applied to column j+1, perhaps expanding at the same time the number of rows to which they apply to i+1.

Since the conditions can be made to hold for the first column, and since they will be made to hold for column j+1 if satisfied already for all columns up to j, then by induction, the conditions must be enforceable throughout the matrix, so that a row-echelon form may be created.

The row-echelon form for the matrix of change of a system facilitates solving the system. Finding a solution to a system of multinomial equations means locating a set of values (1 or -1) for the binary variables of the system, which will simultaneously satisfy all the multinomial equations. Solving the augmented matrix system for the same multinomial equations is a less demanding procedure. This solution need only specify those variables of the current trial solution which must be changed to make the modified trial solution a successful linear algebraic solution of the augmented matrix system in the binary field. The form of this result is a 0-1 matrix of length v which flags with 1's all the variables which must be changed in sign to reduce all the residuals to 0 in the binary field.

The row-echelon form of the sample matrix is:

1	0	0	0	0	1	1	0	0	0	0	1	1
0	1	0	0	1	0	0	1	0	0	1	0	1
0	0	1	1	0	0	0	0	1	1	0	0	1

Since there are three non-zero rows in the reduced matrix, it is clear that a solution, and indeed multiple solutions, can be found to the system of equations which has the first twelve columns of this matrix as a coefficient matrix, and the last column as a right-hand- side. A solution to this system is a vector of twelve 0's and 1's representing the variables which need to be changed to reduce the residuals of the right-hand-side to 0.

The solutions to this system can be represented by the following matrix. The first row is a solution, any solution. The rest of the rows are independent solutions of the corresponding homogeneous equation system. They form a basis of the kernel of the matrix. Any linear combination of the rows of the solution matrix, which must include a contribution from the first row of the matrix, will be a solution of the augmented matrix system. Furthermore, by standard theorems of linear algebra, used here with the binary field, all solutions of the system can be generated in this way.

$$
\begin{array}{cccccccccccc}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}
$$

These results are given more generally in Theorem 3.7.

Theorem 3.7: Let A be the augmented matrix for a system of n multinomial equations in v variables, corresponding to a trial solution $X = (x_i)$, $i=1(1)v$. Let B be the row-echelon form of A. If the last column of B contains a 1 with only 0's to the left of it, then the system of equations has no solution. Otherwise the following results hold:

1) $Y =(y_j)$ is a solution to the system where $y_j = 1$ if and only if column j contains the pivotal element of the i'th row, which has a 1 in its last column.

2) $Z_K = (z_{KL})$ is a vector of the kernel of the system if $z_{KL} = 1$ where either column l is the k'th column to contain no pivotal elements, or else column l contains the pivotal element in row j, and row j has a 1 in that previously-mentioned k'th pivot-less column. In fact, the set of vectors Z_K so formed form a basis of the kernel of

the system.

Proof: It is clear that if the only 1 in a row is in the last column, then no solution can exist, for even in the binary field, a sum of 0's does not equal 1.

Assume then that every 1 in the last column of B is preceded by some other 1. If Y is really a solution to this constrained B, then the product of the first v columns of B, taken as a matrix, and Y, should yield the last column of B. Consider the i'th component of this product. This component is the dot product of the vector formed from the first v elements of row i with the vector Y. Suppose that the j'th element of row i is 1, and that $y_j = 1$. This is the only case that could contribute a non-zero value to component i. By 1) of the theorem, column j must then contain the pivotal element of some row i'. But in a row-echelon matrix, by definition, column j will contain exactly the one 1 in row i, which must then be row i'. Since a row can have only one pivotal element, the dot product must then be 1. But by 1), the last element of row i is 1, and hence the values are equal.

Conversely, if the last element of row i is 1, it must be preceded by a 1 in some column j, and hence by 1), Y will have a 1 in component j, and as before, the dot product will equal the residual.

Therefore, Y must be a solution, since all other cases lead to 0's for both dot product and residual, and since i was chosen as a general row.

Similarly, it can easily be checked that Z_K yields a zero vector when multiplied by the first v columns of B. Again consider the i'th component of the product. Either zero or two 1's will contribute to this component. Zero 1's will occur if the i'th row of B is a zero row, or if row i has no 1 in column l, which is the k'th column to contain no pivot elements in any row. Otherwise, two 1's will be summed, one for the match by 2) between Z_K and the i'th row in the l'th column, and the other from the match between Z_K and the pivot element of the i'th row.

Furthermore, the set of Z_K 's form a basis of the kernel of the system. That they are independent rests upon the fact that each has a 1 in a component which is 0 in all the others, namely that component which corresponds to a column with no pivot element. That they generate the entire kernel can be seen from the fact that the set of vectors formed from the non-zero rows of B by zeroing all but the pivot elements in these rows form a basis to the complementary subspace to the kernel, in the binary field of dimension v.

The matrix of change will prove to be a powerful tool in the next section, and in Chapter 4.

3.3 An Algorithm for Finding Complementary Sequences

The methods of the previous section can be applied to the problem of finding complementary sequences of a specific length L. Due to the form of the equations, the procedures previously described will be modified somewhat to allow for the association of variables into constructs called quads. Use of the quads reduces the number of cases to be considered to $2^{L/2-1}$. Further reductions may be possible with different, but equivalent equation systems.

The problem consists of finding a solution for the system of equations given in (2.1). The application of Theorem 3.1 to this system yields the result known as Theorem 2.6. Basically this theorem states that every group of four variables found by taking the i'th variables from each end of both sequences can have at most eight possible states. For convenience, each such group of four variables will be called a quad. The eight possible quads are listed below with the symbols which will be used to refer to them.

$$Q = \begin{bmatrix} 1 & - \\ 1 & 1 \end{bmatrix} \quad q = \begin{bmatrix} - & 1 \\ - & - \end{bmatrix} \quad R = \begin{bmatrix} 1 & 1 \\ 1 & - \end{bmatrix} \quad r = \begin{bmatrix} - & - \\ - & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} - & 1 \\ 1 & 1 \end{bmatrix} \quad t = \begin{bmatrix} 1 & - \\ - & - \end{bmatrix} \quad S = \begin{bmatrix} 1 & 1 \\ - & 1 \end{bmatrix} \quad s = \begin{bmatrix} - & - \\ 1 & - \end{bmatrix}$$

Since there are eight quads, it seems reasonable to use three 0-1 bits to describe each quad. In effect, this description consists of a change of variable from the original 2L variables to the new 1.5L variables. The old notation will be called the matrix form, and the new notation, the octal form of a quad. The three new quantities will describe the properties of the single sign in each quad which differs from the rest, the odd sign. The components have the following meaning:

Component 1 - Up-down orientation bit (UDOB): 1 if the odd sign is in the first row; 0 otherwise.

Component 2 - Left-right orientation bit (LROB): 1 if the odd sign is in the first column; 0 otherwise.

Component 3 - Sign bit (SB): 1 if the odd sign is "-" ; 0 otherwise.

In the octal notation, the eight quads are described as follows:

Q=(1,0,1)	q=(1,0,0)	R=(0,0,1)	r=(0,0,0)
T=(1,1,1)	t=(1,1,0)	S=(0,1,1)	s=(0,1,0)

A pair of complementary sequences of length L can now be described as a sequence of 0.5L quads. The equations which must be satisfied by a complementary sequences can also be given in terms of quads, but first a multiplication operation must be defined on quads.

The quad product of two quads A and B is a scalar. Its value is found by first multiplying together the corresponding terms of the matrix forms of the two quads, and then summing. The sum is divided by 2 to get the final quad product. For example, using "x" to indicate the quad product,

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} e & f \\ g & h \end{bmatrix} = (ae + bf + cg + dh)/2$$

Several observations can be made. First, the quad product of any quad with itself is 2. Secondly, the quad product of any quad with another which differs from it only in the SB is -2. Finally, the quad product between any other two quads is 0.

One can also define the conjugate of a quad to be the quad obtained by flipping the LROB. As an example, using the symbol "*" to represent conjugation:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{*} = \begin{bmatrix} b & a \\ d & c \end{bmatrix}$$

Using these operations, one can set up the quad equations which must be satisfied by a quad sequence to generate complementary sequences.

Theorem 3.8: Let $X = (x_i)$, for $i=1(1)L$, be a sequence of quads, with $x_{L+1-j} = x_i{}^*$ for $j=1(1)0.5L$. Then X generates complementary sequences if the following equations are satisfied for $k=0, 1, \ldots L-1$:

$$R_{XX}(k) = \sum_{i=1}^{L-K} x_i \, x_{i+K} = 2L\,\delta_{0K} ,$$

where the symbol for the quad product has been suppressed for convenience.

Proof: By definition, the summation

$$\sum_{i=1}^{L-K} x_i \, x_{i+K} = \sum_{j=1}^{L-K} (a_j \, a_{j+K} + a_{L+1-j}\, a_{L+1-j-K} + b_j \, b_{j+K} + b_{L+1-j}\, b_{L+1-j-K})/2$$
$$= 2 \sum_{j=1}^{L-K} (a_j \, a_{j+K} + b_j \, b_{j+K})/2$$
$$= 2L\,\delta_{0K} .$$

In the last section it was observed that after application of Theorem 3.1 to a system of equations, assumptions would have to be made about the remaining variables before further progress could be made in finding the solution of the equations. In an exhaustive search of all possible sequences pairs of length L, this is the stage at which these assumptions must be made. In the programs listed in Appendix 1, values are given to the LROB's of each independent quad. The number of quads being 0.5L, the number of possible allocations of LROB's is $2^{0.5L}$. Without loss of generality, however, the first quad can be given a LROB of 1 in each case, cutting the number of possibilities in half.

Once a set of choices have been made for the values of the LROB's of the quads in the sequence, the algorithm for reducing the equations has three parts. This algorithm is:

Algorithm 3.1

Given: A set of LROB's which partially specify a quad sequence.

Required: To find any complementary sequence which has that LROB pattern in its quad sequence.

Procedure:

1) Use matrix reduction to find a set of UDOB's which satisfy the equations 2.1 modulo 4.

2) Use matrix reduction to find a set of SB's which satisfy the equations modulo 8.

3) Try all possible quad sequences which pass steps 1 and 2 to see if they do indeed generate complementary sequences.

It is perhaps surprising to find that the same technique of matrix reduction which works on binary variables also can be used to determine the up-down orientation of the quads, once their left-right orientation has been selected. The quad products can take on the three values 2, 0, and −2. Both 2 and −2 will contribute 2 to any equation, modulo 4. Therefore, changing the up-down orientation of a quad, which will change modulo 4 the value of all the terms in which it appears, will change some of the residuals of the equations, again modulo 4. It is possible to set up a vector of change for the UDOB of each quad, for, as in Theorem 3.5, the i'th component of the vector of change is congruent mod 2 to the number of terms in the i'th quad expression in which that particular quad occurs. Together, these vectors of change can be formed into a matrix of change. Then, if the trial values of the UDOB's are introduced, an augmented matrix of change can be set up.

The row-echelon form of the augmented matrix of change reveals the same information about the UDOB's as it did about binary variables in Section 3.2. Therefore, either the initial LROB distribution will be discredited, or a successful set of UDOB's will be found.

It may also happen that the system of equations has a non-null kernel. In this case multiple solutions will exist, multiple sets of UDOB's which must be checked separately for the complementary sequences they may conceal.

The rank of the kernel can be diminished by eliminating any vector which leads to solutions isomorphic to ones already found. Since conjugating every quad in the system will change none of the residuals -- the quad product depending in value on similarities between its arguments, and not on their actual values -- the vector with all 1's will be one of the vectors which can be removed from the basis of the kernel. This task can be easily and automatically accomplished by adding an extra equation to the system, asserting that the first UDOB be 0.

In the next step, the process of reducing the residuals to 0 modulo 8 reverts exactly of the binary system of equations. At this point, with both LROB's and UDOB's known, all the 0 terms can be identified. The only terms retained in the equations are the non-zero ones, with values of 2 or -2. The variables of interest are the sign bits of the quads, for a given quad product will be positive only if the sign bits of the two quads agree. The SB's themselves can take on only two values (0 or 1), and so the same matrix of change technique can be used again.

Once more the row-echelon form of the augmented matrix of change will indicate whether any distribution of SB's will force all the residuals to go to 0 mod 8. Once again, if a solution to the system of equations exists, at least one plausible sequence must be tested. As before, the rank of the kernel is the number of columns of the row-echelon form of the augmented matrix which do not contain any pivot elements. Any set of SB's which may produce a pair of complementary sequences must consist of a solution to the equations, summed with any vector from the kernel.

On this occasion, the rank of the kernel can be reduced by 2. First, changing all the sign bits can not change any of the residuals. Secondly, changing the sign bits of all the quads with 1 in the UDOB can not change any of the residuals. These statements hold because all changes in the resulting autocorrelation functions exactly cancel. As a consequence, without losing any solutions not isomorphic to those which will still be found, two equations can be added to the system. They are:

$t_0 = 0,$

and $t_k = 0,$

where t_i is the i'th SB, and k is the index of the first non-zero UDOB. It should be remembered that the first UDOB has been forced to 0, so no conflict can occur.

Usually by the time that step 3 of Algorithm 3.1 has been reached, arguments of the type used in Theorems 3.2 and 3.3 suffice to define very quickly the solution set. Such methods are difficult to program, however, and so it has been found easier to use an exhaustive search in step 3. Experimentation has shown that for sequence lengths short enough to make computer searches possible, this shortcut does not prove prohibitively expensive. In fact, all non-isomorphic complementary sequences of lengths L=2,4,8,10,16,20,26,32 have been found in this way, using the computer programs of Appendix 3. They are listed in Appendix 2. A faster, but even less readable version of this program has been used to show that no sequences exist of lengths 34, 36, and 50.

3.4 Synthesis of New Complementary Sequences .

Not all complementary sequences must be found by exhaustive search. New complementary sequences can be created from old with relative ease. Some of the methods for synthesizing sequences were developed by Golay. Others were added by Turyn. Whatever the source, they all depend upon the quality called orthogonality between two pairs of complementary sequences.

Two pairs of complementary sequences, (A,B) and (C,D) are said to be orthogonal if

$$R_{AC}(k) + R_{BD}(k) = 0$$

for k= ... -2,-1,0,1,2, ... Such orthogonal mates, as they have been called, can be easily constructed.

Theorem 3.9: If A and B are complementary sequences, then B and -A' are also complementary sequences, and the pair (A,B) is orthogonal to the pair (B,-A'), where -A' is the negative of the sequence A, in reverse order. (The usage of the symbol "′" here differs from that of Golay.)

Proof: The pair of sequences B' and $-A'$ are complementary by application of Theorems 2.2, 2.3, and 2.4. That they are orthogonal to A and B follows from the following identity.

$$R_{AB'}(k)+R_{B(-A')}(k)=\sum_{i=-\infty}^{\infty}(a_i\,b_{L+1-i-k}-b_i\,a_{L+1-i-k})$$

Orthogonal mates can be fitted together in various ways to produce complementary sequences of twice the length.

Theorem 3.10: If (A,B) and (C,D) are orthogonal mates then the sequences AC and BD, formed by concatenation of the indicated sequences, are complementary.

Proof: The autocorrelation of AC for a displacement k can be split up into the following expression:

$$R_{AC,AC}(k) = R_{AA}(k) + R_{CC}(k) + R_{CA}(L-k)$$

This decomposition follows directly from the definition of autocorrelation. But since (A,B) and (C,D) are orthogonal complementary pairs, the following relation will hold.

$$\begin{aligned}R_{AC,AC}(k) + R_{BD,BD}(k) &= (R_{AA}(k) + R_{CC}(k)) \\ &\quad + (R_{BB}(k) + R_{DD}(k)) \\ &= 4L\,\delta_{0k}\,.\end{aligned}$$

The two sequences can also be combined by interleaving.

Theorem 3.11: If (A,B) and (C,D) are orthogonal mates, then the interleaved sequences

$$E = (a_1 \ c_1 \ a_2 \ c_2 \ \ldots \ a_L \ c_L)$$

$$F = (b_1 \ d_1 \ b_2 \ d_2 \ \ldots \ b_L \ d_L),$$

where $A =(a_i)$, $B =(b_i)$, $C =(c_i)$, and $D =(d_i)$, are complementary.

Proof: Consider the autocorrelation functions when k is even.

$$\begin{aligned} R_{EE}(k) + R_{FF}(k) &= (R_{AA}(k/2)+R_{CC}(k/2)) \\ &\quad + (R_{BB}(k/2) + R_{DD}(k/2)) \\ &= 4L\delta_{0K} . \end{aligned}$$

When k is odd,

$$\begin{aligned} R_{EE}(k)+R_{FF}(k) &= R_{AC}((k-1)/2)+R_{CA}((k+1)/2)+ \\ &\quad R_{BD}((k-1)/2)+R_{DB}((k+1)/2) \\ &= 0. \end{aligned}$$

Use of these theorems relies upon having short sequences to work with. Since only sequences of length 2, 10, and 26 have been found by other means, all known complementary sequences have lengths given by the formula $L= 2^i \, 10^j \, 26^K$ for any non-negative integers i, j, and k.

Chapter 4: SCOSoV's

4.1 Definition and General Properties of SCOSoV's

The definition of complementary sequences requires such simple concepts that generalization seems to follow naturally after any study of them. Several generalizations have already been attempted [19][13]. In this chapter is developed a previously unexamined extension to the idea of complementary sequences. The basis of this extension lies in generalizing the notion that the elements of a sequence should be scalars. Allowing the sequence elements to be themselves vectors incorporates all the possible structures permitted by the original definition of complementary sequences, with room for innovation.

In order that the sequences to be discussed will be well-defined, the following set of axioms will be used to describe them.

1. A SCOSoV is one sequence of vectors drawn from a discrete set called the basis of the sequence.

2. Every vector in the basis can be used with a coefficient of either 1 or -1.

3. All the vectors in the basis are mutually orthogonal, according to some scalar product.

4. The norm of each vector in the basis, found using the same scalar product , is 1.

5. The following self-complementary property must be observed by a SCOSoV $X = (x\ , x\ , \ldots\ x\ )$ of length L.

$$R_{XX}(k) = \sum_{i=-\infty}^{\infty} x_i\, x_{i+k} = L\, \delta_{0k}$$

Although the definition differs considerably from the definition of complementary sequences, it can easily be shown that every pair of complementary sequences actually defines a SCOSoV.

Theorem 4.1: If $A=(a_1, a_2, \ldots a_L)$ and $B=(b_1, b_2, \ldots b_L)$ are complementary sequences of length L, then the sequence X of L vectors formed by taking $x_i = (a_i, b_i)$, for i=1(1)L, forms a SCOSoV.

Proof: First, let the basis of a SCOSoV be the set S = [(1,-1), (1,1)]. Let the scalar product between two vectors be the ordinary dot product, divided by 2 . That is, (a,b).(c,d) = (ac + bd)/ 2 . Observe that the vectors of S are orthogonal, since (1,-1).(1,1) = (1,1).(1,-1) = 0. Furthermore, the norm of each vector is

(1x1 + (-1)x(-1))/2 = (1x1 + 1x1)/2 = 1.

Every vector in the sequence X either belongs to S, or has an additive inverse which belongs to S, because every a_i and b_i is either 1 or -1. Finally, the complementary property also holds, since

$$R_{XX}(k) = (R_{AA}(k) + R_{BB}(k))/2 = \delta_{0K} L.$$

Therefore, the sequence X is a SCOSoV, since it satisfies all the axioms.

Corollary: Any SCOSoV X of length L based on the vector set S=(u,v) will generate a pair of complementary sequences.

Proof: Let u=(1, -1), v=(1,1), and simply reverse the steps of the proof of the theorem.

If the number of vectors in S is increased beyond 2, however, neither the analogue of the theorem nor of the corollary will be true. For instance, a set of four complementary sequences may be interpreted as a sequence of 4-element vectors. It may happen, however, that the vectors so formed are not mutually orthogonal. On the other hand, a basis may be defined with three mutually orthogonal vectors in it. But due to the following theorem it is not possible to construct a set of three complementary sequences from it. (A set of four complementary sequences can be formed without difficulty in this case.)

Theorem 4.2: No two m-tuples with components 1 and -1 can be orthogonal under the usual dot product if m is odd.

Proof: The dot product of two such vectors is a sum of m terms, each of which is 1 or -1. Since the number of terms is odd, the sum cannot be even, and it is therefore not 0.

The use of SCOSoV's in place of complementary sequences permits a closer examination of the structure of the sequence, as opposed to the actual elements which occur in it. Thus, if the basis is the set S = (r,s), which may be assumed to satisfy the requisite axioms, then the structure of the SCOSoV will be the same whether r = (1,1) and s = (1,-1), or r = sin(t) and s = cos(t) and the scalar product of f(t) and g(t) is

$$f(t).g(t) = \int_0^{\pi} f(t)g(t)\,dt.$$

At the same time, the benefits of using binary variables, which were developed in Chapter 3, have not been lost, since the coefficients of the vectors r and s are still 1 and -1.

Actually, much has been gained, since each special case of a SCOSoV, found by using a specific basis, will have its own qualities. In some cases these qualities will facilitate discussion of a point which would have otherwise remained unclear. For instance, the following theorem can be easily proved by referring to a specific basis.

Theorem 4.3: If $X = (x_1, x_2, \ldots x_L)$ is a SCOSoV of length L based upon the vector set $S = (s_1, s_2, \ldots s_L)$, then L is the sum of m integer squares.

Proof: As in the proof of Theorem 2.7, take the following sum of autocorrelation functions.

$$R_{XX}(0) + 2R_{XX}(1) + 2R_{XX}(2) + \ldots + 2R_{XX}(L-1) = L.$$

Expanding each autocorrelation function into its components yields

$$(x_1 + x_2 + \ldots + x_L).(x_1 + x_2 + \ldots x_L) = L.$$

Let x be the i'th row of an $m \times m$ identity matrix, and define the scalar product to be the usual vector dot product. Then if

$$u = x_1 + x_2 + \ldots + x_L,$$

$$u.u = L,$$

and therefore $u_1^2 + u_2^2 + \ldots + u_m^2 = L$, where u_j is the j'th component of u. Since the components of all the x_i's were 0, 1 or -1, the components of u must be integers, and the theorem has been proved.

Unfortunately, this result is very weak due to an old theorem of Fermat [12], which states that any positive integer is the sum of four integer squares. Only for the case m=3 can the theorem be used to disqualify sequence lengths not previously discarded.

A certain amount of care must be exercised in translations between the general and the specific. In some instances properties of specific SCOSoV's do not have a general application. In others, as suggested by Theorem 4.2, the basis needed to prove a point may not exist.

Many of the theorems that have been proven for complementary sequences can be extended to cover SCOSoV's as well. In most cases changes have to be made.

Theorem 4.4: The following transformations will always convert a SCOSoV X defined on a basis $S=(s_1, s_2, \ldots s_m)$ into a similar SCOSoV.

1. Rearranging the elements of S.
2. Negating Some of the elements of s.
3. Reversing the SCOSoV X.

Proof: The first two alterations obviously just transform the basis S into another S′ without changing any of its properties. The third condition is due to the symmetry of $R_{XX}(k)$, which is the same as $R_{XX}(-k)$.

Theorem 4.5: If $X = (x_1, x_2, \ldots x_L)$ is a SCOSoV of length L over the basis $S=(u,v)$, then $x_i x_{L+1-i} = 0$, for $i=1(1)L$.

Proof: This theorem is a direct consequence of Theorem 2.6. In this case, the proposition, which has been proven only for complementary sequences, will also apply to all SCOSoV's of the stipulated form. If for some other u and v a SCOSoV could be established which violated Theorem 4.5, then by the corollary to Theorem 4.1, a pair of complementary sequences could also be constructed which violated Theorem 2.6.

Unfortunately, Theorem 4.5 holds only when the basis S has just two vectors. The condition could be asserted as one of the axioms for larger bases, either to strengthen the ties between SCOSoV's and complementary sequences, or to simplify the problem of finding SCOSoV's, by reducing the number of possibilities which have to be checked.

4.2 Generating SCOSoV's

A SCOSoV will exist for any length L, provided only that the basis is big enough. Therefore, the problem of finding SCOSoV's concerns itself with finding vector sequences based on the smallest possible vector set. In many respects, this task differs from the problem of finding as many sequences as possible over a selected basis, which was attempted with complementary sequences. For example, partial solutions are possible in the search for a minimum basis. Every time the size of the basis is reduced for a particular L, a little victory is won. In contrast, the search for complementary sequences has offered little satisfaction, for a solution will either exist, or it will not.

Although several synthesis theorems will be given, the nature of a search from first principles for SCOSoV's will be discussed first, since the existence of a large set of solutions for small L will facilitate the synthesis of longer SCOSoV's.

A quick survey of what is possible leads to the following table, where the theoretically minimum basis size is listed opposite the sequence length for L varying from 1 to 20.

Table of Minimum Possible Basis Sizes

Sequence Length	Basis Size
1	1*
2	2*
3	3*
4	2*
5	3*
6	3*
7	4*
8	2*
9	3*
10	2*
11	3*
12	3
13	3
14	3
15	4
16	2*
17	3*
18	3
19	3
20	2*

The asterisks in this table indicate those cases in which minimal solutions have been found. For some quite small sequence lengths, such as 12, the answers are still unknown. For some of these cases, the cost of an exhaustive search is not prohibitive. For instance, with L=12, there exist $(3\times2)^{12}$ possible vector sequences of length L on a basis with three vectors. However, using the methods of Section 3.2, the signs of the vectors can be found as the solution of a matrix equation. The number of possibilities is thus reduced to 3^{12}. Without loss of generality, the choice for the first vector can be made before seeking the SCOSoV, and then the last vector of the sequence can be chosen from the two remaining vectors. It is clear that the first and last vectors must be orthogonal to make $R_{XX}(11) = 0$ for the SCOSoV X. The number of possible sequences has then been reduced to 2×3^{10} = 118098.

Even this manageable number is not the final figure. Suppose the basis is the set S=(u,v,w). Assume that the first vector of the SCOSoV X is u. Form a sequence of 0's and 1's representing the locations of the other u's in the sequence, 11001011l000, for example. The last element must be a 0. There are 2^{10} = 1024 such sequences. Now the vectors with 0 indicators can be either v or w. This is a 2-way choice, and changing the decision for any one vector will change the value of any term it appears in by 1 or -1, which are both congruent to 1 mod 2. (It is assumed that terms containing a u and a v or w have already been discarded.) Hence, any vector with a 0 indicator has a vector of change corresponding to the equations that the SCOSoV must satisfy. The vectors of change can be combined into a matrix of change, which can be reduced into row-echelon form, yielding a set of solutions for the v-w choices. Then the signs can be determined by a similar procedure. Consequently, only 1024 sequences, for which the u positions have been specified, need to be considered in the search for a SCOSoV of length 12 on a 3-vector basis.

The following theorem serves to reduce this number still more. It is the general analogue to Theorem 2.6 on complementary sequences. It is not as strong as is Theorem 2.6, but it aids considerably in keeping down the possibilities.

Theorem 4.6: Let $X = (x_1, x_2, \ldots x_L)$ be a SCOSoV of length L based on the vector set $S = (s_1, s_2, \ldots s_m)$, where $m>1$. Let R_i be the set $(x_1, x_2, \ldots x_i)$, for $i=1(1)0.5L$. Denote the complement of R_i in S by R_i'. Then if the set $(x_{L+1-i},$ $\ldots x_{L-1}, x_L)$ is a subset of R', the element x_{L+1-i} must be orthogonal to x_1.

Proof: Suppose $R_i = (x_1, x_2, \ldots x_i)$, for $i \leq 0.5L$. Consider

$$R_{XX}(L-i) = x_1 \cdot x_{L+1-i} + x_2 \cdot x_{L+2-i} + \ldots x_i \cdot x_L = 0.$$

If $(x_{L+2-i}, \ldots x_{L-1}, x_L)$ belongs to R_i', then this equation reduces to a one-term expression:

$$R\ (L-i) = x_1 \cdot x_{L+1-i} = 0,$$

which proves the theorem.

The significance of this theorem lies in the following application. Consider the 0-1 sequence used before to represent the positions of u's in the SCOSoV for L=12. It was 110010111000. Notice that this sequence begins with two 1's. By Theorem 4.6, since the last element of the sequence is 0, the last two elements must both be 0. That is, whereas this sequence might generate a SCOSoV, the sequence 11001011101O definitely would not. As another example, the sequence 11111000000 is the only possible acceptable SCOSoV structure for L=12 to begin with six u's, according to Theorem 4.6 (applied repeatedly). Eliminating all the sequences which violate this type of restriction leaves only 683 to be tried.

The following algorithm applies the reasoning given above for length 12 and basis size 3 to the general case where the length of the SCOSoV is L and it is based on a vector set of size m.

Algorithm 4.1:

Given: A length L and a basis set of orthogonal vectors S $= (s_1, s_2, \ldots s_m)$.

Required: To find some SCOSoV X of length L based on S, if any such SCOSoV's exist.

Procedure:

1. Let n be the integer part of (m+1)/2. Divide the basis S into n subsets of 1 or 2 vectors each. Let these be $v_1 = (s_1, s_2)$, $v_2 = (s_3, s_4)$, ... $v_n = (s_{m-1}, s_m)$ for n

even. If n is odd, then change the definitions so that v_1 = (s_1), and so that the remaining basis vectors are allocated two each to the rest of the v's.

2. Form all possible sequences of L integers chosen from the set (1, 2, ..., n). Each of these integers must occur at least once in each sequence. Without loss of generality, the sequences can be renumbered so that initial occurrences will come in increasing order. They will then be said to be in canonical form. For instance, 11213214 is such a sequence for L=8, n=4, but neither 13112143 nor 12112332 are. Each sequence represents a SCOSoV, and the integers represent the corresponding v_i's formed in step 1 of this procedure. If an integer sequence is $t = (t_1, t_2, \ldots t_L)$, then in the corresponding SCOSoV, the i'th element must be a vector coming from v_{t_i}, for i=1(1)L. If m is odd, do not accept sequences which do not satisfy Theorem 4.6. That is, if a sequence begins with three 1's, if m is odd, then that sequence cannot have a 1 in its last three elements.

3. For each canonical t-sequence, set up a matrix of change mod 2 under the assumption that the i'th vector of the sequence X will be permitted to alternate between the two vectors in v_i, for i=1(1)L, but that no other changes will be allowed. It will have r = L-1 rows representing the L-1 autocorrelations. The augmented matrix

of change, A, will have L+1 columns, one for each vector in the t-sequence, and one for the residuals. Not all of these columns are needed if m is odd, because then one of the v_i consists of only one vector, and its contributions are therefore fixed.

4. Let B be the row-echelon form of A. By the methods of the last chapter, the row-echelon form can be analyzed to yield a 0-1 solution vector, if one exists, along with a kernel, generated by a basis of 0-1 vectors which have each a zero vector of change. Each possible solution which can be constructed thereby represents a set of choices between the first and second members of each v_i. (Some of these choice sets will be isomorphic. For instance, if everywhere that the first element of v is chosen, the second element is substituted, and vice versa, it is clear that the resulting solution is an independent solution, different from, but equivalent to, the first. As discussed previously, it is possible to guard against such isomorphic solutions when setting the matrix of change A, by inserting extra equations into the system which require that, for instance, the first vector chosen from the set v_i will actually be the first element of v_i.) Define a set of u-sequences which will represent all the possible solutions found. Each u-sequence will consist of L integers, between 1 and m. It will represent one acceptable sequence of vectors of S.

5. Let D be a matrix of change mod 4 indicating the consequences of changing the signs of the vectors fixed by each u-sequence. It will have L-1 rows and L+1 columns. The last column of residuals in this case will be found by dividing the actual residuals (which are now known to be even) by 2, before taking them modulo 2.

6. Let E be the row-echelon form of D. Once again, the row-echelon form will verify the existence of solutions, if any exist. Furthermore, all these solutions can be expressed as linear combinations of the one solution vector, and also of the vectors from the kernel of the system. Once more, certain isomorphic solutions may appear, unless equations were added to the original system to ensure that one condition holds for the set of signs which are used for coefficients of occurrences of the vector s_i, for each i. All the sign sets which emerge as solutions at this point can be used to generate the set of x-sequences. These represent complete potential SCOSoV's, and they may be tested one-by-one to see if any of them are actually SCOSoV's. If so, the search has ended successfully; otherwise, the next u-sequence should be tested.

It is important to be able to estimate the amount of work involved in applying Algorithm 4.1. The number of canonical t-sequences which must be generated should provide a good measure of the work involved. Let $f(n,L)$ be the number of canonical sequences of length L on n integers when $m=2\times n$. Let $g(n,L)$ be the number of canonical t-sequences of length L on n integers when $m=2\times n-1$. Since in this case m is odd, the number j of 1's at the beginning of a sequence must be matched by an equal number of elements at the end of the sequence which are not 1, in order to satisfy Theorem 4.6. Then $f(n,L)$ and $g(n,L)$ can be found recursively from the following theorem.

Theorem 4.7: The sequences $f(n,L)$ and $g(n,L)$ must satisfy the following difference equations for $n>1$ and $L>n-1$:

(4.1) $\quad f(n,L+1) = nf(n,L)+f(n-1,L),$

(4.2) $\quad g(n,L+1) = ng(n,L)+g(n-1,L)+(-1)^{L+1}f(n-1,k),$

where k is the integer part of $(L+1)/2$. Furthermore, the functions $f(n,L)$ and $g(n,L)$ must satisfy the following initial conditions:

(4.3) $\quad f(1,L)=1 \quad$ and $\quad f(n,n)=1,$ for $n>0$;

(4.4) $\quad g(1,L)=\delta_{L1} \quad$ and $\quad g(n,n)=1,$ for $n>0$.

Proof: Let T be a t-sequence of length $L+1$ on the integers $1,2, \ldots ,n$. If m is even, then consider the sequence remaining after removing the last element t_{L+1}. If all n values occur in the truncated sequence, then it must be a t-sequence of length L on $1,2, \ldots ,n$. Since t_{L+1} can take on n possible values, $nf(n,L)$ sequences occur in this fashion. Alternatively, the truncated sequence may have only integers 1, 2, ... , n-1. Then $t_{L+1} = n$, of necessity, and only $f(n-1, L)$ sequences can be found in this way. Since these two are the only possibilities, then (4.1) follows.

On the other hand, suppose m is odd. Then t_{L+1} at least, and perhaps some elements prior to t_{L+1} cannot be 1, because of Theorem 4.6. Again delete t_{L+1} and consider the remaining sequence. If it is a canonical t-sequence of length L on n integers, then t_{L+1} can take on n-1 possible values. If it is a canonical t-sequence on only n-1 integers, then t_{L+1} must be n. Altogether $(n-1)f(n,L) + g(n-1,L)$ sequences will be formed in this manner.

This time, however, there are other possibilities. After deleting t_{L+1}, the remaining sequence may not be in acceptable form because a 1 is too close to the end of the sequence. Only one 1 can be violating Theorem 4.6, since only one element was removed from the sequence. Removing this 1, and replacing t_{L+1} at the end of the sequence will then produce another sequence of length L on 1, 2, ... , n. Another $g(n,L)$ sequences fall into this category. However, exceptions occur for those sequences with a maximal length string of 1's at the beginning. For instance, 112 is illegal, even though it can be formed by the last method mentioned, since deleting the last 1 leaves the legal sequence 12. Thus from the last $g(n,L)$ sequences found must be subtracted $f(n-1,k)$, with k defined as before, when L is even. On the other hand, when L is odd, the following type of duplication occurs. The sequences 1122 and 1212 are both valid, and both belong to the category where deleting the last 2 leaves an invalid sequence, but where deleting the last 1 leaves the acceptable sequence 122. Double solutions occur only when L is odd, and the initial string of 1's is maximal. The number of sequences gained is f(n-1,k), since k elements remain to be chosen.

Assembling the results for all these cases yields (4.2).

The initial conditions are found trivially. A sequence with only 1's can occur in only one way (e.g. 1111111 for L=7), and if m is odd, these are all illegal except for 1. Whatever m is, a sequence with n entries, in canonical form, from the integers 1 to n, can only be 1 2 3 ... n.

From this theorem it is possible to form the following simple expression for f(n,L).

Theorem 4.8: For n>0, L>0, f(n,L) is given by

(4.4) $f(n,L) = (\Delta^{n} 0^{L})/n!$,

where the differences of zero are

$$\Delta 0^{L} = 1^{L} - 0^{L}$$

$$\Delta^{2} 0^{L} = 2^{L} - 2 \times 1^{L} + 0^{L}$$

... .

Proof: The initial conditions (4.3) can be readily verified. First

$$f(1,L) = (\Delta 0^L)/1! = 1^L - 0^L = 1.$$

Secondly,

$$f(n,n) = (\Delta^n 0^L)/n! = n!/n! = 1.$$

This identity holds because of a standard theorem of algebra which states that the n'th difference of an n'th degree polynomial is n! times the coefficient of the highest degree term of the polynomial.

It suffices therefore to prove that (4.4) satisfies the difference equation (4.1). That is, it must be shown that

$$(n\Delta^n 0^L)/n! + (\Delta^{n-1} 0^L)/(n-1)! = (\Delta^n 0^{L+1})/n!.$$

Normalize the numbers by multiplying by n!, and then

$$\begin{aligned}
n\Delta^n 0^L + n\Delta^{n-1} 0^L &= n\Delta^{n-1}(\Delta+1)0^L \\
&= n\Delta^{n-1} 1^L \\
&= n\sum_{k=0}^{n-1} \begin{bmatrix} n-1 \\ k \end{bmatrix} (n-k)^L \\
&= \sum_{k=0}^{n-1} (n-k) \begin{bmatrix} n \\ k \end{bmatrix} (n-k)^L \\
&= \sum_{k=0}^{n} \begin{bmatrix} n \\ k \end{bmatrix} (n-k)^{L+1} \\
&= \Delta^n 0^{L+1}.
\end{aligned}$$

These steps use the binomial expansion of $\Delta = (E-1)$, where E is the forward shift operator, and certain properties of the binomial coefficients.

It has been shown that f(n,L) has a simple closed form. No convenient way of expressing g(n,L) has yet been found. For $n=2$ and $n=3$, the following expressions hold, as may easily be verified.

$$(4.5) \qquad g(2,L) = \frac{2^L}{6} + \frac{(-1)^L}{3}$$

$$(4.6) \qquad g(3,L) = \frac{2\times 3^{L-1}}{7} - \frac{2^L}{6} - (-1)^{L+1}\left\{\frac{1}{3} - \frac{p_L\, 2^{k-1}}{7}\right\}$$

In these formulae, k is again taken as the integer part of one-half $L+1$, and $p_L=1$ for even L; $p_L=2$ for odd L.

For large L, one has the following asymptotic result.

Theorem 4.9: For $n>0$, $L>>0$,

$$(4.7) \qquad g(n,L) \sim (n-1)n^{L-1} / (\ (n-2)!(n^2-n+1)\ ).$$

This expression is an asymptotic upper bound to $g(n,L)$.

Proof: It is clear from (4.5) and (4.6) that this asymptotic relation holds for $n=2$ and for $n=3$. For $n>3$, consider very long sequences with $L>>n$. Rough counting arguments will compute approximately the number of canonical sequences of length L for the given n.

First consider the sequences which commence with 1 2 3 ... $n-1$, n. This arrangement leaves $L-n-1$ positions with n choices, and one with $n-1$. Thus there are $(n-1)n^{L-n-1}$ such sequences. Move the first n one position further to the right. The gap can be filled in $(n-1)$ ways, and one position with n choices has been lost. Another $(n-1)^2 n^{L-n-2}$ sequences can be formed. Continuing to move the first n to the right produces a diminishing number of new sequences, which sums as follows:

$$(n-1)n^{L-n-1}\ (1+p+p^2+p^3+\ \dots\ )=(n-1)n^{L-n}$$

where p is the fraction $(n-1)/n$. This sum has been treated as an infinite series, which is not correct, but for very large L, the error will be small. This expression gives the number of sequences beginning with 1 2 3 ... n-1.

A similar argument holds when the first n-1 is shifted one position at a time to the right. This time, a new position with (n-2) choices is created, and an old position with n choices destroyed at each shift. Thus the number of sequences beginning with 1 2 3 ... n-2 is given approximately by:

$$(n-1)n^{L-n}\ (1+p+p^2+p^3+\ \dots\ )=(n-1)n^{L-n+1}/2,$$

where $p=(n-2)/n$ this time.

Continuing in this fashion, one can form a sequence of partial stages in the summing of all canonical sequences, corresponding to fixed initial strings of monotonically decreasing length. This sequence of partial sums is:

$$\{(n-1)n^{L-n},\quad (n-1)n^{L-n+1}/2,\quad (n-1)n^{L-n+2}/3!,\quad \dots\quad ,\ (n-1)n^{L-3}/(n-2)!\}.$$

Hence, the number of canonical sequences beginning 1 2 is $(n-1)n^{L-3}/(n-2)!$.

Shifting the first 2 one place to the right produces a new effect, since it lengthens the string of numbers at the end which cannot include a 1. In fact, the number of positions with n choices is reduced by 2, while the number with n-1 choices is increased by one. Following along, the total number of canonical sequences is given by:

$$\frac{(n-1)n^{L-3}}{(n-2)!}\,(1+p+p^2+p^3+\ldots) = \frac{(n-1)n^{L-1}}{(n-2)!\,(n^2-n+1)}$$

as in (4.7), and here $p=(n-1)/n$.

Of course this result is only approximate, but as L tends to infinity, the proportion of sequences which are counted erroneously by this procedure tends to 0. Furthermore, these errors are always in the same direction, caused by assuming that a certain number of new sequences will appear, when in fact some will not. Thus (4.7) is an asymptotic upper bound to $s(n,L)$.

4.3 Synthesis of Long SCOSoV's

Once several short sequences have been found, longer sequences can be formed by putting the short ones together in any of several ways. As with complementary sequences, the concept of orthogonal mates proves very useful in the synthesis of long sequences. Two SCOSoV's A and B are orthogonal if $R_{AB}(k) = 0$ for all k.

Theorem 4.10: Let X be a SCOSoV of length L based on a vector set $S = (s_1, s_2, \ldots s_m)$, where m is even. Form the vectors of S into pairs: $v_1 = (s_1, s_2)$, $v_2 = (s_3, s_4)$, ... $v_{m/2} = (s_{m-1}, s_m)$. Let X′ be a SCOSoV formed by:

1. Reversing X.

2. Interchanging the two vectors in each pair, wherever they occur.

3. Negating the first vector in each pair, wherever it occurs.

Then X′ is orthogonal to X.

Proof: As a consequence of Theorem 4.2, it is clear that X′ is a SCOSoV. The orthogonality can be verified easily:

$$R_{XX'}(k) = x_1 \cdot x'_{1+k} + x_2 \cdot x'_{2+k} + \ldots + x_{L-k} \cdot x'_L$$

for k not less than zero. Suppose that $x_i \cdot x'_j$ is non-zero in this summation. By the conditions of the theorem, x_i and x_{L+1-j} belongs to the same v-set. But then $x_{L+1-j} \cdot x'_{L+1-j}$ must also be a non-zero term in the summation. Finally, condition 3 ensures that the two terms will be of opposite sign. Therefore, since every term can be paired with one of opposite sign, the summation must be 0, and X and X′ must be orthogonal, since the same argument will hold for k<0.

Corollary: If X is a SCOSoV of length L based on the set S = $(s_1, s_2, \ldots s_m)$ where m is odd, then a SCOSoV orthogonal to X can be formed on the basis S′ = $(s_1, s_2, \ldots s_m, t)$, where t is another vector orthogonal to the others.

Proof: Although t does not appear in X, X is based upon S′ as well as S. An application of Theorem 4.10 proves the result.

Corollary: If X is a SCOSoV of length L based on a vector set S which has m vectors, where m is even, then a SCOSoV of length 2L can be constructed on the same basis by concatenating X and X′ formed as in Theorem 4.10.

Proof: If Y = XX′ is the new sequence, then

$$R_{YY}(k) = R_{XX}(k) + R_{X'X'}(k) + R_{XX'}(k-L)$$

Since X and X′ are orthogonal,

$$R_{YY}(k) = 2L\delta_{0k}$$

Therefore Y is a SCOSoV.

Corollary: If X is a SCOSoV of length L based on S = (s_1, s_2, ... s_m), where m is odd, then a SCOSoV Y of length 2L can be constructed on the basis S′ = (s_1, s_2, ... s_m, t) formed by adding another orthogonal vector to S.

Proof: Simply combine the previous corollaries.

Theorem 4.10 is not the only source of orthogonal sequences. Any two sequences will be orthogonal if they are defined on orthogonal bases. They do not even need to be the same length. Of course, concatenating two of these sequences will produce a SCOSoV with a length the sum of the smaller lengths, but also with a basis, the size of which is the sum of the two smaller basis sizes.

As well as doubling the lengths of SCOSoV's, one can also form longer SCOSoV's from shorter through an outer product type of operation.

Theorem 4.11: If $(w_1, w_2, \ldots w_n)$ are a set of mutually orthogonal SCOSoV's, all of length L and defined on the same basis $S = (s_1, s_2, \ldots s_m)$, and if $U = (u_1, u_2, \ldots u_{L'})$ is a SCOSoV of length L', defined on a basis $S' = (s'_1, s'_2, \ldots s'_n)$, then the SCOSoV U_s can be formed by replacing every $\pm S'_i$ in U by $\pm w_i$. The sequence U_s has length LL', and it is defined on the basis S.

Proof: The following identity will be used:

$$R_{U_s U_s}(k) = \sum_{i,j=1}^{L} R_{w_{u_i} w_{u_j}}(k + (i-j)L)$$

If k is not congruent to 0 mod L, the $R_{U_s U_s}$ (k) vanishes. if k = pL, for some integer p, then

$$R_{U_s U_s}(k) = \sum_{i=1}^{L} \sum_{j=1}^{L} R_{w_{u_i} w_{u_j}}((p+i-j)L).$$

Here

$$R_{w_{u_i} w_{u_j}}((p+i-j)L) = L(u_j \cdot u_j)$$

if p = j-i, and 0<i, j<L'+1; 0 otherwise. Therefore,

$$R_{U_s U_s}(pL) = R_{U_s U_s}(0) = LL' \delta_{p0}$$

As an example of the usefulness of this theorem, consider the problem of finding a SCOSoV of length 28. The number 28 factors into 4 x 7. The following four SCOSoV's of length 4 based on the set (a,b,c,d) are mutually orthogonal.

w = Abcd

w = abCD

w = aBcD

w = aBCd

Here the negative terms are denoted by capitals. Furthermore, the following SCOSoV of length 7 is defined on the basis (q,r,s,t).

U = qqQrsrt.

Combining them all together yields

U ¯= abcdabcdABCDabCDaBcDabCDaBCd.

Since 28 is not a sum of 3 integer squares, no SCOSoV produced by any other method could have used a smaller basis. Therefore, this method has constructed a minimal 28-SCOSoV.

Besides the factoring method, two other algorithms will be given for preparing new SCOSoV's from old. Each will work better than the others in some circumstances. Probably a combination of the three methods will work best.

Algorithm 4.2:

Given: A length L.

Required: To find a SCOSoV of length L with a small basis.

Procedure:

1. If L has a known minimal solution, stop.

2. Find the largest SCOSoV with a 2-vector basis which has a length L′ not greater than L.

3. Use the SCOSoV found in (1) to begin the sequence. Concatenate onto it the results of applying the procedure recursively with the length L-L′. To ensure that the correlations of the sub-sequences are zero, define each sub-sequence on a new basis, orthogonal to all previous bases.

For example, suppose L = 699. The closest SCOSoV with a 2-vector basis is 26 x 26 = 676. The remainder is 23. The closest SCOSoV with a 2-vector basis is 20. The remainder is 3, which has a minimal solution of a 3-vector basis. Therefore, the basis for the SCOSoV of length 669 will have 2+2+3=7 vectors. This may or may not be minimal, but it is relatively small.

Algorithm 4.3

Given: A length L.

Required: To find a SCOSoV of length L with a small basis.

Procedure:

1. If L has a known minimal solution, stop.

2. If L is odd, put one vector into the basis and subtract 1 from L. The result, being even, can be split into two halves. If a SCOSoV can be formed for the new halved length, an orthogonal mate can be made for it using Theorem 4.10. The two can then be concatenated to produce a SCOSoV of length L. It is clear then that the maximal number of basis vectors required for a SCOSoV of length L is at most 2 greater than the number required for a SCOSoV of length the integer part of L/2 (since the application of Theorem 4.10, or its corollaries, may demand an extra basis vector as well).

3. Apply the above steps recursively until a length is reached for which the minimal SCOSoV is known. Then work backward, doubling the sequence length, and maybe adding one extra vector at each stage, until a SCOSoV of length L is reached.

For the example 699 used with the last algorithm, this one will not work very well. However, consider 768, which yields a basis of 8 using Algorithm 4.2. Notice that $768 = 2 \times 384 = 2 \times 2 \times 192 = 2 \times 2 \times 2 \times 96 = 2 \times 2 \times 2 \times 2 \times 48 = 2 \times 2 \times 2 \times 2 \times 2 \times 24$, and since $24 = 2 \times 12$, the length of the basis need only be 4. Since 768 is not the sum of three squares, one can even conclude that this basis is minimal.

The list given in Appendix 2 of known minima less than 100 is of course incomplete. Undoubtedly better applications of the known methods and better methods could improve the completeness of the list considerably.

Chapter 5: Orthogonal Systems of Complementary Sequences

5.1 Complementary Code Sets

It has been observed that orthogonal mates facilitate the synthesis of long complementary sequences from shorter ones. In the introduction it was also mentioned that orthogonal pairs of complementary sequences provided a boon in the development of S.A.W. devices, due to the efficient use of crystal surface area by providing multiple occupancy of the two signal paths required for the implementation of complementary sequences. Bernard Schweitzer, in his doctoral dissertation, extended the study to multiple signal paths. He showed that for certain integers m, m signal paths could be simultaneously and independently used to transmit information using m orthogonal sets of m complementary sequences. He called these sets complementary code sets. He also found that complementary code sets were intimately connected with the study of orthogonal matrices, and in particular, of Hadamard matrices. An $n \times n$ matrix H of 1's and −1's with the property that $H'H = nI$, where H' is the transpose of H and I is the $n \times n$ identity matrix, is called a Hadamard matrix[7,22]. With this connection, Schweitzer was able to link the study of complementary code sets with a major field of combinatorics.

Schweitzer was not the first to draw a connection between complementary sequences and Hadamard matrices. A paper by Taki et al [17], which dealt with even-shift orthogonal sequences showed how Hadamard matrices could be created using a pair of orthogonal even-shift sequences. A variation of the same argument will show that two pairs of orthogonal complementary sequences will also generate a Hadamard matrix.

Theorem 5.1: If (A,B) and (C,D) are orthogonal pairs of complementary sequences of length L, then the matrix M is a Hadamard matrix, where M is the partitioned matrix shown below.

$$(5.1) \qquad M = \begin{bmatrix} a_1 & a_2 & \cdots & a_L & b_1 & b_2 & \cdots & b_L \\ a_L & a_1 & \cdots & a_{L-1} & b_L & b_1 & \cdots & b_{L-1} \\ \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\ a_2 & a_3 & \cdots & a_1 & b_2 & b_3 & \cdots & b_1 \\ c_1 & c_2 & \cdots & c_L & d_1 & d_2 & \cdots & d_L \\ c_L & c_1 & \cdots & c_{L-1} & d_L & d_1 & \cdots & d_{L-1} \\ \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\ c_2 & c_3 & \cdots & c_1 & d_2 & d_3 & \cdots & d_1 \end{bmatrix}$$

Proof:

The (i,j) element of the product M'M can be found by taking the dot product of the i'th row of M with the j'th row of M. These dot products are actually cyclic correlations. If i and j are both less than L+1, then

$$
\begin{aligned}
(M'M)_{ij} &= C_{AA}(i-j) + C_{BB}(i-j) \\
&= 2L\delta_{ij},
\end{aligned}
$$

since A and B are cyclic complementary sequences if they are complementary sequences, by Theorem 2.11. Similarly, if i<L+1 and j<L+1,

$$
\begin{aligned}
(M'M)_{i,L+j} &= C_{AB}(i-j) + C_{CD}(i-j) \\
&= R_{AB}(i-j)+R_{BA}(L-i+j)+R_{CD}(i-j)+R_{DC}(L-i+j) \\
&= 0,
\end{aligned}
$$

due to the assumed orthogonality. Corresponding results hold for the rest of the coefficients in the matrix product. Therefore, $M'M = 2LI_{2L}$ where I is a 2L x 2L identity matrix. Consequently M is a Hadamard matrix.

Schweitzer proceeded in the opposite direction, showing that Hadamard matrices could be used to construct complementary code sets. In order to describe an entire code set with one matrix, he introduced the concise notation of the z-transform. In this notation, the symbol z is used to represent a unit delay. Furthermore, z^k represents a delay of k units, whether k is positive or negative. A negative delay is a time advance. Using this notation, the complementary sequences

F = 11-11111--

G = 11-1-1--11

can be represented as a pair of polynomials in z.

$$F(z)=z^9+z^8-z^7+z^6+z^5+z^4+z^3+z^2-z-1,$$

$$G(z)=z^9+z^8-z^7+z^6-z^5+z^4-z^3-z^2+z+1.$$

The terms of F and G have been used here as coefficients in the polynomials F(z) and G(z). The correlation function of two such polynomials is given by the following expression:

(5.2) $R_{AB}(z) = A(z)B(1/z).$

The complementary property for F and G can be given concisely as

(5.3) $R_{FF}(z) + R_{GG}(z) = F(z)F(1/z) + G(z)G(1/z) = 20.$

Expansion of the middle expression of this equation will verify that all the coefficients of the product are autocorrelation functions of the original sequences. In general then, any pair of complementary sequences can be represented as a pair of polynomials with coefficients all 1 or -1.

Using the z-transform notation, Schweitzer represented a complementary code set by a matrix with polynomial entries. All the polynomials in one column would be complementary in the sense that the sum of their autocorrelation functions would be a constant. The columns themselves would be mutually orthogonal, in the following sense. If an m x m matrix H(z) represents a complementary code set in z-transform notation, then

(5.4) $\quad H(z)'H(1/z) = mLI_m$,

where L is the length of the individual sequences. Schweitzer did in fact define a correlation function for entire matrices,

(5.5) $\quad R_{AB}(z) = A'(z)B(1/z)$.

His identification of a complementary code set resided in finding m x m z-transform matrices which obeyed the condition

(5.6) $\quad R_{HH}(z) = mLI_m$.

The fact that the coefficients of the polynomials were required to be 1 or -1 does not figure prominently in this formulation. Consequently, Schweitzer went on to discuss the results possible when generalized complementary sequences of real numbers were defined. Some of the interesting results he obtained for the 0-1 case will be developed here. However, the study of real-valued complementary sequences departs too far from the subject of interest to be outlined in this report.

5.2 Some Results on Complementary Code Sets

Schweitzer first obtained the following dimensionality constraint, which will also apply to any SCOSoV's which happen to be orthogonal to his code sets.

Theorem 5.2: If $H(z)$ is a $J \times k$ matrix which obeys condition (5.4), then k is not greater than J.

Proof: A rank argument can be applied to the equation

$$H(z)' H(1/z) = JLI .$$

The rank of the right-hand side is k. The rank of each factor on the left- hand side is no greater than the minimum of J and k. By a well-known theorem of linear algebra, the rank of a product of two matrices is no greater than the rank of any of the factors. Hence, k is no greater than the minimum of J and k, and hence k is no greater than J.

This theorem asserts that the number of orthogonal codes is no greater than the number of complementary sequences in a code. In the most familiar case, a code with two sequences can belong to a set of only two mutually orthogonal mates. Schweitzer termed any complementary code set in which $J=k$ complete. He proceeded to prove the following theorems about complete complementary code sets.

Theorem 5.3: If $H(z)$ is a $J \times k$ complementary code set with sequence length L, then J is even.

Proof: For the i'th column $H(z)$ of $H(z)$,

$$R_{H^iH^i}(z) = JL.$$

Consider the coefficient of z^{L-1} in this expression. If J were odd, this term would be congruent to 1 mod 2, and would not therefore vanish. Hence J must be even.

Theorem 5.4: If H(z) is a J x J complementary code set with sequence length L, in which J>2, then JL = 4t for some integer t.

Proof: Denote the (r,s) element of H(z) by $H\ (z)$. Then

$$JL = H_{11}(z)H_{11}(1/z) + H_{21}(z)H_{21}(1/z) + \dots + H_{J1}(z)H_{J1}(1/z)$$

$$= [H_{11}(z)+H_{12}(z)][H_{11}(z^{-1})+H_{13}(z^{-1})] + [H_{21}(z)+H_{22}(z)][H_{21}(z^{-1})$$

$$+H_{23}(z^{-1})] + \dots + [H_{J1}(z)+H_{J2}(z)][H_{J1}(z^{-1})+H_{J3}(z^{-1})],$$

due to the orthogonality of the columns of H(z). This last form is a sum of terms, each one of which is congruent to 0 mod 4, since each factor is congruent to 0 mod 2. Therefore, JL = 4t, for some integer t.

5.3 Synthesis of Complementary Code Sets

The real usefulness of Schweitzer's thesis lies in the methods he gives for generating complementary codes sets from smaller ones. The first set in any such chain of complementary code sets is called an initializing set. The simplest initializing sets are Hadamard matrices. An m x m Hadamard matrix satisfies all the requirements necessary to make it a complementary code set with L=1. From this starting point, the following generating techniques can be used. The proofs of the following theorems are given in Schweitzer's thesis.

Theorem 5.5: If $H(z)$ is a $J \times J$ complementary code set, with sequence length L, then $AD_L(z)H(z)$ is a $J \times J$ complementary code set, with sequence length JL. Here A represents any $J \times J$ Hadamard matrix, and $D_L(z)$ is a diagonal matrix with diagonal entries $(1, z^L, z^{2L}, \ldots z^{(J-1)L})$.

Theorem 5.6: If $H(z)$ is a $J \times J$ complementary code set, with sequence length L, then $AD_1(z)H(z^J)$ is a $J \times J$ complementary code set, with sequence length JL. Here A and $D_1(z)$ are defined as before.

The following theorems require two complementary code sets.

Theorem 5.7: If $H(z)$ and $G(z)$ are complementary code sets, $J \times J$ and $k \times k$ respectively, with sequence lengths L and M, then $H(z^M)$ (Kr) $G(z)$ is a $Jk \times Jk$ complementary code set with sequence length LM. The operator (Kr) gives the Kronecker product between two matrices, formed by replacing the (r,s) element of the first by that same element multiplied by the second matrix, for all r, s.

Theorem 5.8: If $H(z)$ and $G(z)$ are $J \times J$ complementary code sets with sequence lengths L and M respectively, then $H(z^M)D_{LM}(z)G(z)$ is a $J \times J$ complementary code set with sequence length JLM.

Theorem 5.9: If $H(z)$ and $G(z)$ are $J \times J$ complementary code sets with sequence lengths L and M, then $H(z^{LM})D_M(z)G(z)$ is a $J \times J$ complementary code set with sequence length JLM.

Undoubtedly these theorems would be useful to anyone wishing to apply code sets, for many different sets can be derived from a few basic initializing sets. Schweitzer notes that Hadamard matrices, which must, by a theorem of Payley [11], be of orders which are multiples of 4, are known for all orders under 200, except for 188 [22]. Unfortunately, Schweitzer's synthesis methods are not so useful for finding SCOSoV's.

Systems of orthogonal SCOSoV's could also be useful for generating other SCOSoV's. The theorems given by Schweitzer, however, using a Hadamard matrix as an initializing set, will yield at best sequences of length 4 , for a basis set of size 4.

Chapter 6: Conclusions

6.1 Complementary Sequences

Complementary sequences have been applied in more than one field, but especially in the field of S.A.W. devices. To satisfy the limited requirements of the physical applications demands little more than the information supplied by Marcel Golay in a 1961 paper [5]. Knowledge of orthogonal mates to complementary series has been found useful as well, however, as suggested by C. C. Tseng in a 1971 paper [18].

Investigation of complementary series with the physical uses in mind has lead to more theoretical studies designed to identify characteristics of the complementary property. None of these studies has proved singularly successful, for little more is known today about Golay's sequences than was known to Golay in 1961. Instead, these studies have lead toward generalizations of complementary sequences. C. C. Tseng and C. L. Liu investigated the properties of larger groups of complementary sequences [19]. B. P. Schweitzer produced some interesting results on orthogonal sets of complementary sequences [13]. All of these investigations have produced a few properties which seem to link together in a way suggesting that the complementary condition itself requires certain forms regardless of the mathematical structures to which it is applied.

6.2 SCOSoV's

In an attempt to study qualities of the complementary property divorced from the sequences to which it is applied, I have devised the structures which I call complementary sequences of orthogonal vectors, or SCOSoV's. Once again, the rules which govern the occurrence and behaviour of these sequences have proved elusive. However, working on the principle that information about specific sequences will reveal some of the nature of the abstract SCOSoV, I have attempted to find sample sequences to study. In the process, I have devised algorithms to find solutions to simultaneous binary systems of multinomial equations. These algorithms apply equally well to SCOSoV's and to complementary series as they were originally defined.

The results of my efforts, which have been primarily directed towards complementary series, have been largely negative. Although I have found the complete systems of solutions for L=2,4,8,10,16,20,26, and 32, I have found no new systems of solutions, though the sequence lengths L=34,36, and 50 have been disqualified in the search. The situation with regard to SCOSoV's is much more open, in that I have used mainly pencil-and-paper techniques to find solutions, although I have provided in Chapter 4 an algorithm for automating the search.

6.3 Problems Which Remain

Many problems remain unsolved in the study of complementary sequences. On the numerical side, 58 is the shortest length which may support an unknown pair of complementary sequences. Initial attempts to test for these sequences have been discouraging, but so far fewer than 5% of the possibilities have been tested.

Now that 34, 36, and 50 have failed to produce complementary sequences, room has been provided to conjecture that perhaps no more complementary sequences exist. This conjecture can obviously not be proven through exhaustive searches, so more powerful methods for analyzing sequences should perhaps be sought. Arguments on the line of Kruskal's [10] may in the end prove more useful than those heretofore applied.

Alternatively, techniques may be found for removing the extensive testing at present required to solve binary systems of multinomial equations. The methods which I have devised may have application in other combinatorial problems, on the other hand. Certainly a sound algorithm for solving such systems could prove very useful.

Further research should be expended on SCOSoV's as well. I conjecture that no SCOSoV based on a vector set with an odd number of orthogonal vectors will have an orthogonal mate on the same basis. I further predict that no basis is large enough to support SCOSoV's of any arbitrarily great length. In any case, SCOSoV's offer the opportunity, not available with complementary sequences, to gather enough specimens to build up properties by an empirical process.

All things considered, the examination of complementary structures of all sorts has only reached a tantalizing stage where one feels that the introduction of new ideas and more powerful techniques could produce many large advances.

Appendix 1: A Computer Implementation of Algorithm 3.1

Algorithm 3.1 has been successfully implemented in the computer language C[1] on a PDP 11/45 computer at the University of Manitoba. The program which follows, operating interactively under the UNIX time-sharing system[2], inputs L, the length of the complementary sequences being sought. It proceeds to find all complementary sequences of length L, and then writes them in quad notation onto a file with the name sol[L], where [L] is replaced by the numeric value of L. The program is restartable from checkpoints which are printed periodically on the terminal. These checkpoints indicate the high order bits of the sequence of LROB's currently being tested.

All sequences of LROB's are tested, although without loss of generality it is assumed that the first bit is always 1. To reduce run time, it is desirable to minimize the number of changes which must be made from one LROB sequence to the next, and from one UDOB or SB sequence to the next when partial solutions are found. This is accomplished by using a sequence of Gray codes to produce all possible sequences of n bits, rather than the simple counting order.

[1] Dennis M. Ritchie, "C REFERENCE MANUAL", Internal memorandum, Bell Laboratories, 1974.

[2] D. M. Ritchie and K. L. Thompson, "THE UNIX TIME-SHARING SYSTEM", CACM Vol. 17, No. 7, July, 1974, pp. 365-375.

An algorithm of Bitner et al[3] is used to generate the Gray codes. They are characterized by the property that all binary sequences of a fixed length can be generated in an order which requires the change of only one bit from one code to the next.

The successful complementary sequences written out in quad form by this program are not all unique; some are isomorphic. In fact, exactly two isomorphic copies of each distinct sequence will be found. They can easily be identified in the output, since the duplicate is formed by replacing each R(r) by S(s), and vice versa. These isomorphisms are tolerated because they can be identified so easily in the output, but not so easily in the midst of the program. The sequences listed in Appendix 2 have been vetted so that no isomorphic pairs appear.

Another program, implementing only a part of Algorithm 3.1, has also been implemented. It uses a more efficient, but less comprehensible, representation of the sequences, and hence it runs faster, although it produces only candidates, and not complete complementary sequences. It was used to verify the non-existence of complementary sequences of length 50, a task which would be beyond the program given here.

[3] J. R. Bitner, G. Ehrlich, E. Reingold, "EFFICIENT GENERATION OF THE BINARY REFLECTED GRAY CODE AND ITS APPLICATIONS", CACM Vol. 19, No. 9, September, 1976, pp. 517-521.

```
/*  EXTERNAL DECLARATIONS        */

char buffer[80],             /* For writing solutions        */
     CHECK,                  /* Index for noting checkpoints */
     dim2 {40};              /* Max value for Ndiv2          */

int  fildes;                 /* Descriptor of answer file    */

char filnam[7] {"sol    "},        /* Name of answer file   */
     flag,                   /* True-false indicator         */
     si2,                    /* Index in new2()              */
     si3,                    /* Index in new3()              */
     Gray1[80],              /* Stack for producing Gray codes */
     Gray2[80],              /* Stack for producing Gray codes */
     Gray3[80],              /* Stack for producing Gray codes */
     lim,                    /* For-loop limit               */
     LROB[80],               /*  Left-Right Orientation Bit  */
     matrix[80][40],         /* Augmented matrix of change   */
     N,                      /* length of sequences          */
     Ndiv2,                  /* Half of sequence length N    */
     Ndiv2p,                 /* One plus Ndiv2               */
     Nmin1,                  /* N minus 1                    */
     Nmin2,                  /* N minus 2                    */
     NOS,                    /* End index of SIGNS           */
     p,                      /* General index                */
     quad[16] {"srtq            QTRS"}, /* Output table      */
    *row1,                   /* Row pointer                  */
     rslt1,                  /* Dimension of solut1          */
     rslt2,                  /* Dimension of solut2          */
     SEQ[80],                /* Sequence of LROB's & UDOB's  */
     SEQ2[80],               /* Sequence of quads            */
     SIGNS[10],              /* Total nos of quads with one  sign
*/
     solut1[20][40],         /* Solution vector and kernel  basis
*/
     solut2[20][40],         /* Solution vector and kernel  basis
*/
     sq[10] {0,1,4,9,16,25,36,49,64,81}, /* Squares      */
    *srow1,                   /*  pointer  to solution matrix row
*/
    *srow2;                   /*  pointer  to solution matrix row
*/

int  sum,                    /* General purpose sum          */
     tmp;                    /* General purpose variable     */
```

```
main(argc,argv)

int argc; char **argv;

{


    for( N = q = 0 ; argv[1][q] ; N=10*N+argv[1][q++]-'0');
    printf("#Test  for  complementary  sequences  of  length
N=%4d#",N);

    /*                                           */
    /*  Initialize some frequently used values.  */
    /*                                           */


    Ndiv2 = N/2;
    Ndiv2p = Ndiv2 + 1;
    Nmin1 = N-1;
    Nmin2 = N-2;
    if(!testN()) {

        printf("This   value   of   N   does   not   support
complementary sequences.#");

    }

    else {

        if(argc>2) {

            for(CHECK=q=0 ; argv[2][q] ; )
                CHECK = 10*CHECK+argv[2][q++]-'0';

        }
        else CHECK = 16;
        printf("#Checkpoints  will  be printed whenever LROB
number %2d is changed.#",CHECK);

        /*                                          */
        /*  Create name for file in which all CS    */
        /*  will be stored when found.              */
        /*                                          */

        for(q=0 ; argv[1][q] ; filnam[3+q++] = argv[1][q] );
        filnam[q+3] = '0';
        fildes = creat(filnam,384);
        printf("#All complementary sequences will be  stored
in file %5s#",filnam);

        /*                                          */
        /*  Initialize Gray1 (for Gray codes) and LROB  */
        /*                                          */
```

```
    if(argc > 3) {

        for(q=p=0;argv[3][p];q=10*q+argv[3][p]-'0',p++);

        /*                                          */
        /*  q now describes the arrangement         */
        /*  of 0's and 1's in LROB to start.        */
        /*                                          */

        init1(q) ;
    }
    else init1(0);

    /*                                              */
    /*  Now test all possible LROB sequences which  */
    /*  begin with a 0.                             */
    /*                                              */

    do {

        set1();
        if(rslt1 = solve( solut1 )) {

            init2() ;
            do {

                set2();
                if(rslt2 = solve(solut2)) {

                    init3();
                    do {

                        if(test()) {

                            for( q=0 ; q<Ndiv2 ; q++ )
                                buffer[q]=quad[7+SEQ2[q]];
                            buffer[q] = '#';
                            write(fildes,buffer,Ndiv2p);

                        }

                    } while (new3());

                }

            } while (new2());

        }

    } while (new1());

}

}
```

```
int init1(map)

int map; { int register q,r;

    /*                                                              */
    /*      This    function    initializes    LROB    and   Gray1.
*/
    /*                                                              */

    flag = 0;
    r = 1;
    for(q =Ndiv2 ; q-- ; ) {

        LROB[q] = tmp = map%2;
        map =/ 2;
        LROB[Nmin1-q] = !tmp;
        Gray1[q+1] = q+2;
        if(flag) {

            if(r != tmp)
                Gray1[q+1] = p,
                flag = 0;

        } else {

             if ( r == tmp )
                flag = 1,
                p = q+2;

        }
        r =^ tmp;

    }
    Gray1[0] = 1;

}
```

```
int init2() {

    /*                                                      */
    /*  Initialize SEQ and Gray2                            */
    /*                                                      */

char register a,*row;

    row = solut1[0];
    for(a = 0 ; a<Ndiv2 ; )
        SEQ[a] = tmp = LROB[a]+2*row[a],
        SEQ[Nmin1-a++] = tmp^1;
    for(a=0; a<=rslt1; Gray2[a++] = a+1 );

}



int init3() {

    /*                                                      */
    /*  Initialize SEQ2 and Gray3                           */
    /*                                                      */

char register a,*row;

    row = solut2[0];
    for(a = 0 ; a<Ndiv2 ; )
        SEQ2[a] = tmp = (SEQ[a]^4)*(row[a]? -1 : 1),
        SEQ2[Nmin1-a++] = (tmp>0 ? tmp^1 : -(1^-tmp) );
    for(a=0; a<=rslt2; Gray3[a++] = a+1 );

}
```

```
int new1() {

    /*                                                    */
    /*  Generate a new LROB sequence, using Gray codes so */
    /*  that only one quad need be changed each time.     */
    /*                                                    */

int register si1,q;

    si1 = Gray1[0];
    Gray1[0] = 1;
    Gray1[si1-1] = Gray1[si1];
    Gray1[si1] = si1+1;
    LROB[si1] =^ 1;
    LROB[Nmin1-si1] =^ 1;

    if(si1 >= CHECK) {

        for(sum=0, q=CHECK ; q <Ndiv2 ; )
            sum = LROB[q++]+2*sum;
        printf("Checkpoint: sum %10d\n",sum);

    }
    return(si1 < Ndiv2); }



int new2() {

    /*                                                          */
    /*  Generate a new sequence of LROB's and UDOB's,  using
*/
    /*  Gray codes, so that only one quad need have its UDOB
*/
    /*              changed              each              time.
*/
    /*                                                         */

int register q; char register *row;

    si2 = Gray2[0];
    Gray2[0] = 1;
    Gray2[si2-1] = Gray2[si2];
    Gray2[si2] = si2+1;
    row = solut1[si2];

    for(q=0; q<Ndiv2; q++ ) {

        SEQ[q] =^ 2*row[q];
        SEQ[Nmin1-q] =^ 2*row[q];

    }
    return(si2 < rslt1); }
```

```
int new3() {

    /*                                              */
    /*  Generate a new potential complementary      */
    /*  sequence to be tested by test().            */
    /*                                          */

int register q; char register *row;

    si3 = Gray3[0];
    Gray3[0] = 1;
    Gray3[si3-1] = Gray3[si3];
    Gray3[si3] = si3+1;
    row = solut2[si3];
    for( q = 0 ; q<Ndiv2 ; )
        SEQ2[q] =* (tmp = (row[q]?-1:1)),
        SEQ2[Nmin1-q] =* tmp,
        q++;
    return(si3 < rslt2); }
```

```
int set1() {

    /*                                                      */
    /*  Set up the augmented matrix of a  system  of  binary
*/
    /*   equations  which  the  appropriate  UDOB   sequence
*/
    /*                      must                     satisfy.
*/
    /*                                                      */

char register q,*row;

    row = matrix[0];

    for(p=0; p<Nmin2 ; ){

        lim = ( p++ < Ndiv2 ? Ndiv2 : N-p );

        for(q=0; q<lim ; row[q++] = 0);

        row[ Ndiv2p ] = 0;

        for(flag=sum=q=0; q<lim ; q++ ) {

            if(LROB[q] == LROB[q+p]) {

                if(1 == (row[q] =^ 1)) flag =1;

                if(p+q < Ndiv2 ) {

                    row[p+q] =^ 1;
                    sum =+ 2;

                }
                else sum++;
            }
        }
        if(flag) {

            row[ Ndiv2 ] = (sum/2)%2;
            row[ Ndiv2p ] = lim;
            row =+ dim2;

        }

    }

    row[0] = 1,
    row[ Ndiv2 ] = 0,
    row[ Ndiv2p ] = 1;
    (row+dim2)[Ndiv2p] = 0;

}
```

```
int set2() {

    /*                                                          */
    /*  Set up the augmented matrix of a  system  of  binary
*/
    /*     equations  which  the  appropriate  SB   sequence
*/
    /*                         must                    satisfy.
*/
    /*                                                         */

char register q, *row;

    row = matrix[1];

    for(p=0; p<Nmin2 ; ){

        lim = ( p++ < Ndiv2 ? Ndiv2 : N-p );

        for(q=0; q<lim ; row[q++] = 0);

        for(flag=sum=q=0; q<lim ; q++ ) {

            if(SEQ[q] == SEQ[q+p]) {

                if(1 == (row[q] =^ 1)) flag =1;

                if(p+q < Ndiv2 ) {

                    row[p+q] =^ 1;
                    sum =+ 2;

                }
                else sum++;
            }
        }

        if(flag) {

            row[ Ndiv2 ] = (sum/4)%2;
            row[ Ndiv2p ] = lim;
            row =+ dim2;

        }

    }

    row[0] = 1;
    row[ Ndiv2 ] = 0;
    row[ Ndiv2p ] = 1;
    (row+dim2)[Ndiv2p] = 0;
    row = matrix[0];

    for( q=0 ; q<Ndiv2 ; q++ )
```

```
        if( (2 & SEQ[a]) != (2 & SEQ[0]) ) {
            row[a] = 1;
            break;

        }
        else row[a] = 0;

    for( ; a++<Ndiv2 ; row[a] = 0);

    row[ Ndiv2p ] = Ndiv2;

}
```

```
int solve(sol)

char sol[20][40];

{ char register q,*row;

     row1 = matrix[0];

     for( p = Ndiv2; p-->0 ; ) {

          /*  the columns are reduced from right to left,   */
          /*  top to bottom.                                */

          for(row=row1; p<row[Ndiv2p] ; row =+ dim2) {

               if(row[p]) {

                    if(row != row1) {

                         for( q =0 ; q<p ; )

                              tmp = row[q],
                              row[q] = row1[q],
                              row1[q++] = tmp;

                         tmp = row[Ndiv2];
                         row[Ndiv2] = row1[Ndiv2];
                         row1[Ndiv2] = tmp;

                    }

                    for( row  =+  dim2;  p<row[Ndiv2p];  row  =+
dim2) {

                         if(row[p]) {

                              for(  q=  0  ; q<p ; q++ ) row[q] =^
row1[q];

                              row[Ndiv2] =^ row1[Ndiv2];

                         }

                    }
                    for(row = matrix[0] ; row < row1  ;  row  =+
dim2) {

                         if(row[p]) {


                              for(q=0   ;   q<p  ;  q++)row[q]  =^
row1[q];

                              row[Ndiv2] =^ row1[Ndiv2];

                         }
```

```
                }
                row1[Ndiv2p] = p;
                row1 =+ dim2;
                break;

            }
        }

    }

    /*
*/
    /*  Check  that  the  system  of  equations  is  not
inconsistent  */
    /*
*/

    for(row = row1 ; 0 < row[Ndiv2p] ; row =+ dim2 )
        if(row[Ndiv2]) return (0);

        /*
*/
        /*  Now that the reduction is complete, the solution
*/
        /*       can       be       formed       in       sol.
*/
        /*
*/


        row1 = matrix[0];
        srow1 = srow2 = sol[0];

        for(p = Ndiv2 ; p-->0 ;  ){

            srow1[p] = 0;

            if(row1[Ndiv2p] == p) {

                srow1[p] = row1[Ndiv2];
                row1 =+ dim2;

            }
            else {

                srow2 =+ dim2 ;
                q = Ndiv2;

                for(row   =   matrix[0]   ;   row<row1   ;  row =+
dim2) {

                    for(; --q>row[Ndiv2p] ; srow2[q] = 0);
                    srow2[q] = row[p];
```

```
                }

                for( ; q-->0 ; srow2[q] = 0);
                srow2[p] = 1;

            }

        }

        return(1+(srow2-srow1)/dim2);

}
```

```
int test() {

int register z;

    /*
*/
    /*   First  test  for  an  acceptable  number   of   the
appropriate */
    /*                                                   SB's.
*/
    /*
*/

    for( z = sum = 0 ; z<Ndiv2 ; z++ ) sum =+ (0<SEQ2[z]) ;
    sum =* 2;
    for(SIGNS[NOS] = sum,z = 0 ; SIGNS[z] != sum ; z++ );

    if(z == NOS) return (0);

    /*
*/
    /*  Now do a thorough straightforward check on  all  the
*/
    /*                                        auto-correlations.
*/
    /*
*/

    for(p = Nmin2; p ; p-- ) {

        for(sum = 0, z = N ; p<z-- ; )

            sum =+ (SEQ2[z]  ==  SEQ2[z-p])  -  (SEQ2[z]  ==
-SEQ2[z-p]);

        if(sum != 0) return (0);

    }

    return(1);

}
```

```
int testN() {

    /*                                                        */
    /*  Test the sequence length N to make sure that          */
    /*  it is twice the sum of 2 squares.                     */
    /*                                                        */

int register q,r;

    NOS = 0;
    for( p = 0 ; sq[p] <= Ndiv2 ; p++  ) {

        tmp = Ndiv2-sq[p];

        for( q = p ; sq[q] < tmp; q++);

        if(sq[q] == tmp) {

            tmp = SIGNS[NOS] = p-q+Ndiv2;

            for(r =0 ; tmp != SIGNS[r++] ; ) ;


            if(--r == NOS) {

                NOS++;
                if(q)SIGNS[NOS++] = p+q+Ndiv2;
                if (p)SIGNS[NOS++] = Ndiv2-p-q;
                if(p!=q) {

                SIGNS[NOS++] = Ndiv2+q-p;

                                }
                else return(1);

        }
        else return(1);


        }

    }

    return(NOS);

}
```

Appendix 2:

A Complete List of All Complementary Sequences for L<40

Complete lists of all complementary sequences have been found for L = 2, 4, 8, 10, 16, 20, 26, and 32. A partial list is available for L = 40. These sequences are given here in quad notation. The list has been trimmed by removing isomorphic matches to the sequences included. These isomorphic forms can be recovered by using various transformations described in the text of this thesis.

L = 2

Q

L = 4

QR

L = 8

QQRr QqRR QRQr QRqR QRSt QRsT

L = 10

QRsTR QRsTr QQqRS QqqRS

L = 16

QQQqRrRR QQQqRRrR QQqQRrrr QQqQRRRr QqqqRrRR
QqqqRRrR QqQQRrrr QqQQRRRr QQRRQqrR QQRRqQRr
QQRrqQrr QQRrQqRR QqRrQQrr QqRrqqRR QqRRqqrR
QqRRQQRr QRQRQrqR QRQRqRQr QRQrqrQr QRQrQRqR
QRqrQrQr QRqrqRqR QRqRqrqR QRqRQRQr QRStQRsT
QRStQrST QRStqrSt QRStqRst QRsTqRST QRsTqrsT
QRsTQrst QRsTQRSt QQRRSstT QQRRsSTt QQRrssTt
QQRrSStT QqRrSStt QqRrssTT QqRRsSTT QqRRSstt
QRQRStsT QRQRsTSt QRQrsTst QRQrStST QRqrStSt
QRqrsTsT QRqRsTST QRqRStst

L = 20

QQqqRqRrRR QQqQRQRRrr QqqQRQRRRr QqqqRqRrrR QQqRqRqrRR
QQqRQRQRrr QqqRQRQRRr QqqRqRqrrR QQRRQrQqrR QQRRqRQqrR
QqRrQRQQrr QqRrqrQQrr QRQRRqQrqR QRQRrQQrqR QRqrRQQrQr
QRqrrqQrQr QRQRqrsTQr QRQRqrStqR QRqrqrStqR QRqrqrsTQr
QRQrStSTQr QRQrStSTqR QRqRsTSTQr QRqRsTSTqR QRRQqQSttQ
QRrqqqStTq QRQRqrqRSt QRQRqrQrsT QRqrqrQrsT QRqrqrqRSt
QRStQrSTSt QRStQrSTsT QRsTqRSTSt QRsTqRSTsT

L = 26

QQQRqQQrqQqRS QqQRqqQrqqqRS

L = 32

QQQQQqqQRRrrRrRr	QQQQQqqQRrRrRRrr	QQQQqQQqRRrrrRrR
QQQQqQQqRrRrrrRR	QQQqqQqqRRRrRrRR	QQQqqQqqRRrRrRRR
QQQqqqQqRRRrRRrR	QQQqqqQqRrRRrRRR	QQQqQqQQRRRrrRrr
QQQqQqQQRRrRRrrr	QQQqQQqQRRRrrrRr	QQQqQQqQRrRRRrrr
QQqqQqQqRRRRRrrR	QQqqQqQqRrrRRRRR	QQqqqQqQRRRRrRRr
QQqqqQqQRrrRrrrr	QQqQqQQQRRRrrRrr	QQqQqQQQRRrRRrrr
QQqQqqqQRRrRRRRr	QQqQqqqQRrrrrRrr	QQqQQqqqRRRrRrRR
QQqQQqqqRRrRrRRR	QQqQQQQqRRrRrrrR	QQqQQQQqRrrrRrRR
QqqQQQQQRRrrRrRr	QqqQQQQQRrRrRRrr	QqqQqqqqRRrrrRrR
QqqQqqqqRrRrrrRR	QqqqqQqqRRrRrrrR	QqqqqQqqRrrrRrRR
QqqqqqQqRrrrRRrR	QqqqqqQqRrRRrrrR	QqqqQqQQRRrRRRRr
QqqqQqQQRrrrrRrr	QqqqQQqQRrrrrrRr	QqqqQQqQRrRRRRRr
QqQqQQqqRRRRRrrR	QqQqQQqqRrrRRRRR	QqQqqqQQRRRRrRRr
QqQqqqQQRrrRrrrr	QqQQqQQQRRRrrrRr	QqQQqQQQRrRRRrrr
QqQQqqqQRrrrrrRr	QqQQqqqQRrRRRRRr	QqQQqqqqRRRrRRrR
QqQQQqqqRrRRrRRR	QqQQQQQqRrrrRRrR	QqQQQQQqRrRRrrrR
QQQQRRrrQqqQRrRr	QQQQRRrrqQQqrRrR	QQQQRrRrQqqQRRrr
QQQQRrRrqQQqrrRR	QQQqRrRRQQqQRrrr	QQQqRrRRqqQqrRRR
QQQqRRrRQqQQRrrr	QQQqRRrRqQqqrRRR	QQQqRRRrQQqQrrRr
QQQqRRRrQqQQrRrr	QQQqRRRrqqQqRRrR	QQQqRRRrqQqqRrRR
QQqqRRRRQqQqRrrR	QQqqRRRRqQqQrRRr	QQqqRrrRQqQqRRRR

QQqqRrrRqQqQrrrr	QQqQRrrrQQQqRrRR	QQqQRrrrqqqQrRrr
QQqQRRrRQQQqrrrR	QQqQRRrRQqqqrRRR	QQqQRRrRqqqQRRRr
QQqQRRrRqQQQRrrr	QQqQRRRrQqqqRrRR	QQqQRRRrqQQQrRrr
QqqQRRrrQQQQRrRr	QqqQRRrrqqqqrRrR	QqqQRrRrQQQQRRrr
QqqQRrRrqqqqrrRR	QqqqRrRRQQqQRRRr	QqqqRrRRqqQqrrrR
QqqqRrrrQQqQrrRr	QqqqRrrrQqQQrRrr	QqqqRrrrqqQqRRrR
QqqqRrrrqQqqRrRR	QqqqRRrRQqQQRRRr	QqqqRRrRqQqqrrrR
QqQqRRRRQQqqRrrR	QqQqRRRRqqQQrRRr	QqQqRrrRQQqqRRRR
QqQqRrrRqqQQrrrr	QqQQRrRRQQQqrrrR	QqQQRrRRQqqqrRRR
QqQQRrRRqqqQRRRr	QqQQRrRRqQQQRrrr	QqQQRrrrQQQqRRrR
QqQQRrrrqqqQrrRr	QqQQRRRrQqqqRRrR	QqQQRRRrqQQQrrRr
QQRRQQrrQqRrqQRr	QQRRQQrrqQrRQqrR	QQRRQqrRQqRrQQrr
QQRRQqrRqQrRqqRR	QQRRQqRrQQrrqQRr	QQRRQqRrQqrRQQrr
QQRRQqRrqqRRQqrR	QQRRQqRrqQRrqqRR	QQRRqqRRQqRrQqrR
QQRRqqRRqQrRqQRr	QQRRqQrRQQrrQqrR	QQRRqQrRQqrRqqRR
QQRRqQrRqqRRqQRr	QQRRqQrRqQRrQQrr	QQRRqQRrQqRrqqRR
QQRRqQRrqQrRQQrr	QQRrqQrrQQRrQqRR	QQRrqQrrqqrRqQrr
QQRrqqrRQqRRQqRR	QQRrqqrRqQrrqQrr	QQRrQqRRQQRrqQrr
QQRrQqRRqqrRQqRR	QQRrQQRrQqRRqQrr	QQRrQQRrqQrrQqRR
QqRrQQRRQQrrQqrR	QqRrQQRRQqrRqqRR	QqRrQQRRqqRRqQRr
QqRrQQRRqQRrQQrr	QqRrQQrrQQRRQqrR	QqRrQQrrqqrrqQRr
QqRrQqrRQQRRqqRR	QqRrQqrRqqrrQQrr	QqRrqqRRQQRRqQRr
QqRrqqRRqqrrQqrR	QqRrqqrrQQrrqQRr	QqRrqqrrQqrRQQrr
QqRrqqrrqqRRQqrR	QqRrqqrrqQRrqqRR	QqRrqQRrQQRRQQrr
QqRrqQRrqqrrqqRR	QqRRqQrrQQRrQQRr	QqRRqQrrqqrRqqrR
QqRRqqrRQqRRQQRr	QqRRqqrRqQrrqqrR	QqRRQqRRQQRrqqrR
QqRRQqRRqqrRQQRr.	QqRRQQRrQqRRqqrR	QqRRQQRrqQrrQQRr

QRQRQRqrQrQrqRQr	QRQRQRqrQrqRQrQr	QRQRQRqrqRqRQrqR
QRQRQRqrqRQrqRqR	QRQRQrqRQRqrQrQr	QRQRQrqRqrQRqRqR
QRQRQrQrQRqrqRQr	QRQRQrQrqrQRQrqR	QRQRqrQRQrQrQrqR
QRQRqrQRQrqRqRqR	QRQRqrQRqRqRqRQr	QRQRqrQRqRQrQrQr
QRQRqRqRQRqrQrqR	QRQRqRqRqrQRqRQr	QRQRqRQrQRqrqRqR
QRQRqRQrqrQRQrQr	QRQrqrqRQRqRQRqR	QRQrqrqRqrQrqrQr
QRQrqrQrQRQrQRqR	QRQrqrQrqrqRqrQr	QRQrQRqRQRQrqrQr
QRQrQRqRqrqRQRqR	QRQrQRQrQRqRqrQr	QRQrQRQrqrQrQRqR
QRqrQRQRQrQrQrqR	QRqrQRQRQrqRqRqR	QRqrQRQRqRqRqRQr
QRqrQRQRqRQrQrQr	QRqrQrqRQRQRqRqR	QRqrQrqRqrqrQrQr
QRqrQrQrQRQRQrqR	QRqrQrQrqrqrqRQr	QRqrqrqrQrQrqRQr
QRqrqrqrQrqRQrQr	QRqrqrqrqRqRQrqR	QRqrqrqrqRQrqRqR
QRqrqRqRQRQRqRQr	QRqrqRqRqrqrQrqR	QRqrqRQrQRQRQrQr
QRqrqRQrqrqrqRqR	QRqRqrqRQRqRQRQr	QRqRqrqRqrQrqrqR
QRqRqrQrQRQrQRQr	QRqRqrQrqrqRqrqR	QRqRQRqRQRQrqrqR
QRqRQRqRqrqRQRQr	QRqRQRQrQRqRqrqR	QRqRQRQrqrQrQRQr
QRStQRStQRsTqrSt	QRStQRStQrSTqRst	QRStQRStqrStQRsT
QRStQRStqRstQrST	QRStQRsTqRstqRST	QRStQRsTqrsTQRsT
QRStQRsTQrSTQrst	QRStQRsTQRStqrSt	QRStQrSTqrsTQrST
QRStQrSTqrStqRST	QRStQrSTQRsTQrst	QRStQrSTQRStqRst
QRStqrStQRStQRsT	QRStqrStQrSTqRST	QRStqrStqrsTqrSt
QRStqrStqRstQrst	QRStqrsTqRstqRst	QRStqrsTqrStqrSt
QRStqrsTQrSTQrST	QRStqrsTQRsTQRsT	QRStqRstQRStQrST
QRStqRstQRsTqRST	QRStqRstqrStQrst	QRStqRstqrsTqRst
QRsTqRSTQRStqRst	QRsTqRSTQRsTQrst	QRsTqRSTqrStqRST
QRsTqRSTqrsTQrST	QRsTqrsTQRsTQRSt	QRsTqrsTQrstqRst
QRsTqrsTqrStqrsT	QRsTqrsTqRSTQrST	QRsTqrStqRSTqRST

QRsTqrStqrsTqrsT	QRsTqrStQrstQrst	QRsTqrStQRStQRSt
QRsTQrstqrsTqRst	QRsTQrstqrStQrst	QRsTQrstQRsTqRST
QRsTQrstQRStQrST	QRsTQRsTQRStqrsT	QRsTQRsTQrstqRST
QRsTQRsTqrsTQRSt	QRsTQRsTqRSTQrst	QRsTQRStqRSTqRst
QRsTQRStqrStQRSt	QRsTQRStQrstQrST	QRsTQRStQRsTqrsT
QQRRSSttQqrRSsTt	QQRRSSttQqRrsSTt	QQRRSSttqQrRSstT
QQRRSSttqQRrsStT	QQRRSstTqQrRssTT	QQRRSstTqqRRsStT
QQRRSstTQqRrSStt	QQRRSstTQQrrSsTt	QQRRssTTQqrRsStT
QQRRssTTQqRrSstT	QQRRssTTqQrRsSTt	QQRRssTTqQRrSsTt
QQRRsSTtqQrRSStt	QQRRsSTtqqRRSsTt	QQRRsSTtQqRrssTT
QQRRsSTtQQrrsStT	QQRrssTtqQrrSstt	QQRrssTtqqrRssTt
QQRrssTtQqRRsSTT	QQRrssTtQQRrSStT	QQRrSStTqQrrsSTT
QQRrSStTqqrRSStT	QQRrSStTQqRRSstt	QQRrSStTQQRrssTt
QqRrSSttQQRRSstT	QqRrSSttQqrRSSTT	QqRrSSttqqrrsSTt
QqRrSSttqQRrsstt	QqRrSstTqqrrSStt	QqRrSstTqqRRsstt
QqRrSstTQQrrSSTT	QqRrSstTQQRRssTT	QqRrssTTQQRRsSTt
QqRrssTTQqrRsstt	QqRrssTTqqrrSstT	QqRrssTTqQRrSSTT
QqRrsSTtqqrrssTT	QqRrsSTtqqRRSSTT	QqRrsSTtQQrrsstt
QqRrsSTtQQRRSStt	QqRRsSTTQQRrssTt	QqRRsSTTQqRRSstt
QqRRsSTTqqrRSStT	QqRRsSTTqQrrsSTT	QqRRSsttQQRrSStT
QqRRSsttQqRRsSTT	QqRRSsttqqrRssTt	QqRRSsttqQrrSstt
QRQRStStQRqrsTSt	QRQRStStQrqRSTst	QRQRStStqrQRStsT
QRQRStStqRQrstST	QRQRStsTqRqRstST	QRQRStsTqrQRsTsT
QRQRStsTQrQrSTst	QRQRStsTQRqrStSt	QRQRsTsTQRqrStsT
QRQRsTsTQrqRstST	QRQRsTsTqrQRsTSt	QRQRsTsTqRQrSTst
QRQRsTStqRqRSTst	QRQRsTStqrQRStSt	QRQRsTStQrQrstST
QRQRsTStQRqrsTsT	QRQrsTstqrqRsTst	QRQrsTstqrQrStst

QRQrsTstQRqRsTST	QRQrsTstQRQrStST	QRQrStSTqrqRStST
QRQrStSTqrQrsTST	QRQrStSTQRqRStst	QRQrStSTQRQrsTst
QRqrStStQRQRStsT	QRqrStStQrqRSTST	QRqrStStqrqrsTSt
QRqrStStqRQrstst	QRqrStsTqRqRstst	QRqrStsTqrqrStSt
QRqrStsTQrQrSTST	QRqrStsTQRQRsTsT	QRqrsTsTQRQRsTSt
QRqrsTsTQrqRstst	QRqrsTsTqrqrStsT	QRqrsTsTqRQrSTST
QRqrsTStqRqRSTST	QRqrsTStqrqrsTsT	QRqrsTStQrQrstst
QRqrsTStQRQRStSt	QRqRsTSTQRQrsTst	QRqRsTSTQRqRStst
QRqRsTSTqrQrsTST	QRqRsTSTqrqRStST	QRqRStstQRQrStST
QRqRStstQRqRsTST	QRqRStstqrQrStst	QRqRStstqrqRsTst
QQQQRRrrSsSstTTt	QQQQRRrrsSsSTttT	QQQQRrRrSSsstTTt
QQQQRrRrssSSTttT	QQQqRrRRSSSstTtt	QQQqRrRRsssSTtTT
QQQqRRrRSSSsttTt	QQQqRRrRsssSTTtT	QQQqRRRrSSsSttTt
QQQqRRRrSsSStTtt	QQQqRRRrssSsTTtT	QQQqRRRrsSssTtTT
QQqqRRRRSssStTtT	QQqqRRRRsSSsTtTt	QQqqRrrRSSSStTtT
QQqqRrrRssssTtTt	QQqQRrrrSSsStTTT	QQqQRrrrssSsTttt
QQqQRRrRSSSstttT	QQqQRRrRSssstTTT	QQqQRRrRsssSTTTt
QQqQRRrRsSSSTttt	QQqQRRRrSSsSttTT	QQqQRRRrssSsTTTt
QqqQRRrrSsSstttt	QqqQRRrrsSsSTTTT	QqqQRrRrSSsstttt
QqqQRrRrssSSTTTT	QqqqRrRRSssstTtt	QqqqRrRRsSSSTtTT
QqqqRrrrSSsSttTt	QqqqRrrrSsSStTtt	QqqqRrrrssSsTTtT
QqqqRrrrsSssTtTT	QqqqRRrRSsssttTt	QqqqRRrRsSSSTTtT
QqQqRRRRSssSttTT	QqQqRRRRsSSsTTtt	QqQqRrrRSSSSttTT
QqQqRrrRssssTTtt	QqQQRrRRSSSstttT	QqQQRrRRSssstTTT
QqQQRrRRsssSTTTt	QqQQRrRRsSSSTttt	QqQQRrrrSsSStTTT
QqQQRrrrsSssTttt	QqQQRRRrSsSStttT	QqQQRRRrsSssTTTt
QQRRQQrrSstTSsTt	QQRRQQrrsSTtsStT	QQRRQqrRSStttSsTt

QQRRQqrRssTTsStT	QQRRQqRrSSttsSTt	QQRRQqRrSstTSStt
QQRRQqRrssTTSstT	QQRRQqRrsSTtssTT	QQRRqqRRSstTsStT
QQRRqqRRsSTtSsTt	QQRRqQrRSSttSstT	QQRRqQrRSstTssTT
QQRRqQrRssTTsSTt	QQRRqQrRsSTtSStt	QQRRqQRrSSttsStT
QQRRqQRrssTTSsTt	QQRrqQrrSStTsSTT	QQRrqQrrssTtSstt
QQRrqqrRSStTSStT	QQRrqqrRssTtssTt	QQRrQqRRSStTSstt
QQRrQqRRssTtsSTT	QQRrQQRrSStTssTt	QQRrQQRrssTtSStT
QqRrQQRRSSttSstT	QqRrQQRRSstTssTT	QqRrQQRRssTTsSTt
QqRrQQRRsSTtSStt	QqRrQQrrSstTSSTT	QqRrQQrrsSTtsstt
QqRrQqrRSSttSSTT	QqRrQqrRssTTsstt	QqRrqqRRSstTsstt
QqRrqqRRsSTtSSTT	QqRrqqrrSSttsSTt	QqRrqqrrSstTSStt
QqRrqqrrssTTSstT	QqRrqqrrsSTtssTT	QqRrqQRrSSttsstt
QqRrqQRrssTTSSTT	QqRRqQrrSsttSstt	QqRRqQrrsSTTsSTT
QqRRqqrRSsttssTt	QqRRqqrRsSTTSStT	QqRRQqRRSsttsSTT
QqRRQqRRsSTTSstt	QqRRQQRrSsttSStT	QqRRQQRrsSTTssTt
QRQRQRqrStStsTSt	QRQRQRqrStsTStSt	QRQRQRqrsTsTStsT
QRQRQRqrsTStsTsT	QRQRQrqRStStSTst	QRQRQrqRsTsTstST
QRQRQrQrStsTSTst	QRQRQrQrsTStstST	QRQRqrQRStStStsT
QRQRqrQRStsTsTsT	QRQRqrQRsTsTsTSt	QRQRqrQRsTStStSt
QRQRqRqRStsTstST	QRQRqRqRsTStSTst	QRQRqRQrStStstST
QRQRqRQrsTsTSTst	QRQrqrqRStSTStST	QRQrqrqRsTstsTst
QRQrqrQrStSTsTST	QRQrqrQrsTstStst	QRQrQRqRStSTStst
QRQrQRqRsTstsTST	QRQrQRQrStSTsTst	QRQrQRQrsTstStST
QRqrQRQRStStStsT	QRqrQRQRStsTsTsT	QRqrQRQRsTsTsTSt
QRqrQRQRsTStStSt	QRqrQrqRStStSTST	QRqrQrqRsTsTstst
QRqrQrQrStsTSTST	QRqrQrQrsTStstst	QRqrqrqrStStsTSt
QRqrqrqrStsTStSt	QRqrqrqrsTsTStsT	QRqrqrqrsTStsTsT

QRqrqRqRStsTstst	QRqrqRqRsTStSTST	QRqrqRQrStStstst
QRqrqRQrsTsTSTST	QRqRqrqRStstsTst	QRqRqrqRsTSTStST
QRqRqrQrStstStst	QRqRqrQrsTSTsTST	QRqRQRqRStstsTST
QRqRQRqRsTSTStst	QRqRQRQrStstStST	QRqRQRQrsTSTsTst

L = 40 (Incomplete, with possibly isomorphic pairs)

QQQQqQqqQRrRrrrRRrr	QQQQqQqqQRRrrRRRrRr
QQQqQQqqQRrRrRRRRrr	QQQqQQqqQRRrrrrRrRr
QQqqQQQqQqRRRRrRrrR	QQqqQQQqQqRrrRrRRRR
QQqqQqQqQQRRrrrrrRrr	QQqqQqQqQQRRrRRRRrr
QQqqqqQqQqRRRRrRRrrR	QQqqqqQqQqRrrRRrRRRR
QQqqqqqQqqRRrRrRrrRR	QQqqqqqQqqRRrrRrRrRR
QQqQqQqqQQRRrrrrrRrr	QQqQqQqqQQRRrRRRRRrr
QQqQQQQqqRRrRrRrrRR	QQqQQQQqqRRrrRrRrRR
QqqQQQQQqRrrrRrRrrR	QqqQQQQQqRrrRrRrrrR
QqqQQqQQQQRrRrRRRRrr	QqqQQqQQQQRRrrrrRrRr
QqqQqQqqqQRrrRRRRRr	QqqQqQqqqQRrrrrrrRRr
QqqQqQQQQQRrRrrrRRrr	QqqQqQQQQQRRrrRRRrRr
QqqqqqqQQqRrrrRrRrrR	QqqqqqqQQqRrrRrRrrrR
QqqqQqQqqQRrrRRRRRr	QqqqQqQqqQRrrrrrrRRr
QqQqQQQqqRRRRrRRrrR	QqQqQQQqqRrrRRrRRRR
QqQqqqQQqqRRRRRrRrrR	QqQqqqQQqqRrrRrRRRRR
QQQqqQqRRQqRRRrrRrR	QQQqqqQRRqQRRrRRrRr
QQqqqqQqRRQqRRRrRrrR	QQqqqqqQRRqQRRrRrRRr
QqqQqQqqRrqqRrrrrrRR	QqqQqQQQRrQQRrRRRRrr
QqQqqQqqRrqqRrrrRRRR	QqQqqQQQRrQQRrRRrrrr

QQQQqqRRQqRRQqRrrRrR	QQQQqqRRqQRRqQrRRrRr
QQqqqqRRQqRRQqRrRrrR	QQqqqqRRqQRRqQrRrRRr
QqqQqQRrQQRrQQRRRRrr	QqqQqQRrqqRrqqrrrrRR
QqQqqQRrQQRrQQRRrrrr	QqQqqQRrqqRrqqrrRRRR
QQQQqQRRRQrqqQrRrRRr	QQqqqqRRrqRQqQRRRrRr
QqqQqQRrrQRqqqRrRRRR	QqQqqqRrRqrQqqrrrrRR
QQQqRRRrQqrrQQqQrrRr	QQQqRRRrqQRRQQqQrrRr
QQqQRRrRQqRRQQQqrrrR	QQqQRRrRqQrrQQQqrrrR
QqqqRrrrQQRrQqQQrRrr	QqqqRrrrqqrRQqQQrRrr
QqQQRrRRQQrRQqqqrRRR	QqQQRrRRqqRrQqqqrRRR
QQqqRqRrRRQQqQRQRRrr	QQqqRqRrRRqqQqrqrrRR
QQqQRQRRrrqqQQrQrRrr	QQqQRQRRrrQQqqRqRrRR
QqqQRQRRRrqQQQrQrRRr	QqqQRQRRRrQqqqRqRrrR
QqqqRqRrrRQqqQRQRRRr	QqqqRqRrrRqQQqrqrrrR
QQqRqRqrRRQQqRQRQRrr	QQqRqRqrRRqqQrqrqrRR
QQqRQRQRrrqqQrQrQRrr	QQqRQRQRrrQQqRqRqrRR
QqqRQRQRRrqQQrQrQRRr	QqqRQRQRRrQqqRqRqrrR
QqqRqRqrrRQqqRQRQRRr	QqqRqRqrrRqQQrqrqrrR
QQRRQqRrRRqQQQrrqQRr	QQRRQqRrrrQqQQrrqQRr
QQRRqQrRRRQqQQrrQqrR	QQRRqQrRrrqQQQrrQqrR
QqRrQQRRRrqqQqrRqqRR	QqRrQQRRrRQQQqrRqqRR
QqRrqqrrRrQQQqrRQQrr	QqRrqqrrrRqqQqrRQQrr
QQRRQrQqrRqQrRQRqqRR	QQRRQrQqrRQqRrqrQQrr
QQRRqRQqrRqQrRqrqqRR	QQRRqRQqrRQqRrQRQQrr
QqRrQRQQrrQQRRqRQqrR	QqRrQRQQrrqqrrQrqQRr
QqRrqrQQrrQQRRQrQqrR	QqRrqrQQrrqqrrqRqQRr
QRRQqQqrrQRqqRRrRqqR	QRRQqQQRRQrqqRrRrQQr

QRrqqqQRrqRQqRRRqQr

QRQRQrQrQRqRQRqrqRQr

QRQRqrQRQrQRQrQrqRqR

QRQRqrQrqrQrqrqRQrQr

QRQRqrqrQrqrQrqRQrQr

QRQRqrqRqrqRqrQrqRqR

QRQRqRqRQRQrQRqrQrqR

QRQrqrqRQrQRQRqRQRqR

QRQrQRQrQrqrQRqRqrQr

QRqrQrqRQRQrQRQRqRqR

QRqrqrQRQrQRQrQrQrqR

QRqrqrQrqrQrqrqRqRQr

QRqrqrqrQrqrQrqRqRQr

QRqrqrqRqrqRqrQrQrqR

QRqrqRQrQRqRQRQRQrQr

QRqRqrQrQrqrQRQrQRQr

QRqRQRqRQrQRQRQrqrqR

QRQRRqQrqRqrQRrqqRqR

QRQRrQQrqRqrQRRQqRqR

QRqrRQQrQrQRQRRqQrqR

QRqrrqQrQrQRQRrQQrqR

QQRrQQRrqqrRSStTQqRR

QQRrqqrRqqrRSStTQqRR

QqRRqQrrqQrrsSTTQQRr

QqRRQqRRqQrrsSTTQQRr

QRQrQRQrqrqRStSTQRqR

QRQrqrqRqrqRStSTQRqR

QRrqqqqrRqrQqRrrrQqR

QRQRQrQrqrQrQRqrqRQr

QRQRqrQRqRQRqRqRQrQr

QRQRqrQrQRQrQRQrqRqR

QRQRqrqrqRqrqRQrqRqR

QRQRqrqRQRqRQRqRQrQr

QRQRqRqRqrqRQRqrQrqR

QRQrqrqRqRqrQRqRQRqR

QRQrQRQrqRQRQRqRqrQr

QRqrQrqRqrqRQRQRqRqR

QRqrqrQRqRQRqRqRqRQr

QRqrqrQrQRQrQRQrQrqR

QRqrqrqrqRqrqRQrQrqR

QRqrqrqRQRqRQRqRqRQr

QRqrqRQrqrQrQRQRQrQr

QRqRqrQrqRQRQRQrQRQr

QRqRQRqRqRqrQRQrqrqR

QRQRRqQrqRQRqrRQQrQr

QRQRrQQrqRQRqrrqQrQr

QRqrRQQrQrqrqrrQqRQr

QRqrrqQrQrqrqrRqqRQr

QQRrQQRrqqrRssTtqQrr

QQRrqqrRqqrRssTtqQrr

QqRRqQrrqQrrSsttqqrR

QqRRQqRRqQrrSsttqqrR

QRQrQRQrqrqRsTstqrQr

QRQrqrqRqrqRsTstqrQr

QRqRqrQrqrQrsTSTQRQr	QRqRqrQrqrQrStstqrqR
QRqRQRqRqrQrsTSTQRQr	QRqRQRqRqrQrStstqrqR
QQRrQqRRssTtsSTTQqRR	QQRrQqRRssTtsSTTqQrr
QQRrqQrrSStTsSTTQqRR	QQRrqQrrSStTsSTTqQrr
QqRRqqrRsSTTSStTQQRr	QqRRqqrRsSTTSStTqqrR
QqRRQQRrSsttSStTQQRr	QqRRQQRrSsttSStTqqrR
QRQrQRqRsTstsTSTQRqR	QRQrQRqRsTstsTSTqrQr
QRQrqrQrStSTsTSTQRqR	QRQrqrQrStSTsTSTqrQr
QRqRqrqRsTSTStSTQRQr	QRqRqrqRsTSTStSTqrqR
QRqRQRQrStstStSTQRQr	QRqRQRQrStstStSTqrqR
QQRrQqRRssTtsSTTQqRR	QQRrQqRRssTtsSTTqQrr
QQRrqQrrSStTsSTTQqRR	QQRrqQrrSStTsSTTqQrr
QqRRqqrRsSTTSStTQQRr	QqRRqqrRsSTTSStTqqrR
QqRRQQRrSsttSStTQQRr	QqRRQQRrSsttSStTqqrR
QRQrQRqRsTstsTSTQRqR	QRQrQRqRsTstsTSTqrQr
QRQrqrQrStSTsTSTQRqR	QRQrqrQrStSTsTSTqrQr
QRqRqrqRsTSTStSTQRQr	QRqRqrqRsTSTStSTqrqR
QRqRQRQrStstStSTQRQr	QRqRQRQrStstStSTqrqR
QQRrQQRrqqrRSStTQqRR	QQRrQQRrqqrRssTtqQrr
QQRrqqrRqqrRSStTQqRR	QQRrqqrRqqrRssTtqQrr
QqRRqQrrqQrrsSTTQQRr	QqRRqQrrqQrrSsttqqrR
QqRRQqRRqQrrsSTTQQRr	QqRRQqRRqQrrSsttqqrR
QRQrQRQrqrqRStSTQRqR	QRQrQRQrqrqRsTstqrQr
QRQrqrqRqrqRStSTQRqR	QRQrqrqRqrqRsTstqrQr
QRqRqrQrqrQrsTSTQRQr	QRqRqrQrqrQrStstqrqR
QRqRQRqRqrQrsTSTQRQr	QRqRQRqRqrQrStstqrqR
QQQQqQqqQRrRrrrRRrr	QQQQqQqqQRRrrRRRrRr

QQQQqQQqqQRrRrRRRRrr	QQQQqQQqqQRRrrrrRrRr
QQqqQQQqQqRRRRRrRrrR	QQqqQQQqQqRrrRrRRRRR
QQqqQqQqQQRRrrrrrRrr	QQqqQqQqQQRRrRRRRRrr
QQqqqqQqQqRRRRrRRrrR	QQqqqqQqQqRrrRRrRRRR
QQqqqqqQqqRRrRrRrrRR	QQqqqqqQqqRRrrRrRrRR
QQqQqQqqQQRRrrrrrRrr	QQqQqQqqQQRRrRRRRRrr
QQqQQQQQqqRRrRrRrrRR	QQqQQQQQqqRRrrRrRrRR
QqqQQQQQQqRrrrRrRrrR	QqqQQQQQQqRrrRrRrrrR
QqqQQqQQQQRrRrRRRRrr	QqqQQqQQQQRRrrrrRrRr
QqqQqQqqqQRrrRRRRRRr	QqqQqQqqqQRrrrrrrRRr
QqqQqQQQQQRrRrrrRRrr	QqqQqQQQQQRRrrRRRrRr
QqqqqqqQQqRrrrRrRrrR	QqqqqqqQQqRrrRrRrrrR
QqqqQqQqqQRrrRRRRRRr	QqqqQqQqqQRrrrrrrRRr
QqQqQQQQqqRRRRrRRrrR	QqQqQQQQqqRrrRRrRRRR
QqQqqqQQqqRRRRRrRrrR	QqQqqqQQqqRrrRrRRRRR
QQQQqqQqRRQqRRRrrRrR	QQQQqqqQRRqQRRrRRrRr
QQqqqqQqRRQqRRRrRrrR	QQqqqqqQRRqQRRrRrRRr
QqqQqQqqRrqqRrrrrrRR	QqqQqQQQRrQQRrRRRRrr
QqQqqQqqRrqqRrrrRRRR	QqQqqQQQRrQQRrRRrrrr
QQQQqqRRQqRRQqRrrRrR	QQQQqqRRqQRRqQrRRrRr
QQqqqqRRQqRRQqRrRrrR	QQqqqqRRqQRRqQrRrRRr
QqqQqQRrQQRrQQRRRRrr	QqqQqQRrqqRrqqrrrrRR
QqQqqQRrQQRrQQRRrrrr	QqQqqQRrqqRrqqrrRRRR
QQQQqQRRRQrqqQrRrRRr	QQqqqqRRrqRQqQRRRrRr
QqqQqQRrrQRqqqRrRRRR	QqQqqqRrRqrQqqrrrrRR
QQQqRRRrQqrrQQqQrrRr	QQQqRRRrqQRRQQqQrrRr
QQqQRRrRQqRRQQQqrrrR	QQqQRRrRqQrrQQQqrrrR

QqqqRrrrQQRrQqQQrRrr	QqqqRrrrqqrRQqQQrRrr
QqQQRrRRQQrRQqqqrRRR	QqQQRrRRqqRrQqqqrRRR
QQqqRqRrRRQQqQRQRRrr	QQqqRqRrRRqqQqrqrrRR
QQqQRQRRrrqqQQrQrRrr	QQqQRQRRrrQQqqRqRrRR
QqqQRQRRRrqQQQrQrRRr	QqqQRQRRRrQqqqRqRrrR
QqqqRqRrrRQqqQRQRRRr	QqqqRqRrrRqQQqrqrrrR
QQqRqRqrRRQQqRQRQRrr	QQqRqRqrRRqqQrqrqrRR
QQqRQRQRrrqqQrQrQRrr	QQqRQRQRrrQQqRqRqrRR
QqqRQRQRRrqQQrQrQRRr	QqqRQRQRRrQqqRqRqrrR
QqqRqRqrrRQqqRQRQRRr	QqqRqRqrrRqQQrqrqrrR
QQRRQqRrRRqQQQrrqQRr	QQRRQqRrrrQqQQrrqQRr
QQRRqQrRRRQqQQrrQqrR	QQRRqQrRrrqQQQrrQqrR
QqRrQQRRRrqqQqrRqqRR	QqRrQQRRrRQQQqrRqqRR
QqRrqqrrRrQQQqrRQQrr	QqRrqqrrrRqqQqrRQQrr
QQRRQrQqrRqQrRQRqqRR	QQRRQrQqrRQqRrqrQQrr
QQRRqRQqrRqQrRqrqqRR	QQRRqRQqrRQqRrQRQQrr
QqRrQRQQrrQQRRqRQqrR	QqRrQRQQrrqqrrQrqQRr
QqRrqrQQrrQQRRQrQqrR	QqRrqrQQrrqqrrqRqQRr
QRRQqQqrrQRqqRRrRqqR	QRRQqQQRRQrqqRrRrQQr
QRrqqqQRrqRQqRRRRqQr	QRrqqqqrRqrQqRrrrQqR
QRQRQrQrQRqRQRqrqRQr	QRQRQrQrqrQrQRqrqRQr
QRQRqrQRQrQRQrQrqRqR	QRQRqrQRqRQRqRqRQrQr
QRQRqrQrqrQrqrqRQrQr	QRQRqrQrQRQrQRQrqRqR
QRQRqrqrQrqrQrqRQrQr	QRQRqrqrqRqrqRQrqRqR
QRQRqrqRqrqRqrQrqRqR	QRQRqrqRQRqRQRqRQrQr
QRQRqRqRQRQrQRqrQrqR	QRQRqRqRqrqRQRqrQrqR
QRQrqrqRQrQRQRqRQRqR	QRQrqrqRqRqrQRqRQRqR

QRQrQRQrQrarQRaRarQr	QRQrQRQraRQRQRaRarQr
QRarQraRQRQrQRQRaRaR	QRarQraRaraRQRQRaRaR
QRararQRQrQRQrQrQraR	QRararQRaRQRaRaRaRQr
QRararQrarQraraRaRQr	QRararQrQRQrQRQrQraR
QRarararQrarQraRaRQr	QRarararaRaraRQrQraR
QRararaRaraRarQrQraR	QRararaRQRaRQRaRaRQr
QRaraRQrQRaRQRQRQrQr	QRaraRQrarQrQRQRQrQr
QRaRarQrQrarQRQrQRQr	QRaRarQraRQRQRQrQRQr
QRaRQRaRQrQRQRQrarar aR	QRaRQRaRaRarQRQrararaR
QRQRRaQraRarQRraaRaR	QRQRRaQraRQRarRQQrQr
QRQRrQQraRarQRRQaRaR	QRQRrQQraRQRarraQrQr
QRarRQQrQrQRQRRaQraR	QRarRQQrQrararrQaRQr
QRarraQrQrQRQRrQQraR	QRarraQrQrararRaaRQr

Appendix 3: Known Minimal SCOSoV's

The following SCOSoV's will be expanded on a basis from the following selection. The number of basis vectors required is indicated by the bracketed number following each SCOSoV. Lower-case characters indicate negative vectors.

Bases:

(A)

(A,B)

(A,B,C)

(A,B,C,D)

...

1: A(1)

2: AB(2)

3: ABC(3)

4: ABAb(2)

5: ABCAb(3)

6: AAaBCB(3)

7: AAaBCBD(4)

8: ABAbABaB(2)

9: ABAbCABaB(3)

10: AAaABABBbb(2)

11: AAaABABBbbC(3)

16: ABAbABaBABAbabAb(2)

17: ABAbABaBCABAbabAb(3)

20: AABBAaBbbbAaAAbbaABb(2)

21: AABBAaBbbbAaAAbbaABbC(3)

26: AaAAaaAaaaaABABBBbbBBBbBb(2)

27: AaAAaaAaaaaABABBBbbBBBbBbC(3)

28: ABCdABCdabcDaBCDABcDAbCDABcD(4)

32: ABAbABaBABAbabAbabaBabAbABAbabAb(2)

33: ABAbABaBABAbabAbCabaBabAbABAbabAb(3)

Bibliography

[1] R. H. Barker, "Group Synchronizing of Binary Digital Systems", COMMUNICATION THEORY, Butterworth, London, Pp. 273-287, 1953.

[2] R. L. Frank, "Polyphase Codes with Good Nonperiodic Correlation Properties", IEEE TRANSACTIONS ON INFORMATION THEORY, IT-9, Pp. 43-45, January, 1963.

[3] M. J. E. Golay, "Multislit Spectrometry", JOURNAL OF THE OPTICAL SOCIETY OF AMERICA, Vol. 39, Pp. 437-444, 1949.

[4] M. J. E. Golay, "Static Multislit Spectrometry and its Application to the Panoramic Display of Infrared Spectra", JOURNAL OF THE OPTICAL SOCIETY OF AMERICA, Vol. 41, Pp. 468-472, 1951.

[5] M. J. E. Golay, "Complementary Series", IRE TRANSACTIONS ON INFORMATION THEORY, Vol. IT-7, Pp. 82-87, April 1961.

[6] M. J. E. Golay, "Note on Complementary Series", PROCEEDINGS OF THE IRE, p. 84, January 1962.

[7] J. Hadamard, "Resolution d'une Question Relative aux Determinants", BULLETIN DES SCIENCES MATHEMATIQUES, (2), Vol. 17, Part 1, Pp. 240-246, 1893.

[8] Stephen Jauregui, Jr., "Complementary Sequences of Length 26", IRE TRANSACTIONS ON INFORMATION THEORY, Vol. IT-7, p. 323, July 1962.

[9] Gordon S. Kino and John Shaw, "Acoustic Surface Waves", SCIENTIFIC AMERICAN, Vol. 227, No. 4, Pp. 50-68, October 1972.

[10] Joseph B. Kruskal, "Golay's Complementary Series", IRE TRANSACTIONS ON INFORMATION THEORY, IT-7, Pp. 273-276, October 1961.

[11] R. E. A. C. Paley, "On Orthogonal Matrices", JOURNAL OF MATHEMATICS AND PHYSICS, M.I.T., Vol. 12, Pp. 311-320, 1933.

[12] Hans Rademacher and Otto Toeplitz, "On Waring's Problem", THE ENJOYMENT OF MATHEMATICS, Princeton University Press, Princeton, New Jersey, Pp. 52-61, 1957.

[13] Bernard Schweitzer, GENERALIZED COMPLEMENTARY CODE SETS, Ph.D. Thesis, U.C.L.A., 1971.

[14] W. M. Siebert, "A Radar Detection Philosophy", IRE TRANSACTIONS ON INFORMATION THEORY, Vol. IT-2, Pp. 204-221, September 1956.

[15] J. M. Speiser and H. J. Whitehouse, "Surface Wave Transducer Array Design Using Transversal Filter Concepts", ACOUSTIC SURFACE WAVE AND ACOUSTO-OPTIC DEVICES, Optosonic Press, New York, Pp. 81-90, 1971.

[16] William D. Squire, Harper J. Whitehouse, and J. M. Alsup, "Linear Signal Processing and Ultrasonic Transversal Filters", IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNOLOGY, Vol. MTT-17, Pp. 1020-1040, November 1969.

[17] Y. Taki, H. Miyakawa, M. Hatori, and S. Namba, "Even-Shift Orthogonal Sequences", IEEE TRANSACTIONS ON INFORMATION THEORY, Vol. IT-15, Pp. 295-300, March 1969.

[18] Chin-Chong Tseng, "Signal Multiplexing in Surface-Wave Delay Lines Using Orthogonal Pairs of Golay's Complementary Sequences", IEEE TRANSACTIONS ON SONICS AND ULTRASONICS, Vol. SU-18, Pp. 103-107, April 1971.

[19] C. C. Tseng and C. L. Liu, "Complementary Sets of Sequences", IEEE TRANSACTIONS ON INFORMATION THEORY, Vol. IT-18, Pp. 644-652, September 1972

[20] Richard Turyn, "Ambiguity Functions of Complementary Sequences", IEEE TRANSACTIONS ON INFORMATION THEORY, Vol. IT-9, Pp. 46-47, January 1963.

[21] H. Van de Vaart and L. R. Schissler, "Acoustic Surface-Wave Recirculating Memory", IEEE TRANSACTIONS ON SONICS AND ULTRASONICS, Vol. SU-20, Pp. 154-161, April 1973.

[22] Jennifer Seberry Wallis, "Hadamard Matrices" COMBINATORICS: ROOM SQUARES, SUM-FREE SETS, HADAMARD MATRICES, Springer-Verlag, New York, Pp. 273-489, 1972.

[23] George R. Welti, "Quaternary Codes for Pulsed Radar", IRE TRANSACTIONS ON INFORMATION THEORY, Vol. IT-6, Pp. 400-408, June 1960.

[24] A. M. Yaglom, STATIONARY RANDOM FUNCTIONS, Dover Publicatons Inc., New York, 1962.