Family of Multicarrier Bi-Phase Radar Signals Represented by Ternary Arrays

MESHULIM B. SVERDLIK
Israel

NADAV LEVANON, Life Fellow, IEEE
Tel Aviv University

A $K \times L$ ternary array, comprised of the elements $\{0, 1, -1\}$, with some unique features, represents a multicarrier radar signal with favorable autocorrelation and ambiguity functions. Constructing such an array using Galois fields is described. As in a Costas binary array, only one frequency is transmitted at any time slot, but in our array the same frequency is repeated in several time slots, yielding a signal with considerably larger pulse compression than a Costas signal that uses the same number of frequencies.

Manuscript received March 9, 2005; revised October 3, 2005; released for publication January 9, 2006.

IEEE Log No. T-AES/42/3/884464.

Refereeing of this contribution was handled by V. Chen.

Authors' addresses: M. B. Sverdlik, 37 Hareshet St., Bat Yam, 59462, Israel; N. Levanon, Dept. of Electrical Engineering—Systems, Tel Aviv University, Tel Aviv, 69978, Israel.

I. INTRODUCTION

Multicarrier (or multifrequency) signals can be divided into two categories: 1) signals in which at any given time only one frequency is transmitted, and 2) signals with simultaneous transmission on several subcarriers. Costas signals [1] belong to the first category. Orthogonal frequency division multiplexing (OFDM) signals [2] used in communications, and multicarrier complementary phase-coded (MCPC) signals [3] suggested for radar, belong to the second category. Signals in the first category usually (but not necessarily) exhibit constant amplitude, while signals of the second category exhibit variable amplitude.

A multicarrier signal can be described by a 2 dimensional array in which the rows represent frequencies and the columns represent time slots (which we refer to as bits). The array could be binary, as in Costas, where a 1 in the k,l element implies that the kth frequency is transmitted during the lth bit. A 0 implies no transmission corresponding to that element.

In OFDM and MCPC the array elements are complex numbers, usually with unity magnitude, implying phase coding. In all three signals mentioned, the frequency spacing Δf between subcarriers is related to the bit duration t_b according to

$$\Delta f = \frac{1}{t_b}. \tag{1}$$

This relationship implies orthogonality between the signals on the different subcarriers, when the integration time is a multiple of t_b.

The signal suggested here is based on a ternary array comprised of the elements $\{0, 1, -1\}$. This vocabulary allows for frequency coding, as in Costas, but with additional binary phase coding. A -1 in the k,l element implies that the kth frequency is transmitted during the lth bit, with reversed polarity (i. e., with a phase shift of π). An example of a 4×12 ternary array is given in (2), and a graphic representation is shown in Fig. 1.

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}. \tag{2}$$

A concise representation of the signal appears in Table I. It lists the bit location of each frequency (row). A negative sign indicates polarity reversal. The only requirement from the ternary array in (2) is that the discrete 2-D autocorrelation of the array, obtained, for example, by the MATLAB instruction xcorr 2(**A**), will have in its central row (no vertical shift) sidelobe levels of only 0 or ± 1.

It is interesting to compare the discrete autocorrelation (Fig. 2) of the simple 4×12 array

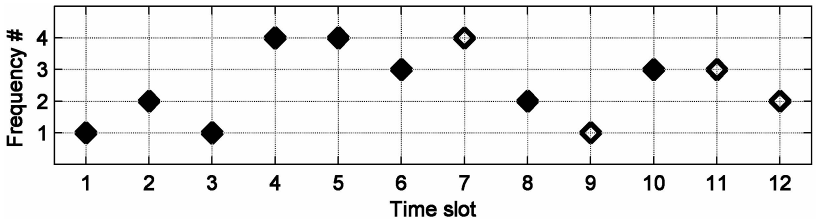


Fig. 1. Graphic representation of ternary array in (2).

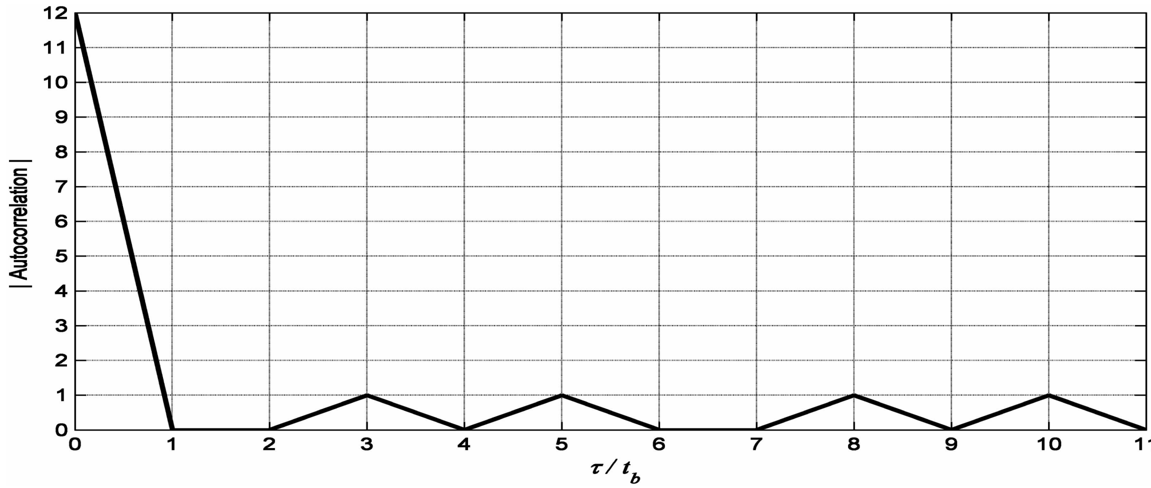


Fig. 2. Discrete autocorrelation of ternary array **A**.

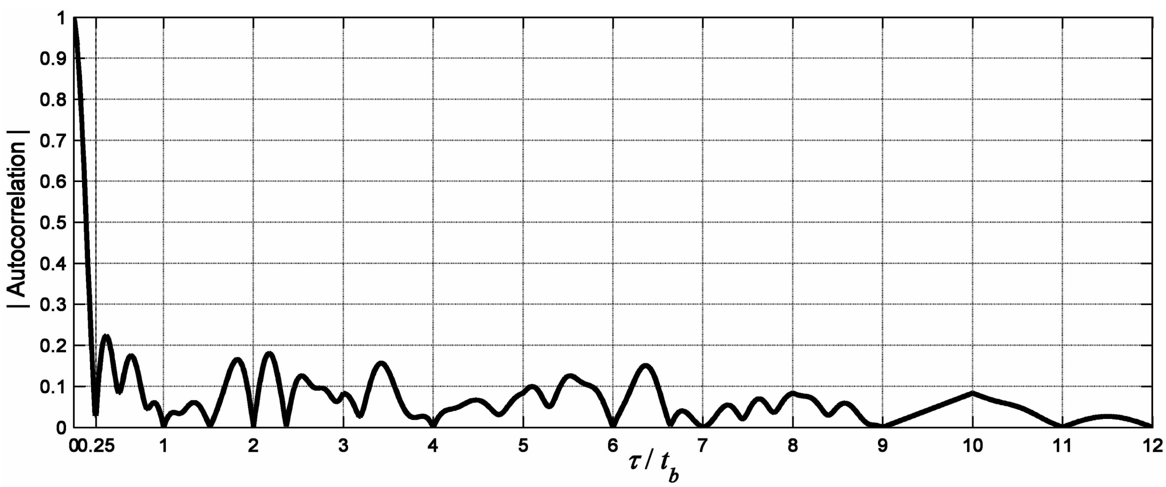


Fig. 3. Magnitude of autocorrelation function of signal based on ternary array **A**.

in (2) with the continuous autocorrelation functions (ACFs) of the actual signal based on it (Fig. 3). In the discrete array **A** the rows are decoupled and the central row of the discrete 2-D autocorrelation is the sum of the autocorrelations of the individual rows. On the other hand, when calculating the ACF of the corresponding multicarrier signal, we cannot assume such decoupling, except at delays equal to multiples of the bit duration. Indeed, at such delays the ACF in Fig. 3 exhibits either the value zero (at $\tau/t_b = 1,2,4,6,7,9,11$) or $1/12 = 0.083$ (at $\tau/t_b = 3,5,8,10$), which are the same values as in the corresponding delays of Fig. 2.

TABLE I
Concise Representation of Ternary Matrix in (2)

4	5	−7
6	10	−11
2	8	−12
1	3	−9

Another major difference between Figs. 2 and 3 is the width of the mainlobe. From the signal's ACF (Fig. 3), we learn, as expected, that the first null appears at a delay equal to the inverse of the bandwidth $K\Delta f$. Thus, since the number of frequencies is $K = 4$, we get

$$\frac{\tau_{\text{null}}}{t_b} = \frac{1}{K\Delta f}\frac{1}{t_b} = \frac{1}{K} = 0.25. \tag{3}$$

Since the number of bits is $L = 12$, the pulse compression ratio is

$$CR = \frac{Lt_b}{\tau_{\text{null}}} = LK = 48. \tag{4}$$

The construction of a multicarrier bi-phase signal is described in Section II. It is based on an early work by the first author [4]. In Section III we demonstrate the performances of a 40×120 signal. Section IV is dedicated to the construction of a family of multicarrier bi-phase signals. A comparison with a Costas signal is given in Section V, and conclusions in Section VI. Appendix A contains a Matlab program that implements the construction algorithm and yields one (or all) 40×120 basic signals. All possible 22 basic signals (arrays) are listed in Appendix B.

II. CONSTRUCTION OF MULTICARRIER BI-PHASE SIGNAL

Our discussion is limited to a construction based on Galois field GF(p^z) where p is a prime and z is an integer ≥ 1. The construction can be divided into four steps.

A. Construction of Ensemble with Property of No More Than One Coincidence

The first stage of the construction algorithm is based on GF(p^z). It calls for the creation of a balanced ensemble of pulse sequences with the feature of not more than one coincidence (ENMOC) [4]. By "balanced" we mean that every sequence in the ensemble has the same number of unity elements. A balanced ensemble of pulse sequences with the ENMOC feature is a family of binary sequences

$$a^i_j : j = 0,1,\ldots,N_i - 1, \quad i = 1,2,\ldots,Q, \quad a^i_j \in \{0,1\}, \quad a^i_0 = a^i_{N_i-1} = 1 \tag{5}$$

where N_i is the length of sequence number i, and the sequences obey the following two properties.

1) Each sequence exhibits a "no more than one coincidence" (NMOC) feature, which implies that the sidelobes of the discrete ACF of every NMOC sequence should not exceed unity, i.e.,

$$r(k) = \sum_{j=0}^{N_i-1-|k|} a_j a_{j+|k|} = \begin{cases} N_0, & k = 0 \\ 0 \text{ or } 1, & |k| \neq 0 \end{cases} \tag{6}$$

where N_0 is the number of 1 elements in the NMOC sequence.

2) The cross-correlation between the ith and the lth sequences should obey

$$x_{i,l}(k) = \left\{ \begin{array}{ll} \displaystyle\sum_{j=0}^{B-1-k} a^i_j a^l_{j+k}, & k \geq 0 \\ \displaystyle\sum_{j=-k}^{B-1} a^i_j a^l_{j+k}, & k < 0 \end{array} \right\} = 0 \text{ or } 1, \quad \text{for} \quad i,l = 1,2,\ldots,Q, \quad i \neq l \tag{7}$$

where $B = \max(N_i, N_l)$ is the size of the basis of the ensemble. Q in (5) and (7) is the number of the different pulse sequences that obey properties 1 and 2. The resulted ENMOC is said to be of power Q.

A detailed description for constructing an ENMOC is given in [4]. It is based on using the property of linear dependence of $p + 1$ elements of the extended Galois field GF(p^z) [4, 5]. A construction example of an ENMOC is also given in [4] and Table II of [4] lists some ENMOCs.

For the remainder of our paper we need to cite from [4] the following three equations:

$$N_0 = p + 1 \tag{8}$$

$$Q = p^{(z-3)} + p^{(z-5)} + \cdots + 1, \qquad z > 3, \quad z \equiv 1(\text{mod}\,2) \tag{9}$$

$$Q = \{p^{(z-4)} + p^{(z-6)} + \cdots + 1\}p, \qquad z > 4, \quad z \equiv 0(\text{mod}\,2). \tag{10}$$

It is known [4] that the elements of the field GF(p^z) are separated from each other by

$$N = \frac{p^z - 1}{p - 1}. \tag{11}$$

Therefore the numbers of elements of the field GF(p^z) will be reduced modulo N. The number of bits in the signal will be

$$L = N - 1. \tag{12}$$

For constructing a multicarrier signal with L bits, it is first necessary to form from the ENMOC an array that uses the results of automorphic transformation of all sequences of the ENMOC and exclude all unity elements that are placed at the zero time position. Then the number of all elements in the array become

$$N_{ar} = Q(p + 1)p. \tag{13}$$

Therefore to construct a signal with L bits we need to equate L and N_{ar}, namely

$$\frac{p^z - 1}{p - 1} - 1 = Q(p + 1)p. \tag{14}$$

For $z \equiv 1(\text{mod}\,2)$, solutions of (14) (using (9)) exist if p is a prime and

$$z = 5 + 2n, \qquad n = 0,1,2,\ldots. \tag{15}$$

It can be shown that increasing p will decrease the ratio K/L and raise the level of peak ACF sidelobes (without bi-phase modulation), and vice versa.

TABLE II
Sequences of ENMOC and Their Automotphism Results

$D_1 = (0,1,36,102)$	$D_2 = (0,35,101,120)$	$D_3 = (0,66,85,86)$	$D_4 = (0,19,20,55)$
$D_5 = (0,2,10,17)$	$D_6 = (0,8,15,119)$	$D_7 = (0,7,111,113)$	$D_8 = (0,104,106,114)$
$D_9 = (0,3,64,108)$	$D_{10} = (0,61,105,118)$	$D_{11} = (0,44,57,60)$	$D_{12} = (0,13,16,77)$
$D_{13} = (0,4,27,92)$	$D_{14} = (0,23,88,117)$	$D_{15} = (0,65,94,98)$	$D_{16} = (0,29,33,56)$
$D_{17} = (0,5,43,80)$	$D_{18} = (0,38,75,116)$	$D_{19} = (0,37,78,83)$	$D_{20} = (0,41,46,84)$
$D_{21} = (0,6,30,51)$	$D_{22} = (0,24,45,115)$	$D_{23} = (0,21,91,97)$	$D_{24} = (0,70,76,100)$
$D_{25} = (0,9,71,82)$	$D_{26} = (0,62,73,112)$	$D_{27} = (0,11,50,59)$	$D_{28} = (0,39,48,110)$
$D_{29} = (0,12,34,81)$	$D_{30} = (0,22,69,109)$	$D_{31} = (0,47,87,99)$	$D_{32} = (0,40,52,74)$
$D_{33} = (0,14,72,103)$	$D_{34} = (0,58,89,107)$	$D_{35} = (0,31,49,63)$	$D_{36} = (0,18,32,90)$
$D_{37} = (0,25,53,79)$	$D_{38} = (0,28,54,96)$	$D_{39} = (0,26,68,93)$	$D_{40} = (0,42,67,95)$

Taking into consideration the limited size of a journal paper, we choose for the remainder of the paper the case in which:

$$p = 3, \qquad z = 5. \tag{16}$$

Using (9), (11), and (12), this choice will result in $Q = 10$, $L = 120$. As will be deduced shortly the number of frequencies K is given by

$$K = (p+1)Q \tag{17}$$

yielding $K = 40$ frequencies.

The construction algorithm described in [4] requires an exhaustive search of all the sequences that can create the ENMOC. With a relatively large value of L (such as $L = 120$), the exhaustive search becomes very computing intensive. Instead, we propose here an improved algorithm.

In the process of our work on this family of signals it became clear that every ensemble fits three different primitive polynomials. Or, in other words, three different polynomials can create the same ensemble. These are known as p-conjugate polynomials [6]. This property can be written as follows:

Let $f^{(1)}(x)$, $f^{(2)}(x)$ and $f^{(3)}(x)$ be p-conjugate primitive polynomials. Of course they form different isomorphic extended GFs, with coordinates columns $A_i^{(1)}$, $A_i^{(2)}$ and $A_i^{(3)}$, $i = 0,1,\ldots,120$. If we denote the companion matrices corresponding to the three p-conjugate polynomials by H_1, H_2, H_3, then for arbitrary u and t [4]

$$A_{u+t}^{(1)} = H_1^u A_t^{(1)}, \qquad A_{u+t}^{(2)} = H_2^u A_t^{(2)}, \qquad A_{u+t}^{(3)} = H_3^u A_t^{(3)}, \qquad u+t = 1,2,\ldots,120. \tag{18}$$

In (18) the subscript of A represents the degree of the primitive elements of the polynomials. It is clear that the coordinates (columns) obey $A_{u+t}^{(1)} \neq A_{u+t}^{(2)} \neq A_{u+t}^{(3)}$, $u+t > 4$. Note that if the set $A_u^{(1)}$, $A_q^{(1)}$, $A_r^{(1)}$ belongs to the ENMOC $\tilde{s}$ then the sets $A_u^{(2)}$, $A_q^{(2)}$, $A_r^{(2)}$ and $A_u^{(3)}$, $A_q^{(3)}$, $A_r^{(3)}$ also belong to the ENMOC $\tilde{s}$.

The improved algorithm for constructing an ENMOC uses simultaneously the properties of linear dependence and of the p-conjugate primitive polynomials. It is much less computational intensive than the original construction. A Matlab version of it is given in Appendix A. The ENMOC listed in the first column of Table II was obtained with this program.

B. Construction of 40×120 Element Array

In the second step we find a 120 element array, in which all integers are different from each other. From the algorithm for constructing the ENMOC it became clear that the sequences resulted from the automorphism of the Q sequences that belong to an ENMOC, do not intersect with the Q sequences of the ENMOC. If a sequence that belongs to the ENMOC is designated by D_i,

$$D_i = (d_i^1, d_i^2, d_i^3, d_i^4) \tag{19}$$

the corresponding $N_0 - 1$ (= 3) automorphism results D_{i+k} can be obtained by

$$D_{i+k} = (d_i^1 - d_i^{k+1}, d_i^2 - d_i^{k+1}, d_i^3 - d_i^{k+1}, d_i^4 - d_i^{k+1})[\mathrm{mod}\,(L+1)], \qquad k = 1,2,\ldots,N_0 - 1. \tag{20}$$

So, for example, from $D_1 = (0,1,36,102)$ we get

$$\begin{aligned} D_2 &= (0-1, 1-1, 36-1, 102-1)[\mathrm{mod}\,(121)] \\ &= (120, 0, 35, 101) \end{aligned}$$

which, after sorting, becomes $D_2 = (0,35,101,120)$. Similarly we will get $D_3 = (0,66,85,86)$ and $D_4 = (0,19,20,55)$. In Table II the sequences of the ENMOC D_i, $i = 1,5,9,\ldots,37$ are listed in the first column, and the sequences created by automorphism, D_i, $i = (2,3,4),(6,7,8),\ldots,(38,39,40)$ are listed in columns 2 to 4. Note that the sequences resulted from automorphism satisfy condition (6) but not (7).

The 120 element binary array described above fits three different primitive polynomials of degree 5, irreducible above the field GF(3):

$$\begin{aligned} f^{(1)}(x) &= x^5 - x - 2 \\ f^{(2)}(x) &= x^5 - 2x^3 - 2x - 2 \\ f^{(3)}(x) &= x^5 - x^4 - 2x^3 - 2x^2 - 2x - 2. \end{aligned} \tag{21}$$

For more details look at Table VI.

All together there are $K = (p+1)Q = 40$ sequences in Table II. If we exclude from all sequences the unity elements at the common location of zero, leaving p (= 3) unity elements in each sequence, we obtain in the modified Table II a total of $L = p(p+1)Q = 120$ numbers, all different from each other. In other words we find all the numbers from 1 to 120 and each one appears only once.

C. Frequency Allocations

Converting the modified Table II (without the locations at 0) to a multicarrier signal involves assigning 40 carrier frequencies to the 40 sequences. One option for the assignment rule is that the normalized frequency $t_b f_n$ corresponding to the kth sequence will be

$$t_b f_k = 20.5 - k, \qquad k = 1,2,\ldots,40. \tag{22}$$

With this choice, for example, during time slots 1, 36, and 102 (which appear in sequence D_1) the transmitted subcarrier will be the one shifted $19.5/t_b$ off the center frequency. The resulted structure appears as the array **B** on the left hand half of Table III. Array **C**, on the right hand half of Table III, lists the delay differences within each row of **B**. Note that in **C** no number appears more than twice.

Representing a transmission of frequency k during bit (time slot) l, by placing a 1 in the (k,l) element of a 40×120 array, and writing 0 at unused elements, will create a binary array with exactly one 1 in each column and three 1s in each row.

The sidelobes of the central row of the discrete 2-D autocorrelation of a binary array created in this way will have values of 0, 1, or 2, relative to a mainlobe height of 120 (see Fig. 4), at locations specified by the numbers in array **C**, on the right hand half of Table III. The highest autocorrelation value is 2 because no number in array **C** appears there more than twice. For reasons that will become clear shortly, it is also required (and achieved by the above construction) that if a number in **C** is repeated twice, the repetition must not occur in the same row of **C**.

Note in Table III that we split the two arrays into 10 groups of 4 rows each. Each group of 4 rows corresponds to one row in Table II. Each row in Table II contains an NMOC sequence and its corresponding automorphism products. On the top group of 4 rows in array **C**, we marked all the repeats of delay differences in order to show that they occur within the group. This property appears in all the other groups (of 4 rows) in array **C**. This is an important structural feature of the synthesized signal. We name this property "separability." Separability provides the following two important properties. 1) It allows to change polarities of frequencies within a group in array **B**, that will affect only the sidelobes at delays listed in the corresponding group in array **C**.

TABLE III
Frequency-Delay Array (Left) and Corresponding Delay Differences

Bit location (**B** =)			Delay differences (**C** =)		
1	36	102	35	66	101
35	101	120	66	19	85
66	85	86	19	1	20
19	20	55	1	35	36
2	10	17	8	7	15
8	15	119	7	104	111
7	111	113	104	2	106
104	106	114	2	8	10
3	64	108	61	44	105
61	105	118	44	13	57
44	57	60	13	3	16
13	16	77	3	61	64
4	27	92	23	65	88
23	88	117	65	29	94
65	94	98	29	4	33
29	33	56	4	23	27
5	43	80	38	37	75
38	75	116	37	41	78
37	78	83	41	5	46
41	46	84	5	38	43
6	30	51	24	21	45
24	45	115	21	70	91
21	91	97	70	6	76
70	76	100	6	24	30
9	71	82	62	11	73
62	73	112	11	39	50
11	50	59	39	9	48
39	48	110	9	62	71
12	34	81	22	47	69
22	69	109	47	40	87
47	87	99	40	12	52
40	52	74	12	22	34
14	72	103	58	31	89
58	89	107	31	18	49
31	49	63	18	14	32
18	32	90	14	58	72
25	53	79	28	26	54
28	54	96	26	42	68
26	68	93	42	25	67
42	67	95	25	28	53

Frequency

2) It allows adding amplitude weighting to the different frequencies, that will not affect the cancellation of ACF sidelobes of level 2, as long as the frequencies of each group are clustered together. This "separability" property exists for other signals constructed in this way, with other values of p and z that satisfy (15), and the number of coincidences will remain $p-1$.

In addition to the separability property, the binary array exhibits a discrete autocorrelation (e.g., Fig. 4) with exactly N_0Q (= 40) sidelobes of level 2, N_0Q sidelobes of level 1, and $N_0Q - 1$ (= 39) sidelobes of level 0. This implies that after nullifying the sidelobes of level 2 (by converting the binary array into a ternary array) only N_0Q (= 40) non-zero sidelobes (all of them ± 1) will remain.

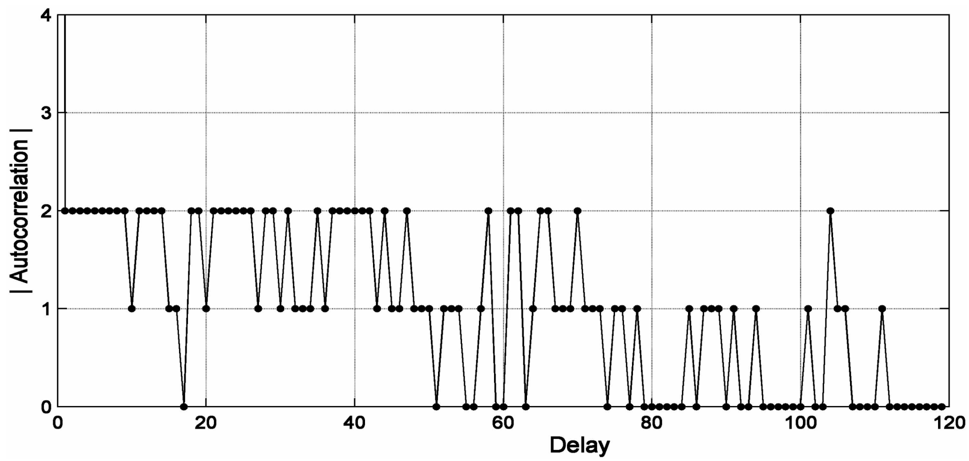


Fig. 4. Central row (positive delays) of 2-D autocorrelation of binary array.

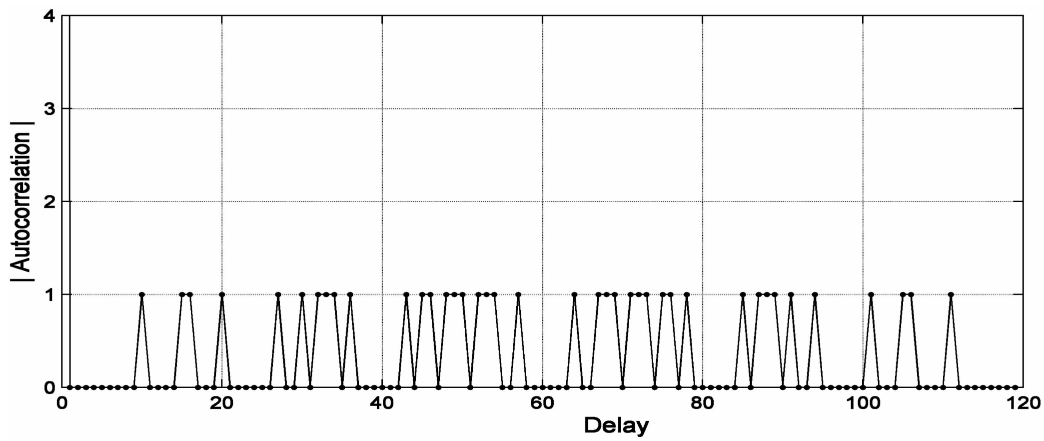


Fig. 5. Central row of 2-D autocorrelation of ternary array.

TABLE IV
SL Polarity Versus Matrix Element Polarity

b_1	b_2	b_3	c_1	c_2	c_3
+	+	+	+	+	+
−	+	+	−	+	−
+	−	+	−	−	+
+	+	−	+	−	−

D. Polarity Reversal Locations

The polarity reversal is intended to transform some of the 1 elements in the binary array, into -1 elements, in order to nullify all the ACF sidelobes of height 2. After polarity reversal the binary matrix will become a ternary matrix.

There are many different ways in which this could be achieved. With the construction algorithm discussed above, reversing the polarity of the bits corresponding to the third column of array **B** in Table III, will nullify all the sidelobes of height 2. The resulted discrete ACF (magnitude) is plotted in Fig. 5. This simple rule is not the only one applicable to binary arrays constructed by the algorithm, and it usually does not work with randomly generated arrays. Nullifying the level 2 sidelobes by polarity reversal can be done for any binary array that meets the four requirements: 1) exactly one 1 in each column, 2) exactly three 1s in each row, 3) any specific difference between locations of 1s within a row will not occur more than twice, in the entire array, and 4) if a location difference appears twice it must be in different rows.

The rules summarized in Table IV show how to reverse the polarity of a sidelobe. The first three columns in Table IV refer to the columns of array **B** in Table III, while the last three columns of Table IV refer to the columns of array **C** in Table III.

An example will show how to apply the polarity reversal rule. Consider, for example, the top four rows in Table III. From the top four rows in **C** we note that four delay differences appear twice: 1, 19, 35, and 66. These repetitions will cause ACF sidelobes of height 2 at these four delays. Starting from row 1, reversing the polarity of b_3 (= 102) will reverse the polarity of the sidelobes at delays c_2 (= 66) and c_3 (= 101), as

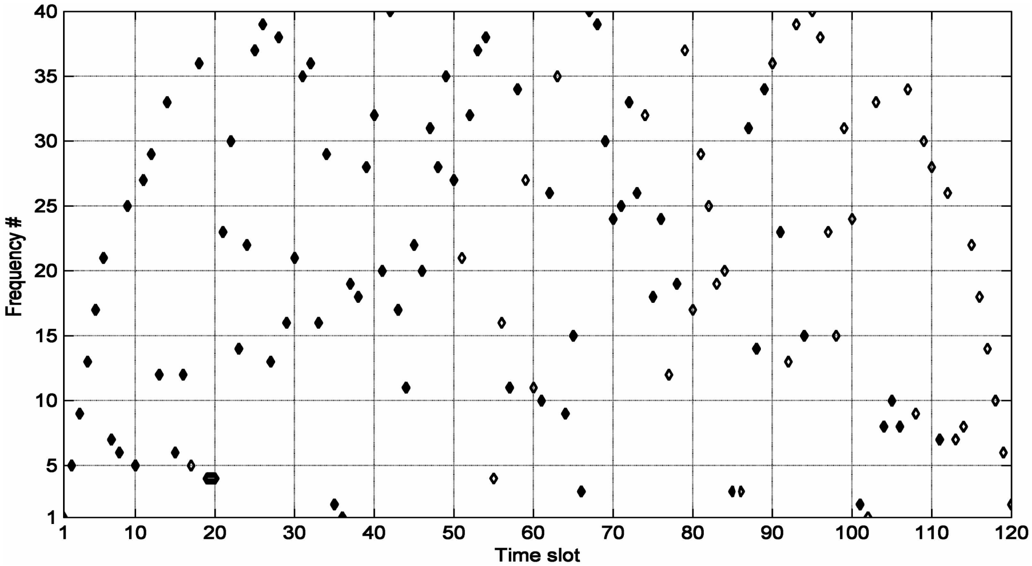


Fig. 6. Ternary array of 40×120 signal.

TABLE V
Relating Frequency Element Polarity to Sidelobe Polarity

b_1	b_2	b_3	c_1	c_2	c_3
x_1	x_2	x_3	x_1x_2	x_2x_3	x_1x_3
x_4	x_5	x_6	x_4x_5	x_5x_6	x_4x_6
x_7	x_8	x_9	x_7x_8	x_8x_9	x_7x_9
x_{10}	x_{11}	x_{12}	$x_{10}x_{11}$	$x_{11}x_{12}$	$x_{10}x_{12}$

indicated by the last row of Table IV. Repeating the same reversal in the second row of Table III, namely of b_3 (= 120), will reverse the polarity of c_2 (= 19) and c_3 (= 85). After reversing b_3 in the 3rd and 4th rows as well, we find out that we caused opposite polarities in one of the two repetitions of each of the four delays (1, 19, 35, and 66). Thus the sidelobes of height 2 will disappear. It is interesting to note that applying the rule of reversing the polarity at the bit locations corresponding to the 3rd column of array **B**, will cause all the remaining autocorrelation sidelobes to be -1. Note that reversing the polarity at bits 55, 85, and 102 will yield the same nullifying result, but the remaining sidelobes could be both -1 or $+1$.

The described method of finding the location of polarity changes may be generalized with the help of Table V. The table refers, for example, to the top four rows of Table III. The variables x_i in the first three columns of Table V can take the values ± 1, representing the polarity of the corresponding time slot. The resulted products in the last three columns will also have values of ± 1, representing the polarity of the corresponding ACF sidelobe. Referring back to the top four rows of array **C** in Table III, in order to nullify the four "level 2" sidelobes at delays 66, 19, 1, and 35, we require the following four corresponding equations to hold

$$\begin{aligned} x_2x_3 + x_4x_5 &= 0 \\ x_5x_6 + x_7x_8 &= 0 \\ x_8x_9 + x_{10}x_{11} &= 0 \\ x_{11}x_{12} + x_1x_2 &= 0. \end{aligned} \qquad x_i \in (+1,-1) \qquad (23)$$

This is a set of 4 equations with 12 variables, for which there are many possible solutions. Few examples are setting to -1 only the following elements: $[x_2, x_8]$ or $[x_5, x_{11}]$ or $[x_3, x_6, x_{11}]$ or $[x_3, x_8, x_{12}]$ or $[x_5, x_9, x_{12}]$ or $[x_2, x_6, x_9]$ or $[x_3, x_6, x_9, x_{12}]$.

The same approach has to be applied to the remaining nine groups of four rows in Table III. An identical solution can be applied to all groups, or to some of the groups, while other solutions are applied to the remaining groups.

III. SIGNAL BASED ON 40×120 TERNARY ARRAY—DETAILS AND PERFORMANCES

An example of a 40×120 ternary array, whose frequency locations follow the array **B** in Table III, is shown graphically in Fig. 6. The phase reversal law is reversing b_3, which results the discrete ACF shown in Fig. 5. The empty diamonds in Fig. 6 represent those elements in which the polarity is reversed.

The details of the corresponding transmitted signal appear in Fig. 7. The three subplots represent (from top) amplitude, normalized frequency, and phase coding. Because only one frequency is transmitted during any bit, the amplitude would normally be a constant. However, in the signal described in Fig. 7, frequency weighting was added, by setting the amplitudes of bits corresponding to a given frequency, according to the weight assigned to that frequency.

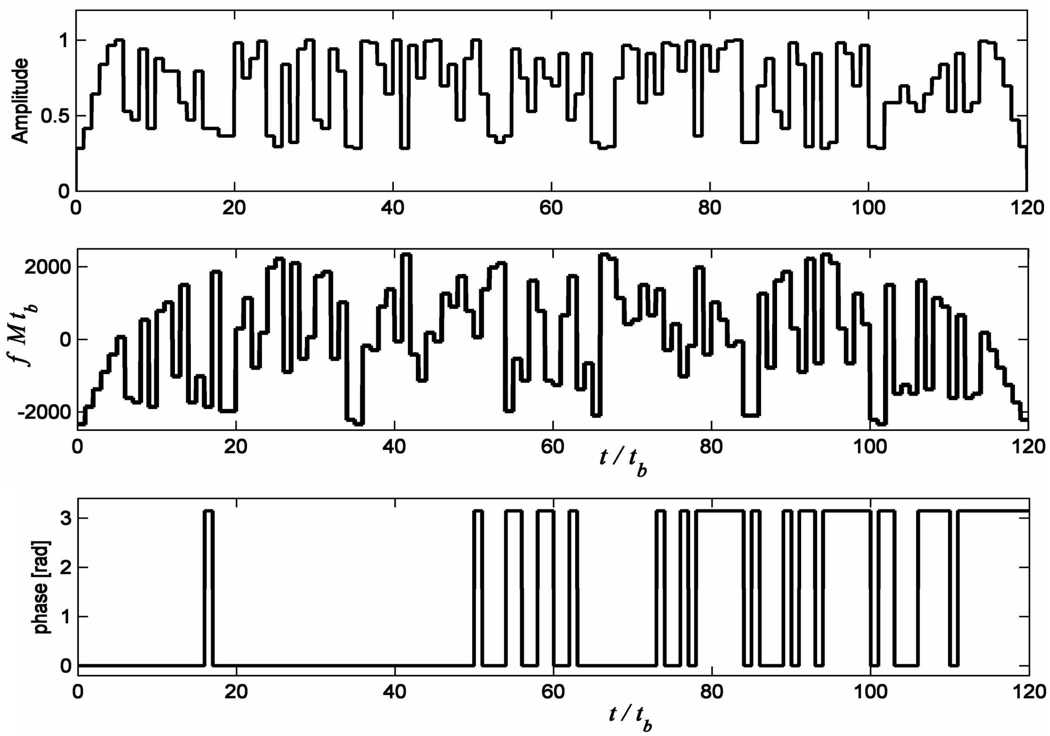


Fig. 7. Amplitude, frequency, and phase coding of 40×120 signal corresponding to ternary array in Fig. 6.

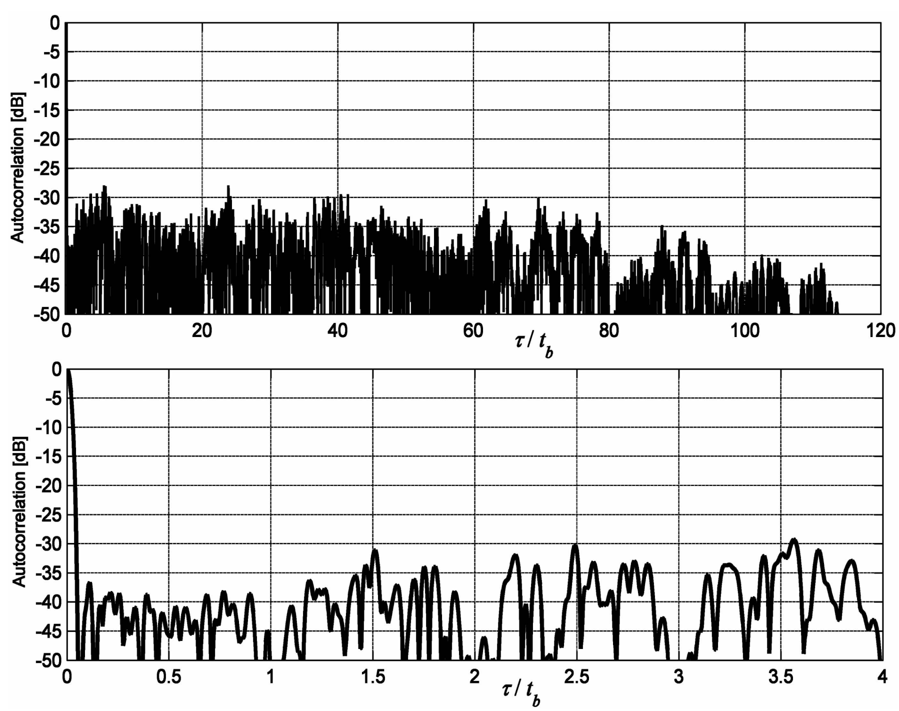


Fig. 8. Top: ACF of signal described in Fig. 7. Bottom: Zoom on first 4 bits.

The weight law was square root of Hamming. Such frequency weighting reduces the ACF sidelobes during the first bit. The resulted ACF of this signal is plotted in Fig. 8. The delay axis of the top ACF plot covers the full length (120 bits). The lower plot zooms on the first 4 bits. Two quadrants of the ambiguity function (AF) (positive Doppler only) are shown in Fig. 9. Because of limits on the density of the mesh, details of the AF in the first bit can be seen only in the bottom part of Fig. 9, which zooms on the first 6 bits.

Considering hardware issues, instead of using square root of the weight window, in both the transmitted and reference signals, it may be preferable

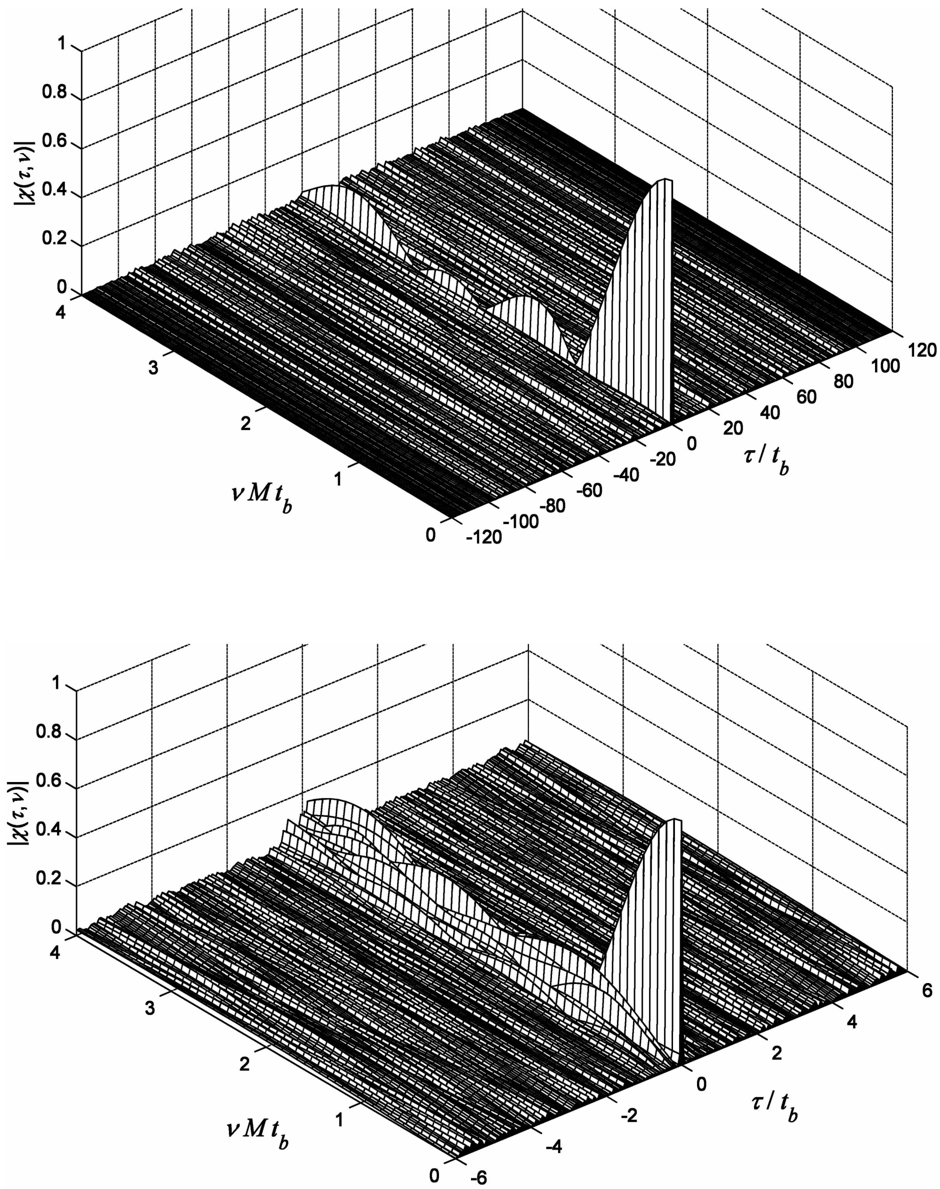


Fig. 9. Top: Ambiguity function of 120 bit signal described in Fig. 7. Bottom: Zoom on first 6 bits.

to transmit a constant amplitude signal and apply the full weight window in the reference signal at the receiver. The delay-Doppler response of such a mismatched receiver will be very similar to the AF of the amplitude weighted signal. The penalty will be a small SNR loss.

Theoretically, such frequency weighting should degrade the nullifying of level 2 sidelobes. A sidelobe of level 2 was created by two different rows (frequencies), and the polarity reversals caused the contribution from these two frequencies to cancel each other. If their amplitudes are different, then the cancellation is not perfect. However, observing the lower part of Fig. 8, we note that the nulls at multiples of t_b are below -50 dB, implying nearly perfect cancellation. This resulted in from the "separability" discussed earlier, which caused any two rows (frequencies) that contribute sidelobes at the same delay, to be within a group of four contiguous frequencies, thus have similar amplitudes. As long as we do not reshuffle the frequencies (or reshuffle but keep each group of four together) the nullifying will be maintained despite the added frequency weighting. Because of the symmetry of the weight function, another frequency shuffling approach, that will minimize the weight effect on the sidelobe nullifying, is to split each group of four rows into two pairs and place them in symmetrical frequency locations. For example, assign rows 1, 2, 3, and 4 of array **B**, to the frequencies 1, 2, 39, and 40, respectively.

IV. CONSTRUCTION OF A FAMILY OF MULTICARRIER BI-PHASE SIGNALS

It is known [7] that the number of different primitive polynomials of degree 5 irreducible above

TABLE VI
Correspondence Between Primitive Polynomials and Signal Arrays

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21	S22
2 1 0 0 0	+													+						+		
2 2 1 0 0				+	+												+					
2 2 0 2 0	+														+						+	
2 0 1 2 0						+					+							+				
2 1 1 2 0										+					+							+
2 0 2 1 0					+									+		+						
2 2 1 1 0						+			+										+			
2 1 0 0 2												+						+	+			
2 0 2 0 2		+	+				+															
2 2 2 0 2																			+		+	+
2 2 0 2 2		+		+				+														
2 1 2 2 2			+								+		+									
2 2 1 2 2							+				+	+										
2 0 0 1 2									+									+				+
2 2 2 1 2															+	+				+		
2 0 1 1 2								+						+			+					
2 0 0 0 1			+			+						+										
2 2 0 0 1					+			+												+		
2 1 1 0 1		+											+				+					
2 2 2 2 1	+									+						+						
2 1 0 1 1									+	+											+	
2 0 2 1 1				+			+						+									

GF(3) is R

$$R = \frac{\varphi(3^5 - 1)}{5} = 22 \qquad (24)$$

where φ is Euler's function. If the primitive polynomial of degree 5 is written as

$$f(x) = x^5 - a_4 x^4 - a_3 x^3 - a_2 x^2 - a_1 x - a_0 \qquad (25)$$

and the companion matrix of $f(x)$ as

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & -a_0 \\ 1 & 0 & 0 & 0 & -a_1 \\ 0 & 1 & 0 & 0 & -a_2 \\ 0 & 0 & 1 & 0 & -a_3 \\ 0 & 0 & 0 & 1 & -a_4 \end{bmatrix} \qquad (26)$$

then the strings in the last column of the matrix H, for all the 22 primitive polynomials, are listed in the first column of Table VI.

There are two methods to construct the family of all 22 ensembles. The first method is through the algorithm implemented in the Matlab program in Appendix A. But it is faster, once that algorithm finds the first ensemble, to calculate the remaining 21 ensembles by using isomprphic multipliers. This approach is based on the use of the formula

$$D_t = tD(\mathrm{mod}\, L + 1) \qquad (27)$$

where t and $L + 1$ are mutually prime, $L = 120$, $T = \{t\}$ is a set of coefficients, and D is a known ENMOC. Clearly, using this second method is possible only if one ENMOC is known. If the resulted D_t is the same as the original D (disregarding a change in the order of elements) then the coefficients t are named automorphic, otherwise—isomorphic.

In order to find $T = \{t\}$ it is necessary to form a multiplicative group in Galois field GF(3^5) with the order 5, which will be named G_5. The multiplicative group is

$$G_5 = (1, 3, 9, 27, 81). \qquad (28)$$

It is easy to check that the results of all multiplicative operations modulo $L + 1$ belong to G_5. The multiplicative group can be divided into adjacent classes h_i, $i = 1, 2, \ldots, 22$ by multiplying the elements of G_5 by the elements of the multiplicative group $G_{120} = (1, 2, \ldots, 120)$, modulo 121. The adjacent classes that contain identical elements are excluded.

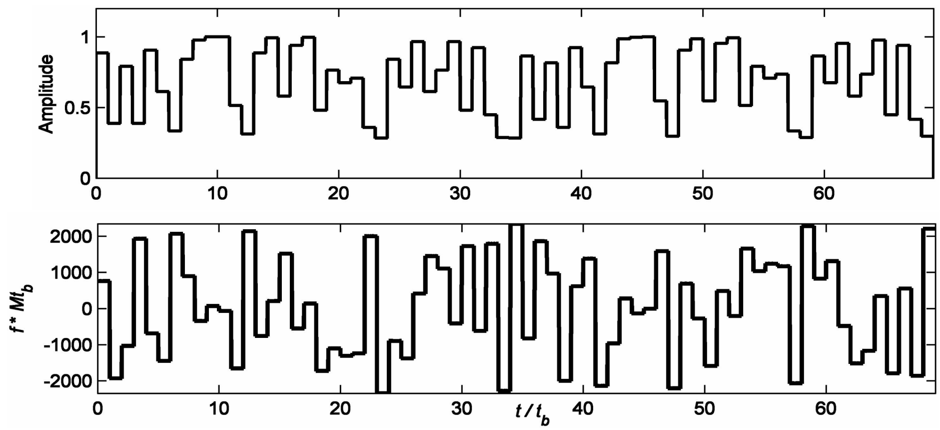


Fig. 10. Amplitude and normalized frequency of signal corresponding to 69×69 binary Costas array with square root of Hamming frequency weighting.

Appendix B presents 22 tables corresponding to 22 arrays of signals. The top rows in the tables contain G_5 (in the first table) and all the adjacent classes (in the remaining tables). The set $T = \{t\}$ of isomorphic coefficients will consist from one integer (any) from every such row. The 21 arrays (S2, S3,…, S22) in Appendix B were found from the ENMOC of S1 (the first column in Table II), using (27). Note that the arrays described by the tables in Appendix B are binary arrays that can describe a multicarrier signal. They still have to be modified into ternary arrays in order to represent a multicarrier bi-phase signal. The simplest modification will be to add a negative sign in front of all the elements in the last (third) column of each table.

It was found that each array fits three polynomials, and each polynomial fits three arrays. The correspondence between arrays and polynomials is outlined in Table VI. The rows represent polynomials and the columns represent arrays.

The 22 tables in Appendix B are listed in a particular way. The first row always starts with 1, and it is followed by the rows obtained through its automorphism. The 5th row starts with 2 (if it did not appear already in an earlier row), and so on. Many variations to each one of the 22 signals can be obtained by reordering the rows (but keeping each group of 4 rows together). Cross correlation between pairs of signals in Appendix B yielded a typical level of -35 dB relative to an autocorrelation peak.

Recall that all the 22 signals in Appendix B posses the important "separability" property. If we are willing to give it up, then it is possible (using random search) to generate many more 40×120 binary arrays with the following properties:

1) There is exactly one 1 in each column.
2) There are exactly three 1s in each row.
3) A specific difference between locations of 1 within a row will not occur more than twice in the entire array.
4) If a specific difference occurs twice, the occurrence will be in different rows.

Arrays obtained by such a random search will usually result in (after the nullifying of sidelobes of magnitude 2) more than 40 remaining sidelobes of magnitude 1.

V. COMPARISON WITH COSTAS SIGNAL

Our 40×120 ternary signal yields a pulse compression factor of 4800. It will be interesting to compare its properties with those of a Costas signal with the same compression factor. The compression of an L element Costas signal is L^2, hence we should compare our 40×120 ternary signal with a 69×69 binary Costas signal. The Costas signal used was picked randomly from the many (> 24) Costas signals known at this length. Its frequency coding is:

46 7 20 63 25 14 65 48 30 36 34 11 66 24 38 57 27

37 10 19 16 17 64 1 22 15 41 56 51 29 60 26 61 2 69

23 62 49 6 44 55 4 21 39 33 35 58 3 45 31 12 42 32

59 50 53 52 5 68 47 54 28 13 18 40 9 43 8 67.

The same frequency weighting function (square root of Hamming) was applied to the Costas signal. The signal's amplitude and frequency modulation are plotted in Fig. 10. The resulted ACF is shown in Fig. 11, and the AF in Fig. 12.

The delay axis of the zoom on the ACF plot (Fig. 11, bottom) extends as far as 2.3 bits out of 69 bits in order to cover the same relative portion of the pulse as in Fig. 8 (bottom), which extends as far as 4 bits out of a signal of length 120. Comparing Figs. 8 and 11 reveals similar mainlobe width. The peak sidelobe level in the Costas case is slightly lower, due probably to the larger number of frequencies. The zoom in Fig. 12 is also proportional to that of Fig. 9 (bottom). In both AF plots the delay axis extends as

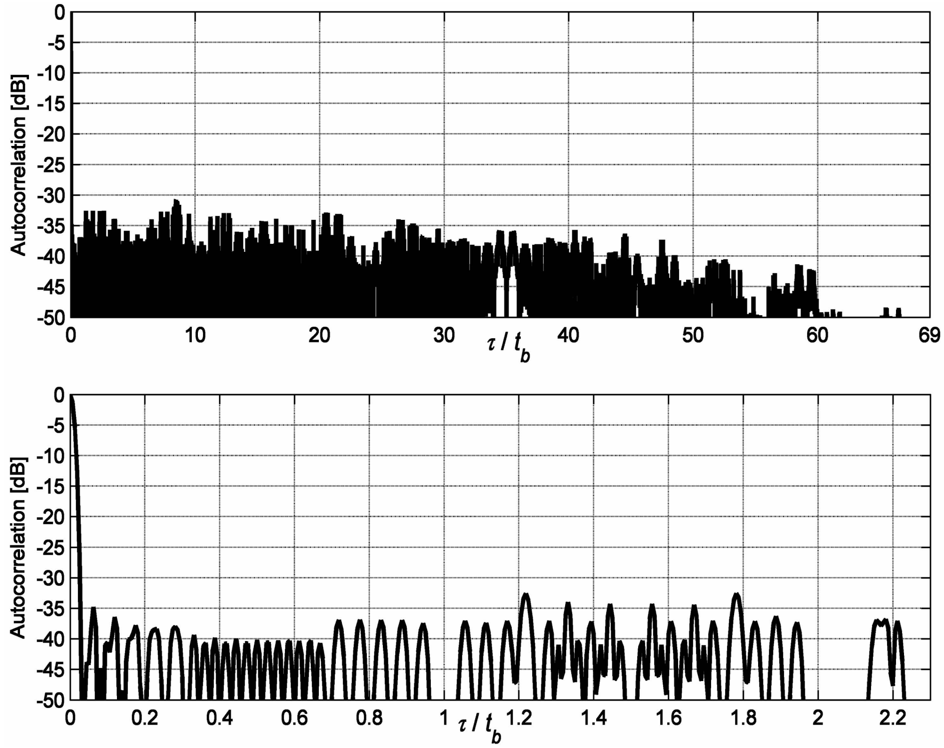


Fig. 11. Top: ACF of 69 bit Costas signal. Bottom: Zoom on first 2.3 bits.

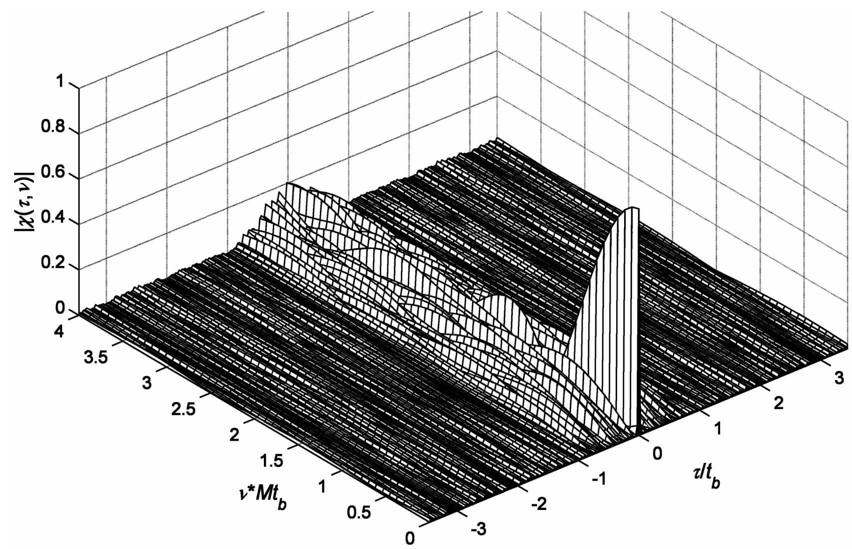


Fig. 12. Ambiguity function of 69 bit Costas signal with zoom on first 3.5 bits.

far as 1/20 of the pulse duration. In general we see similar performances of our 40 frequency ternary signal and the 69 frequency Costas signal.

VI. CONCLUSIONS

A new multicarrier radar signal was described. As in Costas, only one frequency is transmitted during any given time slot. Contrary to Costas, each frequency is repeated several times (3 in the examples given). Another difference is the addition of polarity reversal (180° phase shift) in some of the bits. Thus, while Costas signal can be described by a binary array, our signal has to be described by a ternary array. The given example, of 40 frequencies and 120 bits, yields pulse compression of 4800. To obtain such pulse compression with a Costas signal would have required 69 frequencies. The ordering of the sequences in frequency, and the law of polarity reversals, can have many variations, which calls for further study in order to optimize the AF off the zero-Doppler cut.

APPENDIX A. MATLAB PROGRAM FOR CONSTRUCTING THE ARRAYS

```
% “all_ensemble.m” - Creats all 40x120 Sverdlik/Levanon arrays
clear all
hh_4=[0 0 0 0; 1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1 ]; % first 4 columns
h_last_col=[2 1 0 0 0; 2 2 1 0 0; 2 2 0 2 0; 2 0 1 2 0; 2 1 1 2 0; 2 0 2 1 0; 2 2 1 1 0;...
    2 1 0 0 2; 2 0 2 0 2; 2 2 2 0 2; 2 2 0 2 2; 2 1 2 2 2; 2 2 1 2 2; 2 0 0 1 2;...
    2 2 2 1 2; 2 0 1 1 2; 2 0 0 0 1; 2 2 0 0 1; 2 1 1 0 1; 2 2 2 2 1; 2 1 0 1 1; 2 0 2 1 1]’;
% mp are all the possible multipliers to check linear dependence
mp=[1 1 1 1; 1 1 1 2; 1 1 2 1; 1 1 2 2; 1 2 1 1; 1 2 1 2; 1 2 2 1; 1 2 2 2;...
    2 1 1 1; 2 1 1 2; 2 1 2 1; 2 1 2 2; 2 2 1 1; 2 2 1 2; 2 2 2 1]’;
[ss,tt]=size(mp);
for pol=1:22
    hh(:,:,pol)=[hh_4 h_last_col(:,pol)];
    aa(:,1,pol)=[1 0 0 0 0]’;
    bb(:,1,pol)=[1 0 0 0 0]’;
    for k=1:120
        bb(:,k+1,pol)=hh(:,:,pol)*(:,k,pol);
        aa(:,k+1,pol)=mod(bb(:,k+1,pol),3);
    end
end % of for pol=1:22
% aa are 22 arrays each 5x121. The columns are all the powers of x in the corresponding polynom
pa=1; pb=2; pc=2;
flag_result=0;
results=input(‘How many ensembles to search for (1 to 22)=? ’);
while flag_result<results
    if pc<22
        pc=pc+1;
    else
        pb=pb+1;
        pc=pb+1;
    end
    poly3=[pa pb pc];
    disp(‘ ’)
    disp(‘ Polynomials used ’ ), disp(poly3)
    enssig_temp=[ ];
    enssig=[ ];
    r=[1:120];
    lr=length(r);
    aap=aa(:,:,poly3);
    while lr>11
        m=1+min(r);
        for pol=1:3
            d=1; % the row number
            for n=(m+1):120
                for p=(n+1):121
                    qq(:,:,pol)=[aap(:,1,pol) aap(:,m,pol) aap(:,n,pol) aap(:,p,pol)]’;
                    t=1;
                    while t<(tt+1) % tt is the number of columns of mp (The multiplying options)
                        qqq(:,:,pol)=diag(mp(:,t))*qq(:,:,pol);
                        sq=mod(sum(qqq(:,:,pol)),3);
                        if sq==zeros(1,5) % implying linear dependence
                            t=tt+1; % a multiplier was found. No need to try other multipliers
                            dd(d,:,pol)=sort([0 m-1 n-1 p-1]);
                            d=d+1;
                        else
                            t=t+1;
                        end
                    end % while t
```

```
                end % the loop for p
            end % the loop for n
            ddd(:,:,pol)=dd(:,2:4,pol);
        end % the loop for pol=
        dmax=d;
        for p1=1:dmax-1
            for p2=1:dmax-1
                same_ens(p1,p2)=sum(ddd(p1,:,1)==ddd(p2,:,2));
                if same_ens(p1,p2)==3 % all three elements identical
                    for p3=1:dmax-1
                        samep1p3(p3)=sum(ddd(p1,:,1)==ddd(p3,:,3));
                        if samep1p3(p3)==3
                            enssig_temp=ddd(p1,:,1);
                            % creating the automorphisms
                            for k=1:3
                                stemp(k,:)=mod( [dd(p1,:,1)-dd(p1,k+1,1)], 121);
                                stemp(k,:)=sort(stemp(k,:));
                                enssig_temp=[enssig_temp;stemp(k,2:4)];
                            end % of creating automorphism
                        end
                    end
                end
            end
        end
        if length(enssig_temp)<4
            disp(' No results ')
            enssig=[ ];
            break
        end
        enssig=[enssig;enssig_temp];
        dline=reshape(enssig_temp,1,12);
        enssig_temp=[ ];
        flag=0;
        for q=1:12
            el=find(r==dline(q));
            [s1 s2]=size(el);
            if s2==1 % implying not an empty matrix
                elim(q)=el;
            else
                elim(q)=0;
                flag=1; % implying that at least one element was already used
            end
        end
        if flag==0
            r(elim)=[ ]; % removing the elements used now
            lr=length(r); % updating the length of r
        end % the elements used were taken out of the vector r
        if flag==1
            disp(' No results ')
                break
        end
    end % of while lr>11
    if length(enssig)==40
        disp(' ')
        disp(' Resulted array ')
        disp(' ')
        disp(enssig) % displays the entire signal
        flag_result=flag_result+1;
    end
end % of while flag_result==results-1
```

APPENDIX B. ALL 22 ARRAYS

1, 3, 9, 27, 81		
S1		
1	36	102
35	101	120
66	85	86
19	20	55
2	10	17
8	15	119
7	111	113
104	106	114
3	64	108
61	105	118
44	57	60
13	16	77
4	27	92
23	88	117
65	94	98
29	33	56
5	43	80
38	75	116
37	78	83
41	46	84
6	30	51
24	45	115
21	91	97
70	76	100
9	71	82
62	73	112
11	50	59
39	48	110
12	34	81
22	69	109
47	87	99
40	52	74
14	72	103
58	89	107
31	49	63
18	32	90
25	53	79
28	54	96
26	68	93
42	67	95

20, 47, 56, 59, 60		
S2		
1	53	117
52	116	120
64	68	69
4	5	57
2	49	77
47	75	119
28	72	74
44	46	93
3	38	109
35	106	118
71	83	86
12	15	50
6	26	110
20	104	115
84	95	101
11	17	37
7	16	92
9	85	114
76	105	112
29	36	45
8	30	62
22	54	113
32	91	99
59	67	89
10	43	61
33	51	111
18	78	88
60	70	103
13	27	100
14	87	108
73	94	107
21	34	48
19	42	82
23	63	102
40	79	98
39	58	81
24	65	90
41	66	97
25	56	80
31	55	96

40, 94, 112, 118, 120		
S3		
1	20	86
19	85	120
66	101	102
35	36	55
2	106	113
104	111	119
7	15	17
8	10	114
3	16	60
13	57	118
44	105	108
61	64	77
4	33	98
29	94	117
65	88	92
23	27	56
5	46	83
41	78	116
37	75	80
38	43	84
6	76	97
70	91	115
21	45	51
24	30	100
9	48	59
39	50	112
11	73	82
62	71	110
12	52	99
40	87	109
47	69	81
22	34	74
14	32	63
18	49	107
31	89	103
58	72	90
25	67	93
42	68	96
26	54	79
28	53	95

19, 29, 50, 57, 87		
S4		
1	86	114
85	113	120
28	35	36
7	8	93
2	19	79
17	77	119
60	102	104
42	44	61
3	16	100
13	97	118
84	105	108
21	24	37
4	30	98
26	94	117
68	91	95
23	27	53
5	11	62
6	57	116
51	110	115
59	64	70
9	48	58
39	49	112
10	73	82
63	72	111
12	52	90
40	78	109
38	69	81
31	43	83
14	55	101
41	87	107
46	66	80
20	34	75
15	33	65
18	50	106
32	88	103
56	71	89
22	67	96
45	74	99
29	54	76
25	47	92

25, 70, 75, 89, 104		
S5		
1	70	100
69	99	120
30	51	52
21	22	91
2	15	49
13	47	119
34	106	108
72	74	87
3	58	89
55	86	118
31	63	66
32	35	90
4	64	107
60	103	117
43	57	61
14	18	78
5	41	97
36	92	116
56	80	85
24	29	65
6	26	45
20	39	115
19	95	101
76	82	102
7	88	111
81	104	114
23	33	40
10	17	98
8	50	62
42	54	113
12	71	79
59	67	109
9	25	53
16	44	112
28	96	105
68	77	93
11	48	94
37	83	110
46	73	84
27	38	75

67, 80, 103, 115, 119		
S6		
1	7	33
6	32	120
26	114	115
88	89	95
2	40	51
38	49	119
11	81	83
70	72	110
3	21	99
18	96	118
78	100	103
22	25	43
4	91	105
87	101	117
14	30	34
16	20	107
5	28	64
23	59	116
36	93	98
57	62	85
8	66	75
58	67	113
9	55	63
46	54	112
10	45	92
35	82	111
47	76	86
29	39	74
12	31	73
19	61	109
42	90	102
48	60	79
13	50	65
37	52	108
15	71	84
56	69	106
17	41	94
24	77	104
53	80	97
27	44	68

38, 53, 58, 100, 114		
S7		
1	84	88
83	87	120
4	37	38
33	34	117
2	51	107
49	105	119
56	70	72
14	16	65
3	10	22
7	19	118
12	111	114
99	102	109
5	23	101
18	96	116
78	98	103
20	25	43
6	32	79
26	73	115
47	89	95
42	48	74
8	60	75
52	67	113
15	61	69
46	54	106
9	30	66
21	57	112
36	91	100
55	64	85
11	39	92
28	81	110
53	82	93
29	40	68
13	44	71
31	58	108
27	77	90
50	63	94
17	41	76
24	59	104
35	80	97
45	62	86

10, 28, 30, 84, 90		
S8		
1	85	99
84	98	120
14	36	37
22	23	107
2	63	89
61	87	119
26	58	60
32	34	95
3	13	55
10	52	118
42	108	111
66	69	79
4	15	31
11	27	117
16	106	110
90	94	105
5	82	91
77	86	116
9	39	44
30	35	112
6	25	68
19	62	115
43	96	102
53	59	78
7	54	104
47	97	114
50	67	74
17	24	71
8	46	64
38	56	113
18	75	83
57	65	103
12	45	93
33	81	109
48	76	88
28	40	73
20	49	100
29	80	101
51	72	92
21	41	70

13, 39, 85, 109, 117		
S9		
1	38	97
37	96	120
59	83	84
24	25	62
2	14	66
12	64	119
52	107	109
55	57	69
3	49	114
46	111	118
65	72	75
7	10	56
4	80	102
76	98	117
22	41	45
19	23	99
5	18	110
13	105	116
92	103	108
11	16	29
6	42	77
36	71	115
35	79	85
44	50	86
8	61	89
53	81	113
28	60	68
32	40	93
9	26	100
17	91	112
74	95	104
21	30	47
15	54	88
39	73	106
34	67	82
33	48	87
20	63	90
43	70	101
27	58	78
31	51	94

26, 49, 78, 97, 113		
S10		
1	16	57
15	56	120
41	105	106
64	65	80
2	73	76
71	74	119
3	48	50
45	47	118
4	11	28
7	24	117
17	110	114
93	97	104
5	40	59
35	54	116
19	81	86
62	67	102
6	98	107
92	101	115
9	23	29
14	20	112
8	39	83
31	75	113
44	82	90
38	46	77
10	36	99
26	89	111
63	85	95
22	32	58
12	33	84
21	72	109
51	88	100
37	49	70
13	43	68
30	55	108
25	78	91
53	66	96
18	52	79
34	61	103
27	69	87
42	60	94

16, 23, 48, 69, 86		
S11		
1	37	54
36	53	120
17	84	85
67	68	104
2	9	91
7	89	119
82	112	114
30	32	39
3	41	111
38	108	118
70	80	83
10	13	51
4	94	100
90	96	117
6	27	31
21	25	115
5	57	77
52	72	116
20	64	69
44	49	101
8	73	107
65	99	113
34	48	56
14	22	87
11	26	61
15	50	110
35	95	106
60	71	86
12	40	58
28	46	109
18	81	93
63	75	103
16	59	92
43	76	105
33	62	78
29	45	88
19	42	66
23	47	102
24	79	98
55	74	97

76, 79, 106, 107, 116		
S12		
1	17	30
16	29	120
13	104	105
91	92	108
2	47	55
45	53	119
8	74	76
66	68	113
3	51	90
48	87	118
39	70	73
31	34	82
4	93	102
89	98	117
9	28	32
19	23	112
5	67	100
62	95	116
33	54	59
21	26	88
6	20	44
14	38	115
24	101	107
77	83	97
7	49	110
42	103	114
61	72	79
11	18	60
10	46	81
36	71	111
35	75	85
40	50	86
12	37	64
25	52	109
27	84	96
57	69	94
15	58	80
43	65	106
22	63	78
41	56	99

8, 24, 43, 72, 95		
S13		
1	65	106
64	105	120
41	56	57
15	16	80
2	47	50
45	48	119
3	74	76
71	73	118
4	97	114
93	110	117
17	24	28
7	11	104
5	67	86
62	81	116
19	54	59
35	40	102
6	20	29
14	23	115
9	101	107
92	98	112
8	46	90
38	82	113
44	75	83
31	39	77
10	32	95
22	85	111
63	89	99
26	36	58
12	49	100
37	88	109
51	72	84
21	33	70
13	66	91
53	78	108
25	55	68
30	43	96
18	60	87
42	69	103
27	61	79
34	52	94

2, 6, 18, 41, 54		
S14		
1	89	115
88	114	120
26	32	33
6	7	95
2	72	83
70	81	119
11	49	51
38	40	110
3	25	103
22	100	118
78	96	99
18	21	43
4	20	34
16	30	117
14	101	105
87	91	107
5	62	98
57	93	116
36	59	64
23	28	85
8	54	63
46	55	113
9	67	75
58	66	112
10	39	86
29	76	111
47	82	92
35	45	74
12	60	102
48	90	109
42	61	73
19	31	79
13	69	84
56	71	108
15	52	65
37	50	106
17	44	97
27	80	104
53	77	94
24	41	68

7, 21, 63, 68, 83		
S15		
1	34	38
33	37	120
4	87	88
83	84	117
2	16	72
14	70	119
56	105	107
49	51	65
3	102	114
99	111	118
12	19	22
7	10	109
5	25	103
20	98	116
78	96	101
18	23	43
6	48	95
42	89	115
47	73	79
26	32	74
8	54	69
46	61	113
15	67	75
52	60	106
9	64	100
55	91	112
36	57	66
21	30	85
11	40	93
29	82	110
53	81	92
28	39	68
13	63	90
50	77	108
27	58	71
31	44	94
17	62	97
45	80	104
35	59	76
24	41	86

35, 52, 73, 98, 105		
S16		
1	68	85
67	84	120
17	53	54
36	37	104
2	32	114
30	112	119
82	89	91
7	9	39
3	13	83
10	80	118
70	108	111
38	41	51
4	25	31
21	27	117
6	96	100
90	94	115
5	49	69
44	64	116
20	72	77
52	57	101
8	22	56
14	48	113
34	99	107
65	73	87
11	71	106
60	95	110
35	50	61
15	26	86
12	75	93
63	81	109
18	46	58
28	40	103
16	45	78
29	62	105
33	76	92
43	59	88
19	74	98
55	79	102
24	47	66
23	42	97

4, 12, 36, 82, 108		
S17		
1	25	84
24	83	120
59	96	97
37	38	62
2	57	109
55	107	119
52	64	66
12	14	69
3	10	75
7	72	118
65	111	114
46	49	56
4	23	45
19	41	117
22	98	102
76	80	99
5	16	108
11	103	116
92	105	110
13	18	29
6	50	85
44	79	115
35	71	77
36	42	86
8	40	68
32	60	113
28	81	89
53	61	93
9	30	104
21	95	112
74	91	100
17	26	47
15	48	82
33	67	106
34	73	88
39	54	87
20	51	78
31	58	101
27	70	90
43	63	94

17, 32, 46, 51, 96		
S18		
1	22	52
21	51	120
30	99	100
69	70	91
2	74	108
72	106	119
34	47	49
13	15	87
3	35	66
32	63	118
31	86	89
55	58	90
4	18	61
14	57	117
43	103	107
60	64	78
5	29	85
24	80	116
56	92	97
36	41	65
6	82	101
76	95	115
19	39	45
20	26	102
7	17	40
10	33	114
23	104	111
81	88	98
8	67	79
59	71	113
12	54	62
42	50	109
9	77	105
68	96	112
28	44	53
16	25	93
11	38	84
27	73	110
46	83	94
37	48	75

31, 37, 91, 93, 111		
S19		
1	23	37
22	36	120
14	98	99
84	85	107
2	34	60
32	58	119
26	87	89
61	63	95
3	69	111
66	108	118
42	52	55
10	13	79
4	94	110
90	106	117
16	27	31
11	15	105
5	35	44
30	39	116
9	86	91
77	82	112
6	59	102
53	96	115
43	62	68
19	25	78
7	24	74
17	67	114
50	97	104
47	54	71
8	65	83
57	75	113
18	56	64
38	46	103
12	40	88
28	76	109
48	81	93
33	45	73
20	41	92
21	72	101
51	80	100
29	49	70

5, 14, 15, 42, 45		
S20		
1	92	105
91	104	120
13	29	30
16	17	108
2	68	76
66	74	119
8	53	55
45	47	113
3	34	73
31	70	118
39	87	90
48	51	82
4	23	32
19	28	117
9	98	102
89	93	112
5	26	59
21	54	116
33	95	100
62	67	88
6	83	107
77	101	115
24	38	44
14	20	97
7	18	79
11	72	114
61	103	110
42	49	60
10	50	85
40	75	111
35	71	81
36	46	86
12	69	96
57	84	109
27	52	64
25	37	94
15	56	78
41	63	106
22	65	80
43	58	99

61, 62, 65, 74, 101		
S21		
1	5	69
4	68	120
64	116	117
52	53	57
2	46	74
44	72	119
28	75	77
47	49	93
3	15	86
12	83	118
71	106	109
35	38	50
6	17	101
11	95	115
84	104	110
20	26	37
7	36	112
29	105	114
76	85	92
9	16	45
8	67	99
59	91	113
32	54	62
22	30	89
10	70	88
60	78	111
18	51	61
33	43	103
13	34	107
21	94	108
73	87	100
14	27	48
19	58	98
39	79	102
40	63	82
23	42	81
24	55	80
31	56	97
25	66	90
41	65	96

34, 64, 71, 92, 102		
S22		
1	8	36
7	35	120
28	113	114
85	86	93
2	44	104
42	102	119
60	77	79
17	19	61
3	24	108
21	105	118
84	97	100
13	16	37
4	27	95
23	91	117
68	94	98
26	30	53
5	64	115
59	110	116
51	57	62
6	11	70
9	72	82
63	73	112
10	49	58
39	48	111
12	43	81
31	69	109
38	78	90
40	52	83
14	34	80
20	66	107
46	87	101
41	55	75
15	71	103
56	88	106
32	50	65
18	33	89
22	47	76
25	54	99
29	74	96
45	67	92

REFERENCES

[1] Costas, J. P.
A study of a class of detection waveforms having nearly ideal range-Doppler ambiguity properties.
Proceedings of the IEEE, **72**, 8 (1984), 996–1009.

[2] Le Floch, B., Halbert-Lassalle, R., and Castelain, D.
Digital sound broadcasting to mobile receivers.
IEEE Transactions on Consumer Electronics, **35**, 3 (1989), 493–503.

[3] Levanon, N., and Mozeson, E.
Radar Signals.
New York: Wiley, 2004.

[4] Sverdlik, M. B., and Meleshkevich, A. N.
Synthesis of ensembles of pulse sequences with properties of "not more than one coincidence."
Radio Engineering and Electronic Physics, **21**, 7 (1976), 61–68.

[5] Rao, C. R.
Finite geometries and certain derived results in theory of numbers.
Proceedings of the National Institute of Science, India, **11** (1945), 136–149.

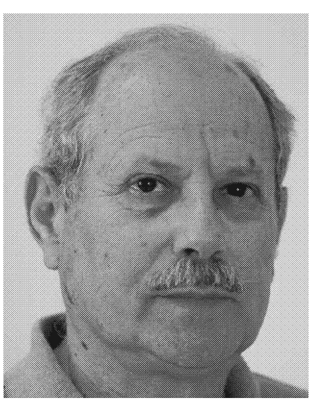
[6] Albert, A. A.
Fundamental Concepts of Higher Algebra.
Chicago: The University of Chicago Press, 1956, ch. IV.

[7] Dickson, L. E.
Linear Groups with an Exposition of the Galois Field Theory.
New York: Dover, 1958, ch. III.

Meshulim B. Sverdlik was born in Rovno, Poland, in 1925. He actively participated in WWII. He received the Dipl. Eng. degree in electronics in 1951 from the Electro-Technical Institute in Odessa, the degree of Candidate of Sciences (which corresponds to the Ph.D. in the West) in 1959, in theoretical communication, and a D.Sc. in radar and navigation engineering in 1974, from the Polytechnic University in Odessa.

From 1952 to 1964 he worked on radar systems at the Novosibirsk Scientific complex. From 1964 to 1994 he was Chairman of the Department of Radio Systems at the Polytechnic University in Odessa.

Dr. Sverdlik has published more than 100 papers (of which about 50 appeared in foreign journals), 15 books, and has 18 patents. He has guided 33 doctoral candidates and 3 Doctor of Science candidates. In 1994 he immigrated to Israel, were he continued his research activity as a free lance.



Nadav Levanon (S'67—M'70—SM'83—F'98—LF'06) received the B.Sc. and M.Sc. degrees from the Technion–Israel Institute of Technology, in 1961 and 1965, and a Ph.D. from the University of Wisconsin–Madison, in 1969, all in electrical engineering.

He has been a faculty member at Tel Aviv University since 1970, where he is a professor in the Department of Electrical Engineering—Systems. He was chairman of that department during 1983–1985. At Tel Aviv University he is also head of the Weinstein Research Institute for Signal Processing and incumbent of the Chair for Radar, Navigation and Electronic Systems. He spent sabbatical years at the University of Wisconsin, The Johns Hopkins University Applied Physics Laboratory, and at Qualcomm Inc., San Diego, CA.

Dr. Levanon is a member of the ION and AGU, and a Fellow of the IET. He is the author of the book *Radar Principles* (Wiley, 1988) and coauthor of *Radar Signals* (Wiley, 2004).