Costas array generator polynomials in finite fields

James K. Beard, *Life Senior Member, IEEE*

***Abstract*—Permutations of order N are generated using polynomials in a Galois field *GF(q)* where *q* ≥ *N+1*, which can be written as a linear transformation on a vector of polynomial coefficients. The Lempel and Golomb methods for generating Costas arrays of order *q*-2 are shown to be very simple examples. The generating polynomial for Costas arrays is examined to form an existence theorem for Costas arrays and a search of polynomial complexity for any given order. Related work is a database on Costas arrays to order 400 and status of an exhaustive search for Costas arrays of order 27.**

I. Introduction

COSTAS arrays are special cases of permutation arrays originally developed as frequency-hop schemes that optimized the usefulness of sonar waveforms, and are now used as components in high-performance radar and communications waveforms [1]. Costas arrays are often characterized as transformations on an identity matrix in which columns are interchanged, resulting in a sparse matrix of ones and zeros. If a column vector consisting of a sequence of integers $\{0,1,2\ldots N-1\}$ is left-multiplied by such a matrix, the resulting vector $\{p_0,p_1,p_2,..p_{N-1}\}$ is the sequence of integers representing this permutation. Each number in the sequence is the column index where the one appears in each row in sequence, so expressing a permutation or Costas array in this format is called *column index notation*.

II. Polynomials Fits in Finite Fields

A. Problem Statement

A permutation of order N where $N \le q-1$ may be characterized as a linear transformation in a finite field or Galois field of order q that is a prime or a power of a prime, denoted as $GF(q)$. The integers p_i are powers of a particular primitive element α, thus representing positions of ones in a row of a permutation matrix, and a sequence of such quantities is generated by a polynomial of order $N-1$ in $GF(q)$,

$$\lambda_0 + \lambda_1 \cdot x^1 + \lambda_2 \cdot x^2 + \ldots + \lambda_{N-1} \cdot x^{N-1} = \phi_x \qquad (2.1)$$

where all quantities in (2.1) are in $GF(q)$. If we take the independent variable x and the sum ϕ_x to be given powers of a primitive element in $GF(q)$, α, we have the form

$$\sum_{i=0}^{N-1} \lambda_i \cdot \alpha^{i\cdot(j-1)} = \alpha^{p_j},\ j \in \{1\ldots N\} \qquad (2.2)$$

where the p_j are an arbitrary set of N integers, all in the range from one to N with no duplications. Matrix notation for this equation is

$$M_{LG} \cdot \underline{l} = \underline{gp} \qquad (2.3)$$

where the matrix M_{LG} and the vectors $\underline{l}$ and $\underline{p}$ are

$$M_{LG} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \ldots & \alpha^{N-1} \\ 1 & \alpha^2 & \alpha^4 & \ldots & \alpha^{2\cdot(N-1)} \\ 1 & \alpha^3 & \alpha^6 & \ldots & \alpha^{3\cdot(N-1)} \\ \vdots & & & & \vdots \\ 1 & \alpha^N & \alpha^{2N} & \ldots & \alpha^{N\cdot(N-1)} \end{bmatrix},\ \underline{l} = \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_{N-1} \end{bmatrix},\ \underline{gp} = \begin{bmatrix} \alpha^{p_0} \\ \alpha^{p_1} \\ \alpha^{p_2} \\ \vdots \\ \alpha^{p_{N-1}} \end{bmatrix}. \qquad (2.4)$$

We see from Eq. (2.3) that any permutation $\underline{p}$ has a one-to-one correspondence with a set of polynomial coefficients $\underline{l}$ if the matrix M_{LG} is nonsingular. The matrix M_{LG} is a Vandermonde matrix [2], and has determinant

$$\left|M_{LG}\right| = \alpha^{\frac{N\cdot(N-1)}{2}} \cdot \prod_{0\le i<j<N} \left(\alpha^j - \alpha^i\right) \qquad (2.5)$$

which is a polynomial valid in $GF(q)$ that is nonzero when α is a primitive element and

$$N \le q-1. \qquad (2.6)$$

B. Computing the Generating Polynomial

The inverse of a Vandermonde matrix can be written as simple equations in the field of real numbers, avoiding or minimizing the usual problem of ill conditioning seen in Vandermonde matrices [2]. However, that result is not directly applicable to finite fields because it involves posing the polynomials with non-integer coefficients, which have little or no meaning in finite fields. Direct methods such as Gauss-Jordan elimination provide efficient computation of inverses of matrices in finite fields. Thus for any permutation $\underline{gp}$ we can easily find a set of polynomial coefficients $\underline{l}$ by

$$\underline{l} = M_{LG}^{-1} \cdot \underline{gp}. \qquad (2.7)$$

Manuscript received February 7, 2008.

James K Beard is self-employed with over 40 years experience in Aerospace Engineering (phone: 609-654-6559; fax: 609-654-8751; e-mail: jkbeard@ieee.org).

C. The Lempel and Golomb Generators

The polynomial $\underline{l}$ has very simple form for permutations that are Costas arrays generated by the Lempel, Golomb, and Welch generators [3]. The simplest permutation is the identity, which we use here as a first example. In this case, the polynomial coefficients are

$$\text{Identity: } \lambda_i \begin{cases} = 1, i = 1 \\ = 0, i \neq 1 \end{cases} \tag{2.8}$$

Costas arrays of order $q-2$ are generated by means of the Lempel generator [3],

$$\text{Lempel: } \alpha^i + \alpha^{p_i} = 1, i \in \{1 \ldots q-2\}. \tag{2.9}$$

For the Lempel generator, the polynomial coefficients are

$$\text{Lempel: } \lambda_i \begin{cases} = 1, \ i = 0 \\ = -1, \ i = 1 \\ = 0, \ i > 1 \end{cases} \tag{2.10}$$

The Golomb generator is given by [3],

$$\text{Golomb: } \alpha^{pg_i} + \beta^i = 1, \ i \in \{1 \ldots q-2\} \tag{2.11}$$

where β is also a primitive element that is distinct from α. The polynomial is given by

$$\text{Golomb: } \lambda_i \begin{cases} = 1, \ i = 0 \\ = -1, \ \alpha^i = \beta \\ = 0 \ \text{ otherwise} \end{cases}, \tag{2.12}$$

i.e. the -1 is in another position.

The Welch generator can be used when q is simply a prime and not a power of a prime, and $GF(pr)$ is implemented as simple integer arithmetic modulo the prime pr. The Welch generator produces Costas arrays of order $pr-1$ and is [3]

$$\text{Welch: } pw_i + 1 = \alpha^{i+r} \bmod pr, \ i \in \{0, \ldots pr-2\} \tag{2.13}$$

where r is an offset integer in the range zero to $pr-1$; Costas arrays that are produced by the Welch generator are periodic in r and are thus singly periodic, but we will use zero offset in our illustrations here.

The polynomial for the Welch generator is that given by Eq. (2.10) with the mapping

$$\text{Welch: } pw_i = \alpha^i. \tag{2.14}$$

These results are surprisingly simple in the light of the fact that each of the polynomial coefficients may be any element in $GF(q)$.

III. The Difference Triangle

A. Definition

Costas arrays are often analyzed using a difference triangle [6], which offers a simple method to verify the Costas condition. The top row is the column indices. Numbering the rows and columns from 0 to $N-1$, row i is the difference between the column index above it and the column index i columns over. The element in row i, column j is

$$d_{i,j} \begin{cases} = p_j, \ i = 0 \\ = p_j - p_{j+i}, \ i > 0. \end{cases} \tag{3.1}$$

An example for an order 5 Costas array is shown below as (3.2).

Costas Array:	3	1	4	0	2
Difference 1:	2	–3	4	–2	
Difference 2:	–1	1	2		
Difference 3:	3	–1			
Difference 4:	1				

(3.2)

B. Conditions for Permutation

The top row is the column indices of a permutation if all its elements are between 0 and $N-1$ inclusive, and none of them are repeated. The condition that there are no zeros in the difference triangle below the top row is equivalent to the condition that no column indices be repeated.

C. The Costas Condition

The Costas condition is that the difference in rows and columns between a pair of ones in the permutation matrix not be repeated for any other pair of ones. This is equivalent to requiring that there can be no duplicate entries in any row of the difference triangle.

D. The Difference Triangle in $GF(q)$

We can pose the difference triangle in terms of elements of a Galois field as follows. Replace the top row with the elements of the vector $\underline{gp}$ as given in (2.4). The elements of the other rows are given as ratios of elements in the top row rather than differences. Thus each element in this new difference triangle is equal to α taken to the power of the corresponding element of the integer difference triangle given in (3.2).

IV. Mapping the Difference Triangle to the Generator Polynomial

A. The Notation and Mapping

Equation (2.7) defines a relationship between polynomial coefficients and a sequence of column indices. We know that $GF(q)$ exists only for q equal to a prime or a power of a prime and we wish to address Costas arrays of any order. In $GF(q)$, there is a zero element or additive identity, a one element or multiplicative identity, and for any element γ

$$\gamma^q = \gamma. \tag{4.1}$$

When γ is a primitive element of $GF(q)$, the sequence of powers 0 to $q-2$ of γ do not repeat and all elements occur in the sequence except the zero element. Thus the highest order of the matrix M that is nonsingular is $q-1$ and we can use this to define a permutation of the same order. We define this equation as

$$M \cdot \underline{l} = \underline{gp} \tag{4.2}$$

where M is

$$M = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \alpha^1 & \alpha^2 & \dots & \alpha^{q-2} \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{2\cdot(q-2)} \\ \vdots & & & & \vdots \\ 1 & \alpha^{q-2} & \alpha^{2\cdot(q-2)} & \dots & \alpha^{(q-2)\cdot(q-2)} \end{bmatrix}. \tag{4.3}$$

We use a row of (4.2) to define a mapping from gp_i as given by a row of the definition in (2.4) to the polynomial vector $\underline{l}$ and write it as

$$gp_i = \alpha^{p_i} = \underline{rm}_i^T \cdot \underline{l} \tag{4.4}$$

where $\underline{rm}_i$ is row i of the matrix M that we defined in (4.3). We now have a difference matrix of elements of $GF(q)$, and its elements are defined from (3.1) by

$$gd_{i,j} \begin{cases} = gp_j = \alpha^{p_j}, \ i = 0 \\ = \dfrac{gp_j}{gp_{j+i}} = \dfrac{\underline{rm}_j^T \cdot \underline{l}}{\underline{rm}_{j+i}^T \cdot \underline{l}}, \ 0 < i < N. \end{cases} \tag{4.5}$$

B. Costas Arrays of Lesser Order

Costas arrays of order $N < q-1$ can be treated by using Vandermonde matrices of order less than $q-1$ as we did in showing that the Lempel and Golomb generators were simple examples; the Costas array will be the first N rows and columns of a $q-1$ by $q-1$ permutation matrix. Here we consider $N < q-1$ with the full $q-1$ by $q-1$ Vandermonde matrix. The surplus elements are free degrees of freedom, and will be the last $q-N-1$ rows and columns, which we will define as an identity matrix to constrain those extra degrees of freedom and uniquely define a $q-1$ by $q-1$ permutation matrix. For these supplementary rows and columns we have

$$\underline{rm}_i^T \cdot \underline{l} = \alpha^i, \ N \le i \le q-2. \tag{4.6}$$

We write (4.6) in matrix form as

$$\begin{bmatrix} 1 & \alpha^N & \alpha^{2N} & \dots & \alpha^{(q-2)\cdot N} \\ 1 & \alpha^{N+1} & \alpha^{2\cdot(N+1)} & \dots & \alpha^{(q-2)\cdot(N+1)} \\ 1 & \alpha^{N+2} & \alpha^{3\cdot\{N+2\}} & \dots & \alpha^{(q-2)\cdot(N+2)} \\ \vdots & & & & \vdots \\ 1 & \alpha^{q-2} & \alpha^{2\cdot(q-2)} & \dots & \alpha^{(q-2)\cdot(q-2)} \end{bmatrix} \cdot \underline{l} = \begin{bmatrix} \alpha^N \\ \alpha^{N+1} \\ \alpha^{N+2} \\ \vdots \\ \alpha^{q-2} \end{bmatrix} \tag{4.7}$$

or, defining notation for (4.7),

$$M_{N,q-2} \cdot \underline{l} = \left[\alpha^i\right]_{N,q-2} \tag{4.8}$$

from which we have a basis constraining the extra degrees of freedom

$$\underline{l}_0 = M^{\#}_{N,q-2} \cdot \left[\alpha^i\right]_{N,q-2} \tag{4.9}$$

where we are using the pseudoinverse for a matrix with more columns than rows,

$$M_{N,q-2}{}^{\#} = M_{N,q-2}{}^T \cdot \left[M_{N,q-2} \cdot M^T_{N,q-2}\right]^{-1}. \tag{4.10}$$

We see that $\underline{l}$ must satisfy

$$M_{N,q-2} \cdot (\underline{l} - \underline{l}_0) = \underline{0}. \tag{4.11}$$

C. Conditions for Permutation

The requirement that the row indices be between zero and $q-1$ is ensured by the effective modulo $q-1$ of the exponent as shown in (4.1). The remaining condition for the sequence to be a permutation is that there are no ones in the new permutation matrix below the top row. This means that

$$\underline{rm}_j^T \cdot \underline{l} \ne \underline{rm}_{j+i}^T \cdot \underline{l}, \ i > 0. \tag{4.12}$$

or,

$$\left(\underline{rm}_j - \underline{rm}_{j+i}\right)^T \cdot \underline{l} \ne 0. \tag{4.13}$$

The number $N_{Permutation}$ of conditions that must be met for no duplications in the sequence of column indices is

$$N_{Permutation} = \binom{N}{2} = \frac{N \cdot (N-1)}{2}. \tag{4.14}$$

Note that (4.13) shows that for any permutation matrix, the vector $\underline{l}$ is not orthogonal to the difference between any two rows of the matrix M as defined by

D. The Costas Condition

The Costas condition is that no two values in any row of the difference matrix be duplicates, or,

$$\frac{\underline{rm}_j^T \cdot \underline{l}}{\underline{rm}_{j+i}^T \cdot \underline{l}} \ne \frac{\underline{rm}_{j+k}^T \cdot \underline{l}}{\underline{rm}_{j+k+i}^T \cdot \underline{l}}, \ i,k > 0 \tag{4.15}$$

where k is the number of columns between the two values of $gd_{i,j}$ that are being compared. Clearing the denominators in (4.15) gives us

$$\left(\underline{rm}_j^T \cdot \underline{l}\right) \cdot \left(\underline{rm}_{j+k+i}^T \cdot \underline{l}\right) \ne \left(\underline{rm}_{j+k}^T \cdot \underline{l}\right) \cdot \left(\underline{rm}_{j+i}^T \cdot \underline{l}\right). \tag{4.16}$$

Moving the terms to the left hand side and factoring the vector $\underline{l}$ from both vectors gives us the quadratic form and inequality

$$\underline{l}^T \cdot \left(\underline{rm}_j \cdot \underline{rm}_{j+k+i}^T - \underline{rm}_{j+k} \cdot \underline{rm}_{j+i}^T\right) \cdot \underline{l} \ne 0. \tag{4.17}$$

The matrix in the quadratic form in (4.17) is the difference between two vector outer products and thus is rank two. For an order N permutation, there will be

$$N_{Costas} = \binom{N}{3} = \frac{N \cdot (N-1) \cdot (N-2)}{6} \tag{4.18}$$

conditions defined by (4.17).

E. The Vector Space of the Generator Polynomial

We define the vector space of $\underline{l}$ from the three conditions that must be met for a generator vector to produce a Costas array:

- The supplementary condition of (4.11) must apply when $N < q-1$. This condition requires that $(\underline{l} - \underline{l}_0)$ be in the null space of the row vectors of $M_{N,q-2}$.
- Equation (4.13) requires that the generator vector

$\underline{l}$ *not* be in the null space of any vector in parenthesis for any allowable set of values of the subscripts.

- The Costas condition of (4.17) is satisfied when the generator vector $\underline{l}$ is *not* in the null space of the matrix of (4.17) for any allowable set of values of the subscripts.

Two of many options to define the allowable space for $\underline{l}$ are the intersection of the allowable vector spaces for these three conditions and as the complement of the union of the prohibited vector spaces. We select the latter option here, because matrices that span unions and complements of vector spaces are simple to construct from matrices that span these spaces. The complement of the vector space of $\underline{l}$ is the union of these vector spaces:

- When $N < q-1$, then (4.11) requires that $\underline{l} - \underline{l}_0$ must *not* be in the space spanned by the rows of $M_{N,q-2}$ as given by (4.7) and (4.8). Call this space S_0.
- The permutation condition (4.13) requires that the generator vector $\underline{l}$ *not* be orthogonal to the vector in parenthesis for any allowable combination of subscripts. Call this space $SP_{j,i}$ and its union SPU.
- The Costas condition of (4.17) requires that the generator vector $\underline{l}$ *not* be in the null space of the matrix in parenthesis in for any allowable combination of subscripts. Call this space $SC_{j,i,k}$ and its union SCU.

A summary of the conditions is that the generator vector $\underline{l}$ must be in the complement of the union of the right null spaces of the vectors in (4.13) and the null spaces of the matrices in (4.17). When $N < q-1$ then we have the additional condition that $\underline{l} - \underline{l}_0$ must be in the right null space of $\left[M_{N,q-2}\right]$, so that the union of the null and the complement of this space to define the allowable space of $\underline{l}$. In equation form, this forbidden space is

$$SF = S_o \bigcup \left(\bigcup_{j,i} SP_{j,i} \right) \bigcup \left(\bigcup_{j,i,k} SC_{j,i,k} \right) \tag{4.19}$$

so that $\underline{l}$ must satisfy

$$\underline{l} \in \backslash SF \,. \tag{4.20}$$

V. A Condition for Existence

A. Definition of the Condition for Existence

A necessary and sufficient condition for existence of Costas arrays of order N is that the space given in (4.20) have nonzero rank. This is equivalent to the condition that a matrix of basis vectors spanning SF must not be full rank. Note that a necessary condition is that the strongest condition SCU not span the space, and we show below that this condition alone is useful in constructing Costas arrays.

B. Using the Condition for Existence

The usual tool for finding, constructing, and complementing vector spaces spanned by available matrices is singular value decomposition (SVD). SVD is not available in $GF(q)$ because square roots, magnitudes, non-integral multiples, and other operations not meaningful in $GF(q)$ are needed to use classical methods to implement a SVD. However, Gram-Schmidt orthogonalization, complementary space constructions and other classical linear algebra concepts are relevant and defined in $GF(q)$ that suffice for our purposes.

C. Issues with Finite Fields in Linear Algebra

A property of linear algebra of finite fields is that some vectors are self-annihilating; that is, the dot products of some vectors with themselves are the zero element, even though the vectors are not zero. As an example, consider a vector whose elements are powers of a given element γ:

$$\underline{v}_{SA} = \begin{bmatrix} \gamma^0 \\ \gamma^1 \\ \vdots \\ \gamma^{q-2} \end{bmatrix}. \tag{5.1}$$

We see that the dot product of this vector with itself is

$$\underline{v}_{SA}^T \cdot \underline{v}_{SA} = \sum_{i=0}^{q-2} \gamma^{2i} = \frac{1-\gamma^{2\cdot(q-1)}}{1-\gamma^2} \begin{cases} = 0, & \gamma^2 \neq 1 \\ = (q-1)\cdot\gamma, & \gamma^2 = 1. \end{cases} \tag{5.2}$$

Here we define an integer multiplied by an element of $GF(q)$ as the element added to itself a number of times equal to the integer,

$$k \cdot \gamma \equiv \sum_{i=1}^{k} \gamma \,. \tag{5.3}$$

A property that accrues to this definition is

$$p \cdot \gamma = 0 \text{ in } GF\left(p^k\right). \tag{5.4}$$

Some consequences of (5.4) are that the second row on the right hand side of (5.2) is never zero, and every element of $GF\left(2^k\right)$ is its own negative.

Since the order of the powers of γ in the summation of (5.2) is irrelevant to the value of the sum in (5.2), we see that all vectors $\underline{gp}$ corresponding to permutations are self-annihilating. Also, the general form of (5.1) and (5.2) shows us that all but two of the rows (and columns) of M as given by (4.3) are self-annihilating.

VI. Finding Costas Arrays

A. Using the Costas Conditions

The strongest condition is the Costas condition (4.17), because it constrains the rank of the vector space of $\underline{l}$ to a

maximum of. This is because the null space of $\underline{l}$ is $N-2$ for each combination of subscripts in (4.17), so that the maximum allowable rank R_{CG} of the vector space of $\underline{l}$ is two, once (4.11) has been incorporated. Call the vectors spanning this space

$$\underline{cg}_i,\ 0 \le i < R_{CG}. \tag{6.1}$$

Then, any set of elements γ_i in $GF(q)$ will generate a candidate Costas array generator vector $\underline{l}$ by

$$\underline{l} = \sum_{i=0}^{R_{CG}-1} \gamma_i \cdot \underline{cg}_i = CG \cdot \underline{\gamma} \tag{6.2}$$

where the matrix CG is the concatenation of the vectors $\underline{cg}_i$ as its columns. Since all values in $GF(q)$ are allowed in $\underline{\gamma}$ which has R_{CG} elements, then N_{CG} distinct values of $\underline{\gamma}$ are possible, where

$$N_{CG} = q^{R_{CG}}. \tag{6.3}$$

We do not add the permutation conditions so each resulting sequence is then checked for validity as a Costas array. Clearly this search over polynomial space is preferable to searching over all possible values of the generator vector. When $N < q-1$ but $q-N-1$ is small, the rank of the candidate space can be as large as $q-N+1$ but even this can be a practical solution when this rank remains small.

Equation (6.3) puts an upper limit on the number of Costas arrays that may exist of order N. Non-Costas arrays and even non-permutations and violations of (4.11) will, in general, occur so the actual count of separate Costas arrays of order N will, in general, be smaller than N_{CG} as given by (6.3). Also, Costas arrays come in sets of four or eight that are defined using transposition and rotation of any one permutation matrix, and (6.2) will generate all of them.

VII. Results and Conclusions

A. Some Results in $GF(27)$

All 56 Costas arrays of order 26 were analyzed using (4.2) to find the polynomial generating vector $\underline{l}$ for each one. The polynomial coefficients λ_i are elements in $GF(27)$, and were converted to integers by finding $\log_\alpha(\lambda_i)$ for output, with -99 as a special case for the zero element. Program output for these selected generator polynomials are shown in Figure 2. Visible patterns are seen for Costas arrays generated by the Taylor 1 method, which is Lempel-Golomb with a corner dot added, and Inhomogeneous Additive 1, which is a modification of the Lempel-Golomb method first reported in [4]. Note that all the vectors $\underline{l}$ have a zero first element as expected, and that the vectors given, when a row of M or its inverse is subtracted, yield a sparse vector.

Of these 56 Costas arrays, 40 are found by the generators reported in [4] and [5] and 16 are two sets of eight polymormphs first reported in [4]. One of each set was found to be orthogonal to one in the other set; the generator polynomials for these are given at the end of Figure 2.

This result supports out (6.2), in that all of them are seen to be the sums of a few simple vectors that can be derived from rows of M or its inverse.

B. Results from Other Costas Array Work

1) Extension of Exhaustive List

Earlier work reported on in CISS 2006 [5] provided an exhaustive list of 663,702 Costas arrays of orders 2 through 200; this list is available as a CD-ROM from the author.

Databases of Costas arrays to orders 300 and 400 are now available from the author. The cumulative number of Costas arrays versus order is shown below as Figure 1. The behavior of the curve for orders below 100 is dominated by the large numbers of Costas arrays below order 20 but a straight line on a log-log plot begins to emerge at higher orders. A curve fit over the range 200 to 400 yields this fit:

$$N_{CA}(Order) \sim 0.9214 \cdot \mathrm{Order}^{2.5253} \tag{7.1}$$

The database of orders 2 through 200 has 633,702 Costas arrays. Extending to order 300 produces a database of 1,640,542 Costas arrays, and extending to order 400 produces a database of 3,609,550 Costas arrays. The database for orders to 400 is 7,438,337,014 bytes, which is too large for any convenient media other than 8 GB or larger portable flash memory or hard drives, or a double-sided DVD. It is available as a 139 MB compressed archive on http://jameskbeard.com.

The data extraction routine is all new, with a simple interactive one-screen user interface, with a few advanced options available as separate screens.

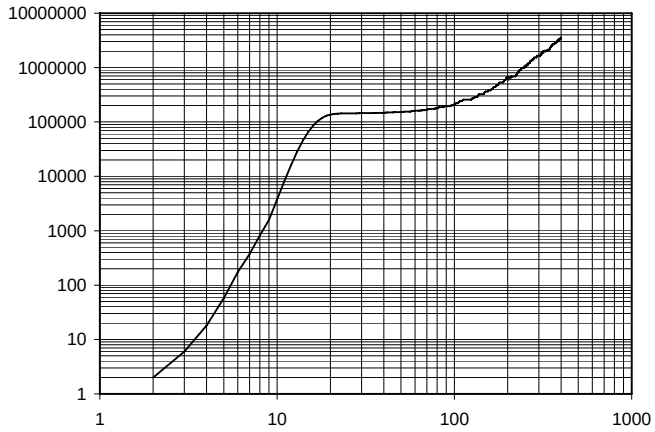

Figure 1. Cumulative Number of Costas Arrays vs. Order

C. Work on Exhaustive Search over Order 27

The present author, with a team of four others, has performed exhaustive searches for Costas arrays of orders 24, 25, and 26 [4], and are performing an exhaustive search over order 27. We are just under 50% done as of this writing. Our sampling scheme in the column index notation space indicates that there are no likely spurious Costas arrays of order 27, and our tentative estimate of the total number of Costas arrays of

order 27 is 196, the number produced by the generator program reported on in CISS 2006 [5].

D. Conclusions

The theory of linear algebra over finite fields provides a link to the classical Costas array generators by Welch, Lempel, and Golomb and a basis for extending the theory of Costas arrays to existence theorems and other new principles. One major result is a method for showing whether or not Costas arrays exist for a given order. Entirely new focused searches are provided, based on linear algebra in finite fields, which are of polynomial complexity.

REFERENCES

[1] Nadav Levanon and Eli Mozenson, "Orthogonal train of modified Costas pulses," *Proceedings of the IEEE 2004 Radar Conference*, April 26-29 2004, ISBN 0-7803-8234-X., pp. 255-259.

[2] Richard Bellman, *Introduction to Matrix Analysis*, Second Edition, SIAM Classics Series, ISBN 0-89871-399-4, p. 193 and 359-360.

[3] Solomon W. Golomb and Herbert Taylor, "Constructions and Properties of Costas Arrays," *Proceedings of the IEEE* **72**, 9 (September 1984) 1143-2263.

[4] J. K. Beard, J. C. Russo, K. Erickson, M. Monteleone, and M. Wright, Costas Array Generation and Search Methodology, *IEEE Transactions on Aerospace and Electronic Systems*, **43**, 2 (April 2007), 522-538.

[5] J. K. Beard, Generating Costas Arrays to Order 200, *Conference on Information Sciences and Systems (CISS) 2006*, March 23, 2006, paper number 178, Princeton University.

[6] S. W. Golomb and G. Gong, The Status of Costas Arrays, *IEEE Trans. On Information Theory*, **53**, 11 (November 2007), 4260-4265.

James K Beard (M'64–LM'04–LSM'05) became a Member (M) of IEEE in 1964, Life Member (LM) in 2004, and Life Senior Member in 2005. He was born in Austin, TX in 1939. He received a BS degree from the University of Texas at Austin in 1962, an MS from the University of Pittsburgh in 1963, and the Ph. D. from the University of Texas at Austin in 1968, all in electrical engineering.

Between 1959 and 2004, he worked in Government laboratories, industry, and as an individual consultant. He is the author of a number of symposia papers including paper number 178 at CISS 2006 [5] and a book, "The FFT in the 21st Century" (Kluwer, 2003).

Dr. Beard is a member of IEEE and other professional societies. He is a member of Phi Eta Sigma, Eta Kappa Nu, Tau Beta Pi, and Sigma Xi. He studied for his Ph. D. under a GSRF Fellowship (matched U. Texas Austin and Ford Foundation funding, administered by U. Texas Austin) and a NSF Fellowship.

Costas array methods: Taylor 1, Cols Reversed, Transposed
-99 13 0 13 13 13 13 -99 13 0 13 13 13 13 -99 13 13 13 13 13 13 0 13 13 13 13
Costas array methods: Taylor 1
-99 14 15 16 -99 18 19 20 8 22 23 24 25 0 1 -99 3 4 5 19 7 8 9 23 11 12
Costas array methods: Inhom. A 1, Cols Reversed
-99 3 18 -99 9 24 -99 15 4 -99 21 10 -99 1 16 -99 7 22 -99 13 2 -99 19 8 -99 25
Costas array methods: Taylor 1, Rows Reversed, Cols Reversed, Transposed
-99 14 15 16 17 5 19 20 21 22 23 24 -99 0 1 2 3 17 5 -99 7 8 9 10 24 12
Costas array methods: Inhom. A 1, Rows Reversed, Cols Reversed
-99 0 -99 11 23 -99 8 20 -99 5 17 -99 2 14 -99 25 11 -99 22 8 -99 19 5 -99 16 2
Costas array methods: Inhom. A 1, Rows Reversed
-99 10 9 8 7 6 5 4 3 2 1 0 25 24 23 22 21 20 19 18 17 16 15 14 13 25
Costas array methods: Taylor 1, Rows Reversed, Transposed
-99 13 14 15 -99 4 18 19 20 8 22 23 24 25 0 -99 2 3 4 5 19 7 8 9 10 11
Costas array methods: Inhom. A 1
-99 0 15 17 19 21 23 25 1 3 5 7 9 11 13 15 17 19 21 23 25 1 3 5 7 9
Costas array methods: Taylor 1, Cols Reversed
-99 13 14 2 16 -99 18 19 20 21 -99 23 24 25 0 1 2 3 4 18 6 7 8 9 23 11
Costas array methods: Taylor 1, Transposed
-99 12 12 12 12 12 25 12 12 12 12 -99 12 12 12 12 12 25 12 12 12 25 -99 12 12 12
Costas array methods: Taylor 1, Rows Reversed, Cols Reversed
-99 12 25 12 12 12 12 25 12 12 12 12 12 12 12 12 -99 12 12 12 12 -99 12 25 12 12
ORTHOGONAL PAIR
Costas array polynomial coefficients, 2 zeros
-99 24 2 7 24 16 5 19 10 13 22 -99 3 9 11 3 5 23 7 15 6 8 18 24 15 9
Costas array polynomial coefficients, 3 zeros
-99 -99 6 10 15 14 23 0 17 9 25 16 0 5 9 4 9 8 20 6 7 11 15 22 -99 3
ORTHOGONAL PAIR
Costas array polynomial coefficients, 2 zeros
-99 10 17 1 22 13 12 22 15 6 15 14 23 22 17 -99 12 4 2 12 25 11 20 4 0 23
Costas array polynomial coefficients, 3 zeros
-99 4 -99 25 19 16 13 13 2 17 19 15 21 18 14 5 15 0 9 19 17 9 11 7 4 -99

Figure 2. Polynomial Coefficients for Selected Costas Arrays Shown as $\log_{\alpha}(\lambda_i)$