Singular Value Decomposition of a Matrix Representation of the Costas Condition for Costas Array Selection

James K Beard, *Life Senior Member, IEEE*

***Abstract*—Costas arrays are permutation matrices that meet the additional Costas condition. The Costas condition requires that when a Costas array defines the frequencies in a frequency-jump-burst and a signal is processed with a matched filter, then for any time or Doppler offset other than both zero, only one pulse will be at peak response at a time.**

Specific formulations of the Costas conditions provide a vector-matrix arithmetic representation where the matrix has eigenvalues that are square roots of integers and eigenvectors that can be scaled to have elements that are all integers. Polynomials are given for squared eigenvalues and for elements of scaled right eigenvector elements. A database of these SVDs is provided on IEEE DataPort for orders from 3 to 1030, with left eigenvectors to order 100. We include proofs of validity and configuration.

The right eigenvectors and the eigenvalues map a Costas array vector into its eigenspace. Functions of these vectors can be associated with system performance metrics so that these mapped vectors may be used to select and rank Costas arrays, and is suitable for real-time system implementation.

***Index Terms*—Costas arrays, Costas condition, singular value decomposition, SVD.**

I. Introduction

COSTAS arrays [1], [2], [3], [4], [5], [6] are used as frequency hop schemes in many radar and communications systems [7], [8], [9] and also in digital codes[10].

Costas arrays are often analyzed using a construct called the *difference triangle*, which is computed from differences between the row indices of pairs of ones in the Costas array matrix. The Costas condition is equivalent to conditions on the difference triangle. These conditions can be posed as a vector-matrix equation [11].

Specific formulations of the vector-matrix Costas condition provides a matrix that, under singular value decomposition (SVD), reveals eigenvalues that are the square roots of integers and eigenvectors that can be scaled so that all elements are integers, all of which are given here by simple polynomial equations. These simple polynomial equations were used to provide a database of SVDs to order 1030 on IEEE Dataport[12].

Dr. Beard, ORCID 0000-0003-4430-8704, is an independent consultant located in the Philadelphia, PA area. USA e-mail: jkbeard@ieee.org

This work is unfunded research by the author.

Manuscript received May 21, 2020; revised August 26, 2020; released for publicaiton October 28, 2020.

Refereeing of this contributin was handled by S. D. Blunt.

Digital Object Identifier: 10.1109/TAES.2020.3040057

TABLE I
Costas Array Matrix Example

	Col 1	Col 2	Col 3	Col 4
Costas array	3	1	2	4
Row 1		●		
Row 2			●	
Row 3	●			
Row 4				●

TABLE II
Costas Array and Difference Triangle Example

Costas array	3	1	2	4
Row 1	-2	1	2	
Row 2	-1	3		
Row 3	1			

The right eigenvector matrix offers an orthogonal transformation to an eigenspace that may be useful in real-time selection of Costas arrays in electronic systems.

II. Equivalent Forms of the Costas Condition

Here we characterize Costas arrays as square matrices of n rows and columns, which are permutation matrices (i.e. which have only a single one in each row and column, and all the other elements are zero) which also satisfy the *Costas condition*. The *order* of the Costas array is the number n.

The definitions and conventions of this Section are laid out in [2] and contemporary related works. They are presented and shown to be equivalent in Theorem 2.1 for the purpose of proving that this, and any other work, applies equally to applying the Costas condition through the *difference triangle*, *difference vectors*, or the *discrete ambiguity function* (DAF); and, in general, development of a conclusion from any one of them is mathematically equivalent to development of the same conclusion from any of the others.

A. Notation and Mathematical Foundation

Here we will define the terms most often used in stating the Costas condition and prove their equivalence. Please refer to Tables I, II, III and IV as a simple example in the following narratives.

TABLE III
COSTAS ARRAY AND DIFFERENCE VECTORS EXAMPLE

Costas array	3	1	2	4
Row 1		(+2,-1)	(+1,-2)	(-1,-3)
Row 2	(-2,+1)		(-1,-1)	(-3,-2)
Row 3	(-1,+2)	(+1,+1)		(-2,-1)
Row 4	(+1,+3)	(+3,+2)	(+2,+1)	

TABLE IV
DISCRETE AMBIGUITY FUNCTION (DAF) FOR THE EXAMPLE

Columns	-3	-2	-1	0	+1	+2	+3
Row -3		●					
Row -2			●		●		
Row -1	●		●			●	
Row 0				❹			
Row 1		●			●		●
Row 2			●		●		
Row 3						●	

The *Costas array vector* is a vector $\vec{c}$ whose elements are, for each column of the Costas array matrix, the row indices of the ones in the Costas array. The elements of the Costas array vector are a permutation of a sequence of integers from 1 to n. The first rows of Tables I through III are Costas array vector for the order 4 Costas array used in the example.

The *Costas array matrix* shown in Table I corresponds to the Costas array vector shown as the first row of the table. Rows are numbered downward in Table I and Table IV. If you number the rows positive upward to have that convention in the DAF, one or both the Costas array matrix and the DAF will be flipped vertically from the depictions in Tables I and IV, or the equations tying them together will develop sign changes that complicate comparisons and relationships.

Difference vectors are the vectors, in the Costas array matrix, from any 1 to or from another 1. In Table III, the difference vectors from the one in each column are given as a table; the corresponding Costas array vector is given as the first row of the table. The distance vectors have a two-to-one correspondence to the difference triangle entries, and are given by looking at the Costas array matrix and writing the row j to row $j+i$ distances,

$$\vec{v}_D \begin{cases} = (j, d(i,j)), & j > 0, i+j \le n \\ = -(-j, -d(i,j)), & j < 0, i-j \le n. \end{cases} \quad (1)$$

Since there are $n-1$ rows to the difference triangle, there are $n \cdot (n-1)/2$ elements in the difference triangle, so, by (1), there are $n \cdot (n-1)$ difference vectors.

The *difference triangle* is a triangular matrix of n rows, numbered from 1 to $n-1$, that is constructed from the Costas array vector as follows:

- An auxiliary row which we will number here as 0 is the Costas array vector $\vec{c}$, with n elements, and may not be considered an actual part of the difference triangle,
- Row i, for $1 \le i \le n-1$, each of which has $(n-i)$ elements, where
- Each element $d(i,j)$ at position (i,j), $i > 0$ is computed from the Costas array vector as $c(i+j) - c(j)$.

We formalize the definition of the elements of the difference triangle with an equation that we will reference in mathematical developments to follow:

$$d(i,j) = c(i+j) - c(j); 1 \le i \le n-1, 1 \le j \le n-i. \quad (2)$$

A permutation matrix has a single 1 in each row and each column, and every Costas array matrix is a permutation matrix. In a permutation matrix, no two ones are on the same row, so that no two elements of the Costas array vector are equal, and no element $d(i,j)$ of the difference triangle is zero. Equation (2) is illustrated by Table II.

The *discrete ambiguity function* (DAF) is produced as follows. We use the concept of a FJB waveform passign thogugh a matched filter. This is visualized or demonstrated by drawing a Costas array matrix with dots for ones and blanks for zeros that represents the matched filter, then overlaying it with a duplicate that represents the signal which one moves a fixed number of rows, representing Doppler shift, and columns, representing time offset from centering in the matched filter. When the number of overlaid dots is tabulated versus position, the DAF as shown in Table IV is produced. Note that the center dot has a ”4” that represents that four dots overlay, not just one as with all the other dots. The Costas condition is that not more than two ones, or dots, will overlay except for zero offset in both time and frequency, in which case all of the ones will overlay.

The DAF is relevant to the ambiguity function as a quick guide to an actual ambiguity function; the ambiguity function of a waveform using the Costas array as a frequency hop scheme is the coherent sum of ambiguity functions of the individual tones or “chips”, placed in time-frequency space at the positions of the ones in the DAF, with a central chip ambiguity function weighted by a factor of n.

The DAF is a square matrix $2n-1$ elements to a side. The value of the center element is n. The ones in the DAF are at positions displaced from $(0,0)$ by the distance vectors, i.e. at $\vec{v}_D(i,j)$ and $\vec{v}_D(-i,-d(-i,j))$ for $i < 0$. Thus, the DAF is 2-fold rotationally symmetric about the central point; it is unchanged by rotating $180°$. Note that, as can be seen by (1), there are no $1s$ in row 0 or column 0 of the DAF except at the center $(0,0)$ because j is never zero in (2) and no difference triangle entry $d(i,j)$ is ever zero. Also, since the DAF has ones at positions relative to its center that are the distance vectors of the Costas array matrix, then when we rotate or transpose a Costas array matrix, the DAF will undergo the same transformation.

Details of the definition and trade space for a simple frequency hop waveform are given in [7], [9].

B. The Costas Condition

We state the following theorem which states that the Costas condition is exactly the same when stated in terms of the difference triangle, the difference vectors, or the DAF.

Theorem 2.1 (Costas Condition Equivalences): The Costas condition is met by a permutation matrix when any one of the following conditions is met, and if one of them is true for a given matrix then all of them are true:

- The DAF for a permutation matrix consists of all ones and zeros except the center element, which is n.
- No difference vector in a permutation matrix is repeated.
- On any row of the difference triangle of a permutation matrix, no number is repeated.

Proof: The first item holds because the definition of the DAF and the requirement that any element of the DAF other than the center element is either zero or one; this is the definition of the Costas condition that derives from its use as a frequency-hopping scheme for a sonar waveform[13].

The second item follows from the first item because difference vectors between ones on the Costas array matrix are equal to vectors in the DAF from the center to the ones, and if any two were equal then a value on the DAF other than the center element would be greater than one.

The third item follows from the first and second items, and (1), because no two difference vectors may be equal.

The first and second items follow from the third by (1) and (2). The first item follows from the second item because vectors from the center of the DAF have a one-to-one correspondence with the difference vectors, which also shows that the first item follows from the second item. ■

Here we note the conditions applied by the third item of Theorem 2.1 as a theorem for use in the next Section.

III. Matrix Representation of the Costas Condition

We begin by simply establishing a notation for a matrix formulation of the Costas condition as left-multiplying a Costas array vector, a column (not row) vector $\vec{c}$ to produce another column vector $\vec{b}$. The Costas condition is that all elements of $\vec{b}$ are nonzero.

We then explain the construction of the matrix A. The exact formulation of the rows of the matrix A determines the critical results in that all of the elements of the singular value decomposition of A can be produced as integers, so that a database supports use to any precision the user chooses to implement.

We conclude this Section with a discussion of some of the properties of the matrix A, specifically the number of rows, which increases as the cube of the order of Costas arrays. The rank is proven to be $n-1$ in Theorem 3.2 in Section III-C2.

A. Formulation of the Matrix A

The Costas condition can be expressed as a vector-matrix equation,

$$A \cdot \vec{c} = \vec{b} \tag{3}$$

where $\vec{c}$ is a Costas array vector as a column vector, inner products of the rows of the matrix A and the Costas array vector $\vec{c}$ express individual elements of the Costas condition. Elements of the matrix A are all zero or ± 1 or $+2$. The vector $\vec{b}$ is an inhomogeneity vector of integers, none of which is zero; the magnitudes of these elements does not exceed $(2n-3)$.

B. Formulation of the Rows of A

In our formulation of the matrix A in (3), we have found that if we augment the Costas condition rows, and the new rows express the conditions that none of the elements of the difference triangle may be zero (as required for any permutation matrix), the eigenvalues found by an SVD of the augmented matrix A are all square roots of integers. In the augmented portion of A expressing the condition that no element of the difference triangle is zero, each row has one $+1$, one -1 and the rest of that row is all zeros. The remaining rows compute elements of $\vec{b}$ as differences between elements of each row vector.

We define each row of the matrix A as follows. We will use the difference triangle as shown in Table II here; you may choose to use the difference vectors of Table III or the positions of the ones in the DAF as shown in table IV, or directly from the Costas array vector.

1) Permutation Condition Rows: The difference triangle, row i, has $(n-i)$ elements, each of which is the difference between two row indices from the Costas array vector. Our convention as shown in Table II is, for the element of the difference triangle at row i and column j, repeating (2),

$$d(i,j) = c(j+i) - c(j).$$

To produce a value of $d(i,j)$ as an element of $\vec{b}$ as a dot product of that row with the Costas array vector $\vec{c}$, a row $a(ai,:)$ of A is defined as

$$A(ai,aj) \begin{cases} = -1, \ aj = j \\ = +1, \ aj = j+i \\ = 0 \text{ Otherwise.} \end{cases} \tag{4}$$

Here, the row index ai of A varies over $(n-i)$ elements of row i of the difference triangle, each of which corresponds to a row of A and an element of $\vec{b}$. Thus there are $n \cdot (n-1)/2$ such rows of A. The $n \cdot (n-1)/2$ elements of $\vec{b}$ are the elements of the difference triangle. The requirement than none of these elements of $\vec{b}$ be zero means that no two elements of the Costas array vector $\vec{c}$ are the same; this simply requires that we have a permutation matrix.

2) Costas Condition Rows: Each of the remaining rows of the matrix A, the Costas condition rows, express the condition that the difference between each possible pair of the elements of each row of the difference triangle is not zero. Thus each row is the difference between two rows among the first $n \cdot (n-1)/2$ rows.

The Costas condition rows fall into $(n-2)$ blocks, one for each row of the difference triangle except for the last; the last row of the difference triangle has only one element so there are zero differences between its elements. Row i of the difference triangle has $(n-i)$ elements; each element $d(i,j)$ has $(n-i-1)$ possible differences between $d(i,j)$ and $d(i,j+k)$. Each of these rows of A, $\overrightarrow{A(ai)}$, is the difference between two of the first $n \cdot (n-1)/2$ rows and produces an element of $\vec{b}$ equal to $d(i,j+k) - d(i,j)$:

$$\overrightarrow{A(ai,:)}^T \cdot \vec{c} = b(ai) = d(i,j+k) - d(i,j) \tag{5}$$

TABLE V
MATRIX A FOR ORDER $n = 3$

	Col 1	Col 2	Col 3
d(1,1)	-1	+1	0
d(1,2)	0	-1	+1
d(2,1)	-1	0	+1
d(1,1)-d(1,2)	-1	+2	-1

TABLE VI
MATRIX A FOR ORDER $n = 4$

	Col 1	Col 2	Col 3	Col 4
d(1,1)	-1	+1	0	0
d(1,2)	0	-1	+1	0
d(1,3)	0	0	-1	+1
d(2,1)	-1	0	+1	0
d(2,2)	0	-1	0	+1
d(3,1)	-1	0	0	+1
d(1,1)-d(1,2)	-1	+2	-1	0
d(1,1)-d(1,3)	-1	+1	+1	-1
d(1,2)-d(1,3)	0	-1	+2	-1
d(2,1)-d(2,2)	-1	+1	+1	+1

where the row $\overrightarrow{A(ai)}$ is the difference between two rows of the first $n \cdot (n-1)/2$ rows as defined by (52) to produce the difference between two elements in a given row of the difference triangle as given by (2).

3) Examples: The matrix A for order $n = 3$ and $n = 4$ is shown as Tables V and Table VI. The rows are labeled in the tables to show that the rows above the line correspond to the elements $d(i, j)$ of the difference triangle; the rows below the line correspond to differences between elements of the difference triangle.

In the examples, the order of rows below the line is chosen for clarity in illustration, but the order is different in the database because the software that generated the database stepped the indices differently. As we will show in Section IV-D, the order of the rows of A and their signs have a one-to-one correspondence with the rows of the left eigenvector matrix VL and their signs but have no effect on the eigenvalues or the right eigenvectors.

4) Remarks: The lower portion of the matrix A that expresses the inequality of elements of a pair of rows of the difference triangle definition rows was suggested in [11].

Note that there is a duplicate row in the matrix A for order $n = 4$ in Table VI. In the portion of the matrix A representing differences between elements of the difference matrix, we note that

$$\begin{aligned} d(i_1, j_1) - d(i_2, j_2) = \\ (c(i_1 + j_1) - c(j_1)) - (c(i_2 + j_2) - c(j_2)) = \\ (c(i_1 + j_1) - c(i_2 + j_2)) - (c(j_1) - c(j_2)) = \\ d((i_1 + j_1) - (i_2 + j_2), i_2 + j_2) - d(j_1 - j_2, j_2) \end{aligned} \tag{6}$$

predicts duplications in the rows of the matrix A. The number of duplicate Costas condition lines is

$$N_{Dups}(n) = \left\lfloor \frac{(n-1) \cdot (n-3) \cdot (2n-1)}{24} \right\rfloor \tag{7}$$

$N_{Dups}(n)$ is about one-half of the total number of rows of the matrix A when n is large. However, if the duplicate rows are omitted, the eigenvalues of the smaller matrix are not all square roots of integers. For this reason, we do not omit duplicate rows in the matrix A in our singular value decomposition formulations.

This redundancy is noted in [14] and used in [15] in connection with reducing resource requirements that use logic related to the difference triangle to determine whether or not a permutation matrix is a Costas array.

Theorem 3.1 (Number of Rows of A): The matrix A is constructed as follows.

The augmented portion of A is the portion in which the elements $d(i, j)$ of the difference matrix are constrained to be nonzero and has N_P rows,

$$N_P = \frac{n \cdot (n-1)}{2}. \tag{8}$$

In each of these rows, we place a $+1$ in column $(i + j)$ and a -1 in column j, and the remaining elements of this row zero; this implements (2).

The number of conditions that require that no two elements on row i of the difference triangle be equal, $Count_{NR}(i)$, is equal to the number of combinations of two elements of row i of the difference triangle, and the total number of Costas condition rows is the sum of these for $1 \leq i \leq (n-1)$, and the total for all rows, is

$$Count_{NR}(i) = \frac{(n-i) \cdot (n-i-1)}{2} \tag{9a}$$

$$Count_{TNR} = \sum_{i=1}^{n} Count_{NR}(i) = \frac{(n-2) \cdot (n-1) \cdot n}{6} \quad . \tag{9b}$$

The total number of rows m of A is the sum of N_P and $Count_{TNR}$,

$$m = \frac{(n-1) \cdot n \cdot (n+1)}{6}. \tag{10}$$

Proof: The counts in row i of the difference triangle follow from the preceding and included narratives. The total count $Count_{TNR}$ is found by summing the counts $Count_{NR}(i)$ on the row i from 1 to $(n-1)$, or, more conveniently, on $j = n - i$ summed from 1 to $n - 1$. ■

C. Singular Value Decomposition

1) Definitions: The augmented matrix A as formulated in III-A was subjected to a numerical singular value decomposition that was executed using numerical methods outlined in [16] and [17], with restructuring and other changes by the author, in 2012, 2014 and 2016. These SVDs were applied for orders to 35, beyond which practical considerations and the adequacy of that data indicated that this is sufficient for the numerical basis of this work. Extended precision was required and used for the higher orders. See Section VII for a discussion of mathematics, numerical analysis, and computer resources in this first work.

From a first look that computed the right eigenvectors and squared eigenvalues from a SVD of $[A^T \cdot A]$ using Jacobi's method, patterns were noted and simple closed forms were formulated and checked against numerical results for orders form 3 through 35.

A full singular value decomposition of the matrix A that produces the left eigenvectors was also done using the methods of [17] as modified by the author to meet the challenges such as an A matrix with millions of elements. The preliminary and closed form results were checked against these new numerical results.

For the purposes of establishing notation, the SVD is stated as

$$A = VL \cdot \Lambda \cdot V^T \tag{11}$$

where VL is an m by n matrix of n orthogonal left eigenvectors, and V is an n by n matrix whose columns are the right eigenvectors of the m by n matrix A. The row count for A and VL is m as given by (10), and Λ is a diagonal n by n matrix of eigenvalues. Both left and right eigenvectors may be scaled to vectors that have integral elements. The mathematics and numerical methods that were used to produce the scaling to integers of each eigenvectors is explained in Appendix B.

We denote the eigenvector matrices that have been scaled to have elements that are all integers, and these scalings, as

$$VL = IVL \cdot DL^{-1} \tag{12a}$$
$$V = IV \cdot DR^{-1} \tag{12b}$$

where DL is a diagonal matrix of scale factors for the corresponding columns of IVL, and DR is a diagonal matrix of common denominators for the corresponding columns of IV,

$$DL^2 = IVL^T \cdot IVL \tag{13a}$$
$$DR^2 = IV^T \cdot IV. \tag{13b}$$

Using (12a), (12b), (13a) and (13b), we can pose (11) as

$$A = IVL \cdot DL^{-1} \cdot \Lambda \cdot DR^{-1} \cdot IV^T \tag{14}$$

The method presented in Appendix B was used to provide integer numerators and denominators for rational ratios between eigenvector elements, followed by finding the greatest common divisor (GCD) of the two numbers to express each eigenvector as a set of ratios with a common denominator, which became an element of DL or DR. The method inherently eliminates common factors in the integer ratios. When closed forms are used to produce IVL and IV using the method presented in Section IV, common factors are not removed, but the corresponding DL and DR can be produced directly from IVL and IV as the square root of the squared lengths of the columns of IVL and IV using (13a) and (13b).

We also can combine the lengths of columns of IVL and IV with the eigenvalue matrix Λ to form a scale factor diagonal matrix SF:

$$SF^2 = DL \cdot \Lambda^{\#} \cdot \Lambda^{\#} \cdot DR \tag{15}$$

where $\Lambda^{\#}$ is the pseudoinverse of Λ,

$$\Lambda^{\#} = \text{Diag}\left(0, \frac{1}{\lambda_2}, \frac{1}{\lambda_3}, ... \frac{1}{\lambda_n}\right). \tag{16}$$

With (15) we can write (14) in as a factorization using integer quantities from the database files,

$$A = IVL \cdot SF^{\#} \cdot IV^T \tag{17}$$

where $SF^{\#}$ is the pseudoinverse of SF,

$$SF^{\#} = \text{Diag}\left(0, \frac{1}{SF(2,2)}, \frac{1}{SF(3,3)}, ... \frac{1}{SF(n,n)}\right).$$

Use of (17) to reconstruct A, followed by subtraction of A as stored, is the simplest way to verify that IV, IVL and Γ are valid. This is simplified by the fact that the scale factors from (15), the diagonal elements of SF, are always integers, notwithstanding the square roots involved in finding SF from DL, Λ, and DR. Number theory applied to (17) shows that SF must be rational because A, IVL and IV are all rational. We have scaled IVL and IV so that SF is all integers.

2) Proof of the Rank of A: The matrix A is not of full rank because the sum of the elements of every row of A is zero, so that multiplying the matrix A by a vector $\vec{v}_0$ with elements that are all the same value will produce a zero vector; thus $\vec{v}_0$ normalized to unit length is an eigenvector with an eigenvalue of zero. We state the rank of A, $(n-1)$ as a theorem.

Theorem 3.2 (Rank of A is $(n-1)$): The rank of the matrix A as defined in Section III-A is one less than the order of the Costas arrays.

Proof: We express the elements of the eigenvector proportional to $\vec{c}_0$ as

$$v_i = \frac{1}{\sqrt{n}}, \; 1 \leq i \leq n. \tag{18}$$

The the elements of the vector produced by left-multiplying A into $\vec{v}_0$ are all zeros because the first $n \cdot (n-1)/2$ rows are differences between two elements of $\vec{v}_0$, and these differences are always zero; the remaining rows are differences between two of these results, which, of course, is also always zero. Thus the rank of the matrix A cannot exceed $(n-1)$.

No vector $\vec{v}_0'$ can produce all zero elements of $\vec{b}$ in the first $n \cdot (n-1)$ rows because for every element of $\vec{v}_0'$ there is a row of A among the first $n \cdot (n-1)$ rows that produces an element of $\vec{b}$ that is the difference between this element and another element of $\vec{v}_0$, and if any two elements are different from each other the corresponding element of $\vec{b}$ will be the nonzero value of that difference. Thus all the elements of $\vec{v}_0'$ must be equal, so it is proportional to $\vec{v}_0$, and we have established the theorem. ■

3) Eigenvalues and Right Eigenvectors: By definition of the SVD, the columns of the matrices VL and V are orthogonal and of unit length. Substituting from (11) in (3), we have

$$A \cdot \vec{c} = VL \cdot \Lambda \cdot V^T \cdot \vec{c} = \vec{b}. \tag{19}$$

The Costas condition is that all elements of $\vec{b}$ be nonzero. When the elements of $\vec{c}$ are integers from 0 to $(n-1)$ or from 1 to n, and (19) is met, then $\vec{c}$ is a Costas array vector, and vice versa.

Note that (19) by itself is necessary but not sufficient for $\vec{c}$ to be a Costas array vector, because (19) by itself does not constrain the range of values of the elements of the Costas array vector $\vec{c}$, which is usually 0 to $(n-1)$ or 1 to n.

4) Finding the Left Eigenvectors from A, V and Λ: When the right eigenvectors are found by operating on $A^T \cdot A$, we use (11) to obtain

$$A^T \cdot A = V \cdot \Lambda^2 \cdot V^T \tag{20}$$

to find V and Λ^2, which has the advantage of producing the right eigenvalues and squared eigenvalues from a real, symmetrical n by n matrix. Since (10) shows that $m \gg n$, we accrue advantages in the computational requirements and accuracy. When this is done, the left eigenvectors may be found from (11) by right-multiplying both sides first by V and then by $\Lambda^{\#}$,

$$VL \cdot I^{\#} = A \cdot V \cdot \Lambda^{\#} \tag{21}$$

where $\Lambda^{\#}$ is the pseudoinverse of Λ as given by (16), which must be used in (21) because A and thus Λ is not full rank, and $I^{\#}$ is given by

$$I^{\#} = Diag\,(0, 1, 1, ...1)\,. \tag{22}$$

Note that (21) will not define the first column of VL, which we store as a zero vector in the database.

The nonzero eigenvalues can be taken as positive without loss of generality. This is seen by noting that (11) can be written

$$A = \sum_{i=1}^{n} \overrightarrow{vl}_i \cdot \lambda_i \cdot \overrightarrow{vr}_i^T$$

which shows that a sign change in λ_i is accompanied by a sign change in $\overrightarrow{vl_i}$.

5) Numerical SVD for Orders 3 Through 35: The author performed an SVD on the matrix A for orders 3 through 35. Two numerical SVDs were performed for each order. One SVD of the n by n matrix $[A^T \cdot A]$ uses Jacobi's method ([16] Contribution II/1 *The Jacobi Method for Real Symmetric Matrices* pp 202-211 by H. Rutishauser), which produces the squared eigenvalues and the right eigenvectors, but not the left eigenvectors.

A more general method ([16] Contribution I/10 *Singular Value Decomposition and Least Squares Solutions* pp 134-151 by G. H. Golub and C. Reinsch) and [17] uses Householder transformations to bi-diagonalize a square or rectangular matrix followed by a recursive chasing procedure to diagonalize the bidiagonal matrix. The procedure operates as a matrix factorization by initializing an $m \times m$ left-multiplier and an $n \times n$ right-mulitplier as identity matrices, then mirroring the simple individual operations on the subject matrix with their inverses on the left and right multipliers. This procedure produces both left and right eigenvectors and the eigenvalues. The eigenvalues and right eigenvectors agree between the two numerical results. A process presented in detail in Appendix B used the extended accuracy floating point results for VL and V to produce the integer ratios between the elements of each eigenvector, which was used to produce the matrices IVL and IV.

IV. Simple Polynomials for Squared Eigenvalues and Integral Scaled Elements of Eigenvectors

The following summary was originally obtained by studying the results of the numerical SVDs of the matrix A for orders 2 through 35 and producing the elements of the eigenvalues as ratios of integers. The properties of the eigenvalues and right eigenvectors, with the eigenvalues placed in order of increasing magnitude are summarized as:

- The eigenvalues, squared, are always integers. We order the eigenvalues in order of increasing magnitude. The first eigenvalue is zero. Other than the second eigenvalue, which is $\sqrt{n}$, and the largest eigenvalue for odd n, the rest of the eigenvalues are in pairs of equal numbers.
- All of the right eigenvectors are either odd or even about their center point.
- The right eigenvectors for a given order, except for the first two, are repeated for higher orders; eigenvectors for an order that is 2 larger will reproduce $(n-2)$ of the eigenvectors.
- The right eigenvector matrix can be updated when the order is increased by two. You rescale and add end elements to the leftmost two eigenvectors to find the second two eigenvectors, and add a linear ramp and a constant eigenvector as the first two eigenvectors.
- The added end elements to the second two eigenvectors are selected to make them orthogonal to new eigenvectors that are either constant values or linear ramps. These two conditions determine the end values.

The eigenvectors for Costas arrays of odd and even orders have different forms. As a minor simplification in normalizing eigenvectors, we also present equations for sums of squares of elements of eigenvectors. These can be computed from the equations for the eigenvector elements algebraically or by fitting a polynomial to numerical results. We present the polynomial equations for elements of eigenvectors for odd and even Costas array orders separately. A proof of validity and uniqueness of the polynomials is given in Section V-A.

A. A Polynomial for the Squared Eigenvalues

The squared eigenvalues were placed in order of increasing magnitude and fitted with a polynomial in n and the order j in which they appear, $1 \le j \le n$, in order of increasing magnitude. This polynomial, which fits values of squared eigenvalues obtained numerically from $n = 3$ through $n = 35$, is

$$\lambda_j^2 \begin{cases} = 0 & j = 1, \\ = n & j = 2, \\ = \frac{n\cdot(n+1)}{2} + (n+1)\cdot\lfloor\frac{j-2}{2}\rfloor - \lfloor\frac{j-2}{2}\rfloor^2, & 3 \le j \le n. \end{cases} \tag{23}$$

Note that the operation of truncation in (23) means that eigenvalues are repeated in pairs. With each pair of identical eigenvalues, there will be two corresponding eigenvectors of V; one of these paired eigenvectors will be odd about the center point and the other will be even.

B. Right Eigenvectors for Odd Order n

Here we present the right eigenvector matrix V by computing IV and the diagonal elements of DR^2

The eigenvalues for odd order, in order of increasing magnitude, are 0, then $\sqrt{n}$, then there are $(n-3)/2$ paired values, and finally a unique largest eigenvalue. The first column of IV, corresponding to the zero eigenvalue, is all ones, as explained above in III-C3,

$$IV_{i,1} = 1, 1 \le i \le n. \tag{24}$$

The second column of IV is a linear ramp, symmetrical about the center of the column with center element zero,

$$IV_{i,2} = i - \frac{n+1}{2}, 1 \le i \le n. \tag{25}$$

The eigenvectors, after the first two, repeat for all successive odd orders, with zeros added at both top and bottom to keep the nonzero values centered in each column.

The last eigenvector, which has its own unique eigenvalue $(n+3)\cdot(3n-5)/4$, is even, and is just three numbers,

$$IV_{*,n} = \begin{bmatrix} \vdots \\ 0 \\ -1 \\ +2 \\ -1 \\ 0 \\ \vdots \end{bmatrix} \tag{26}$$

This is the beginning of a pattern for all odd-numbered eigenvectors beyond the first, for which the center $(2(n-j)+1)$ elements are all $+2$. The general form for the eigenvectors is that each pair of eigenvectors for columns $2j-1$ and $2j$, $2 \le j \le n$, are even and odd about the center element, respectively. The odd-numbered eigenvectors are given by

$$IV_{i,j} \begin{cases} = 2, \frac{j+1}{2} \le i \le n - \frac{j+1}{2} + 1 \\ = -(n-j+1), i = \frac{j-1}{2}, \text{ or } i = n - \frac{j-3}{2} & n, j \text{ odd} \\ = 0 \text{ otherwise} \end{cases} \tag{27}$$

That is, the center $(n-j+1)$ elements are $+2$, and the end elements are $-(n-j+1)$ so that the sum of all the elements of that eigenvector is zero.

The even-numbered eigenvectors are linear ramps, odd about the center, with center element zero, but with end elements that makes them orthogonal to longer linear ramps, and are given by

$$IV_{i,j} \begin{cases} = \left(i - \frac{n+1}{2}\right)\cdot(m+1), \frac{j}{2} \le i \le n - \frac{j}{2} + 1 \\ = -\frac{m\cdot(m+1)\cdot(2m+1)}{6}, i = n - \frac{j}{2} + 2 \\ = +\frac{m\cdot(m+1)\cdot(2m+1)}{6}, i = \frac{j}{2} - 1 \\ = 0 \text{ otherwise} \end{cases} \tag{28}$$

where, in (28), m is the count of even-numbered eigenvectors starting from the right,

$$m(n,j) = \frac{n-j+1}{2}.$$

Note that the center element is always zero.

The squared lengths of the eigenvectors are given by

$$\sum_{i=1}^{n} IV_{i,j}^2 \begin{cases} = n, \ j = 1 \\ = \frac{(n-1)\cdot n\cdot(n+1)}{12}, \ j = 2 \\ = 2\cdot(2p-1)\cdot(2p+1), \ j = 3, 5, ..., n \\ = \frac{p(p+1)^2\cdot(p+2)\cdot(2p+1)\cdot(2p+3)}{18}, \ j = 4, ..., n-1 \end{cases} \tag{29}$$

where p in (29) is given by

$$p = \left\lfloor \frac{n-j+2}{2} \right\rfloor$$

and is the count of eigenvalue pairs from the right. The floor function for p is needed in (29) for even j.

C. Right Eigenvectors for Even Order n

As with odd orders, the first column of IV, corresponding to the zero eigenvalue, is all ones, as explained above in III-C3,

$$IV_{i,1} = 1, 1 \le i \le n. \tag{30}$$

Again as with the even orders, the second column of IV is a linear ramp. When the order n is even, there is no center element, and the minimum increment for a linear ramp is 2. So,

$$IV_{i,2} = 2i - n - 1, 1 \le i \le n. \tag{31}$$

The even numbered eigenvectors, odd about the center, are given by

$$IV_{i,j} \begin{cases} = 6\cdot(i - \frac{n-1}{2}), \ \frac{j}{2} \le i \le (n - \frac{j}{2} + 1), \ j = 4, 6, ..., n \\ = +p\cdot(2p-1), \ i = \frac{j}{2} - 1 \ j = 4, 6, ..., n \\ = -p\cdot(2p-1), \ i = n - \frac{j}{2} + 2, \ j = 4, 6, ..., n \\ = 0 \text{ otherwise} \end{cases} \tag{32}$$

where p in (32) is given by

$$p = \frac{n-j+2}{2}$$

is again the count of eigenvector pairs from the right.

The odd numbered eigenvectors, which are even about the center, are given by

$$IV_{i,j} \begin{cases} = 1, \ \frac{j+1}{2} \le i \le n - \frac{j+1}{2} + 1, \ j = 3, 5, ..., n-1 \\ = -(p-1), \ i = \frac{j+1}{2} - 1, \ j = 3, 5, ..., n-1 \\ = -(p-1), \ i = n - \frac{j+1}{2} + 2, \ j = 3, 5, ..., n-1 \\ = 0 \text{ otherwise} \end{cases} \tag{33}$$

where p in (33) is

$$p = \frac{n-j-1}{2}$$

is again the count of eigenvector pairs from the right, this one time counting from 0 instead of from 1.

The squared lengths of the eigenvectors are given by

$$\sum_{i=1}^{n} IV_{i,j}^2 \begin{cases} = n, \ j = 1 \\ = \frac{(n-1)\cdot n\cdot(n+1)}{3}, \ j = 2 \\ = 2p(p+1)(2p-1)(2p+3), \ j = 4, 6, ..., n \\ = 2p\cdot(p+1), \ j = 3, 5, ..., n-1 \end{cases} \tag{34}$$

where p in (34) is given by

$$p = \left\lfloor \frac{n-j+2}{2} \right\rfloor$$

and is the count of eigenvalue pairs from the right. The floor function for p is needed in (29) and (34) for odd j, and is used in both for consistency.

D. Left Eigenvectors

We need a method of finding VL, IVL and DL for large orders because the resources requirements of the methods of Appendix B exceed order n^4; the number of rows of the matrix A is given by Theorem 2.1 as of order n^3 and the matrix A has n columns; the precision required to compute rational ratios for the elements of the VL and V matrices increases with order, as inferred from the equations for the eigenvalues and elements of IV given in Section IV.

The SVD as presented in (14) can be rearranged to place IVL and DL on the left hand side,

$$VL = IVL \cdot DL^{-1} = A \cdot IV \cdot DR^{-1} \cdot \Lambda^{\#}. \quad (35)$$

At this point we address a fundamental ambiguity in the definition of the matrix A and how that maps to the matrix IVL. There is no mathematical rationale that uniquely defines the order of the rows of A. In fact, left multiplying (35) by any m by m permutation matrix P,

$$P \cdot VL = P \cdot IVL \cdot DL^{-1} = P \cdot A \cdot IV \cdot DR^{-1} \cdot \Lambda^{\#} \quad (36)$$

shows that nothing we have done up to this point is affected by the row order of A. If we additionally change the signs of rows of P we change the signs of the corresponding rows of IVL but again we do not affect the eigenvalues or right eigenvectors. We proceed by noting that A is sparse, and each row has two, three, or four nonzero elements that are -1, $+1$, or $+2$. Thus we write $VL = IVL \cdot DL^{-1}$ as sums of a few rows of $V \cdot \Lambda^{\#}$. And, we do not further express the permutation matrix P.

From (35), we see that

$$\begin{aligned} \text{Row}(i \text{ of } VL) = &+ \sum_{a_{i,k1}=+1} \text{Row}(k1 \text{ of } V \cdot \Lambda^{\#}) \\ &+ 2 \sum_{a_{i,k2}=+2} \text{Row}(k2 \text{ of } V \cdot \Lambda^{\#}) \\ &- \sum_{a_{i,k3}=-1} \text{Row}(k3 \text{ of } V \cdot \Lambda^{\#}) \end{aligned} \quad (37)$$

Equation (37), with the equations of Section IV is as close to a closed form for the elements of VL as we will get. Note that one of the three sums in (37) will always be null, and that there are only a total of three or four terms in all the sums combined.

Given the elements of VL, the methods of Appendix B are used to find the elements of IVL and DL. An additional relationship is revealed by examination of results for orders 3 through 35: the elements of SF as given by (15) are equal to those of DR^2, sometimes with a small integer multiple. So, using (15) and this observation, a working solution to this problem is

$$SF = DL \cdot \Lambda^{\#} \cdot DR = k \cdot DR^2$$

or,

$$DL \cdot I^{\#} = k \cdot DR \cdot \Lambda \quad (38)$$

where k is a small integer. The product $\Lambda^{\#} \cdot \Lambda$ is a modification $I^{\#}$ of the identity matrix I in which the first element at $(1,1)$ is zero, as given in (22).

Using (38) to left-multiply the last two forms in (35), we have IVL as

$$\begin{aligned} IVL \cdot I^{\#} &= k \cdot VL \cdot DR \cdot \Lambda \\ &= k \cdot \left[A \cdot IV \cdot DR^{-1} \cdot \Lambda^{\#}\right] \cdot DR \cdot \Lambda \\ &= k \cdot A \cdot IV \cdot I^{\#}. \end{aligned} \quad (39)$$

In developing (39) we have noted that diagonal matrices are commutative in multiplication.

The first column of IVL is left undefined by (39). This is not a functional problem because $\lambda_1 = 0$, which removes the first columns of IVL and IV from the results of any computation involving Λ such as reconstruction of A.

If VL is found by numerical methods then the full m by m left eigenvector matrix is available including the ”don’t-care” columns 1 and those past n. If a first eigenvector of length m is required, the only requirement is that it be orthogonal to the other $n-1$ left eigenvectors. The null space of the left eigenvectors is given by the columns of the m by m matrix of rank $(m-n+1)$

$$I - IVL \cdot \left[IVL^T \cdot IVL\right]^{\#} \cdot IVL^T.$$

The ”don’t-care” first eigenvector that is orthogonal to the significant $n-1$ left eigenvectors can be taken as any linear combination of the columns of this matrix. In the database [12], this ”don’t-care” vector is simply stored as all zeros.

To use IV with (39), the common factors in the eigenvalues should not be removed before IVL is computed so that $k \geq 1$ is achieved consistently. Once IVL, IV, DL and DR have been defined then the common factors in all the eigenvectors can be removed.

Coincidentally we can use (38) to produce an estimate of the lengths of the squared left eigenvectors. When we follow the preceding procedures to obtain the matrix IVL, we will get squared left eigenvector lengths that are k^2 times the squared right eigenvector lengths,

$$\sum_{i=1}^{m} IVL_{i,j}^2 = k^2 \cdot \left[\sum_{i=1}^{n} IV_{i,j}^2\right] \cdot \lambda_j^2. \quad (40)$$

V. The SVD Databases

On IEEE Dataport, the author has posted two SVD databases [12], one that includes left eigenvectors and the A matrices for orders 3 through 100, and a separate database of eigenvectors and right eigenvectors to order 1030.

The database that consists of files to order 100 includes the eigenvalues and the left and right eigenvectors for the specific form of the matrix A that specifies the Costas conditions as

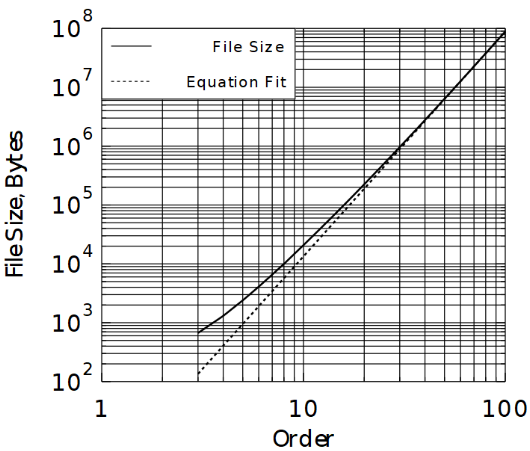

Fig. 1. Database File Sizes as a Function of Order.

explained in Section II-B; the matrix A is also provided in each database file to simplify verification. We have verified this database as explained below in Section VII-C. Residual numerical residues that are plotted as Figs. 2 and 6 show that the reconstruction is accurate to within the precision used, floating point with 112 bits in the mantissa, when the amount of computation as illustrated by the file sizes as shown in Fig. 1 is taken into account.

The integer format, enabled by the results of Section IV, is made possible through the fact that the eigenvalues are square roots of integers, and the elements of each eigenvector are rational multiples of each of the other elements of the same eigenvector. This allows the database to be used to compute the eigenvalues and eigenvectors to whatever precision is required by the user.

The database contains an additional vector of scale factors for each order that collects the denominators for the eigenvectors and the eigenvalues as give by (15) so that the reconstruction of the matrix A can be written entirely in terms of integers as in (17).

A second database that omits the left eigenvectors is also available on IEEE DataPort [12] as a separate set of files. This database is for orders 3 through 1030.

A. Proof of Validity of the Database

1) Summary of Results to be Proven: In Section III-C2 we proved in Theorem 3.2 that one, and only one, of the eigenvalues of the matrix A is zero. We note that we have, without loss of generality, taken the remaining eigenvalues as positive and placed them in order of increasing magnitude.

We have performed a numerical SVD for the matrix A and scaled the results to provide an integer database to order 35, generalized that result to simple polynomials as provided in Section IV.

The form of the right eigenvector matrices is given in Section IV. In particular, we established by example through numerical methods that the form of the SVD of the matrix A as constructed in accordance with Section III, through order 35, has the features

- The rank of A and thus Λ is $(n-1)$, as proven in Theorem 3.2.
- The eigenvalues, in order of increasing magnitude, are given by (23).
- The right eigenvectors are given as presented in Sections IV-C and IV-B.
- The left eigenvectors are given in terms of the right eigenvectors, the eigenvalues, and A by (21).
- The elements of the matrix $[A^T \cdot A]$ as given in (42) and (43) in Section V-A2 below with examples in Table VII.

We have used these equations to provide eigenvectors and both left and right eigenvectors to order 100. Because the number of rows of the matrix A and IVL with associated increasingly large file sizes that grow as the fourth power of the order and the limited utility of left eigenvectors in radar applications, the eigenvalues and right eigenvectors only are provided for orders higher than 100; the database includes an entirely separate set of files of eigenvalues and right eigenvectors for orders from 3 to 1030.

The left eigenvectors VL may be left as dependent on Λ and V because they are computed from these and the matrix A using (21). Also, the elements of the diagonal scale factor matrix SF are included in the database to order 1030, and IVL is computed by right-multiplying (17) by IV and then by SF to obtain

$$IVL \cdot I^{\#} = A \cdot IV \cdot SF \tag{41}$$

to complete the SVD; right-multiplication by $I^{\#}$, given by (22), arises from multiplication of SF by its pseudoinverse and indicates that the first column of IVL is not defined by (41). The user may construct a matrix A according to the procedure given in Section III-A.

2) Structure and Symmetries of the matrix $[A^T \cdot A]$: We define the right eigenvectors and eigenvalues to provide a framework for a proof as follows.

Table VII shows $[A^T \cdot A]$ for orders 4 through 8 which serves to support these results:

- The $[A^t \cdot A]$ matrix is palindromic between upper left and lower right, or along the main diagonal.
- The $[A^T \cdot A]$ matrix is palindromic between the lower left and the upper right, or along the antidiagonal.
- In particular, the main diagonal and antidiagonal are palindromic.
- The trace of $[A^T \cdot A]$ is equal to the sum of the squares of the eigenvalues as given in the database.
- The sums of elements for each row and for each column are all zero.

The diagonals of $[A^T \cdot A]$ are given for orders 4 through 13 in Table VIII. The top row of Table VIII is the orders, the bottom row is the trace as well as the sum of the squared eigenvalues of A for each order. This sum is

$$\sum_{i=1}^{n} \lambda_i^2 = \text{trace}(A^T \cdot A) = \left\lfloor \frac{(n^2-1)\cdot(4n-3)}{6} \right\rfloor . \tag{42}$$

Equation 42 agrees with the sum of the squared eigenvalues as given by (23).

TABLE VII
MATRICES $[A^T \cdot A]$ FOR ORDERS 4 THROUGH 8

Order 4								
	6	-5	-2	1				
	-5	10	-3	-2				
	-2	-3	10	-5				
	1	-2	-5	6				
Order 5								
	10	-7	-4	-1	2			
	-7	15	-5	-2	-1			
	-4	-5	18	-5	-4			
	-1	-2	-5	15	-7			
	2	-1	-4	-7	10			
Order 6								
	15	-9	-6	-3	0	3		
	-9	21	-7	-4	-1	0		
	-6	-7	25	-5	-4	-3		
	-3	-4	-5	25	-7	-6		
	0	-1	-4	-7	21	-9		
	3	0	-3	-6	-9	15		
Order 7								
	21	-11	-8	-5	-2	1	4	
	-11	28	-9	-6	-3	0	1	
	-8	-9	33	-7	-4	-3	-2	
	-5	-6	-7	36	-7	-6	-5	
	-2	-3	-4	-7	33	-9	-8	
	1	0	-3	-6	-9	28	-11	
	4	1	-2	-5	-8	-11	21	
Order 8								
	28	-13	-10	-7	-4	-1	2	5
	-13	36	-11	-8	-5	-2	1	2
	-10	-11	42	-9	-6	-3	-2	-1
	-7	-8	-9	46	-7	-6	-5	-4
	-4	-5	-6	-7	46	-9	-8	-7
	-1	-2	-3	-6	-9	42	-11	-10
	2	1	-2	-5	-8	-11	36	-13
	5	2	-1	-4	-7	-10	-13	28

TABLE VIII
MAIN DIAGONALS OF $A^T \cdot A$ FOR ORDERS 4 THROUGH 13

4	5	6	7	8	9	10	11	12	13
6	10	15	21	28	36	45	55	66	78
10	15	21	28	36	45	55	66	78	91
10	18	25	33	42	52	63	75	88	102
6	15	25	36	46	57	69	82	96	111
	10	21	33	46	60	73	87	102	118
		15	28	42	57	73	90	106	123
			21	36	52	69	87	106	126
				28	45	63	82	102	123
					36	55	75	96	118
						45	66	88	111
							55	78	102
								66	91
									78
32	68	122	200	304	440	610	820	1072	1372

TABLE IX
ANTI-DIAGONALS OF $A^T \cdot A$ FOR ORDERS 4 THROUGH 13

4	5	6	7	8	9	10	11	12	13
1	2	3	4	5	6	7	8	9	10
-3	-2	-1	0	1	2	3	4	5	6
-3	18	-5	-4	-3	-2	-1	0	1	2
1	-2	-5	36	-7	-6	-5	-4	-3	-2
	2	-1	-4	-7	60	-9	-8	-7	-6
		3	0	-3	-6	-9	90	-11	-10
			4	1	-2	-5	-8	-11	126
				5	2	-1	-4	-7	-10
					6	3	0	-3	-6
						7	4	1	-2
							8	5	2
								9	6
									10
-4	18	-6	36	-8	60	-10	90	-12	126

The center element of each diagonal is given by

$$[A^T \cdot A]\left(\left\lfloor\frac{n+1}{2}\right\rfloor, \left\lfloor\frac{n+1}{2}\right\rfloor\right) \begin{cases} = (3*n^2-8)4, \; n \text{ even} \\ = (3*n^2-3)/4, \; n \text{ odd.} \end{cases} \tag{43}$$

The antidagonals are even more revealing, as shown in Table IX. Again, the top row is the order and the bottom row is the sum. When the order n is even, the sum is -1; when the order n is odd, the sum is the main diagonal element and the off-diagonal elements sum to zero. The corner elements are $(n-3)$ and along the antidiagonal they decrease by 4 from lower left, upper right toward the main diagonal. A similar pattern is seen in the rows of Table IX: the first nonzero element is $(n-1)$ as expected, and the elements decrease by 3 toward the center, but after the center the values start over at $-(2i-1)$ and increase by 1. The center is marked by an element of the main diagonal for odd numbered rows, but not for even numbered rows.

The elements of the main diagonal of $[A^T \cdot A]$ matrix exhibit these properties:

- The elements fall away linearly from the center, in steps of $\{3, 5, 7, ..\}$ for odd n and steps of $\{4, 6, 8, ...\}$ for even n.
- Similar patterns are seen in diagonals parallel to the main diagonal except that the falloff is in steps of 2.
- Number patterns repeat among sequences of $[A^T \cdot A]$ matrices of odd order and for even order.
- The values in each of these subdiagonals increases by 4 in each step away from the main diagonal.
- The sum of each row and column is zero, which is to be expected since an eigenvector associated with an eigenvalue of zero is a vector with constant elements.

The elements of the eigenvectors are up to sixth order polynomials, but we have those in the computed database too. We use the same argument there, then invoke equation (22) to include the left eigenvectors in the proof, and we are done.

3) Note Concerning Form of Results: We have shown that the elements of the matrix $[A^T \cdot A]$ are quadratic polynomials, that the squared eigenvalues of A (which are the eigenvalues of $[A^T \cdot A]$) are quadratic eigenvalues as given in (23) in Section

IV-A. The squared eigenvalues are roots of a polynomial of degree n with coefficients that are polynomials in the elements of $[A^T \cdot A]$.

The cornerstone of the proof is that, in the right hand side of the equation

$$[A^T \cdot A] = V \cdot \Lambda^2 \cdot V^T \tag{44}$$

the eigenvectors, properly scaled, and the squared eigenvalues are given by polynomials in n that evaluate to integers. This means that the elements of the eigenvectors are ratios of polynomials to square roots of polynomials.

We refer to (17) in Section III-C4 where the squared diagonal scale factor matrix SF, given by (15), is a set of integers, but the resulting matrix SF that is computed using square roots is made up of integers, which means that SF^2 as computed in (15) are all rational. Elementary number theory shows that shows that, in (17), the elements of SF are rational because the other quantities in (17) are rational. As a final note, we left-multiply both sides of (44) by V^T and right-multiply (44) by V,

$$V^T \cdot [A^T \cdot A] \cdot V = \Lambda^2$$

shows that the eigenvalues squared are be rational functions of the elements of $[A^T \cdot A]$ and the eigenvector matrix V. When the elements of the diagonal matrix Λ^2 are integers and the constraint of unit length is removed, we left and right multiply by DR from (13b) have

$$IV^T \cdot [A^T \cdot A] \cdot IV = DR^2 \cdot \Lambda^2.$$

4) Proof of Form and Uniqueness of the SVD Database: We have shown that the elements of $[A^T \cdot A]$ are quadratic polynomials, as are the squared eigenvalues. From the methods in Appendix A, sums of polynomials of a given order are polynomials of order one higher than the order the elements of the sum. From the definitions of the eigenvalues and eigenvectors in Section IV, the squared lengths of the right eigenvectors and the squared eigenvalues are given by polynomials of limited order. Equation (40) shows that the squared length of the left eigenvectors are also polynomials. This enables a uniqueness theorem for the polynomials provided in Section IV for the integer values in the database.

Theorem 5.1 (Form and Uniqueness of SVD): The values given in the database are correct and unique for all orders 3 and higher, when square roots are taken as non-negative.

Proof: All the elements of $[A^T \cdot A]$, IV and the squared eigenvalues, when separated into four groups by odd and even order of Costas array, and by odd and even column number when appropriate, are polynomials of order 6 or less (worst case being the last line on the right hand side of (29)).

The elements of the right eigenvector matrices as given in Section III-B are polynomials in the order n of the type and order of those presented in Section IV. This assertion includes the elements given as zero, which are equal to the zero polynomial. We will call the correct polynomial for any of these elements $P_G(x, j)$, where we have expressed the possible dependence on the column j of A, IV, etc. or the index of the eigenvalue; when j appears in a polynomial for an element of IV in an equation for m, a count of columns or column pairs of IV from the right, it is linear in n.

For each such element, we construct another polynomial that is the difference between the globally correct polynomial $P_G(x, j)$ and the polynomial given in Section IV $P_S(x, j)$,

$$P_x(x, j) = P_G(x, j) - P_S(x, j). \tag{45}$$

We have established that the polynomials are equal for the independent variable x equal to orders 3 through 35 and relevant column j are equal, so $P_x(x, j)$ is zero for all values of x from 3 through 35. By the Fundamental Theorem of Algebra, $P_x(x, j)$ is given by the product of its zeros and a scale factor,

$$P_x(x, j) = a \cdot \prod_{k=1}^{K} (x - c_k) \tag{46}$$

where K is the order of the polynomial, a is the proportionality constant, and the c_k are the roots of the polynomial; note c_k is equal to integers from 3 up through $K+3$, which is far less than 35.

The roots of $P_x(x, j)$ have been established to be the integers 3 through 35, which is 33 values of x. The orders of the polynomials $P_G(x, j)$ and $P_S(x, j)$ are all less than 33, but $P_x(x, j)$ is zero for at least 33 distinct values of x. Thus the proportionality constant a in 45 is zero, and $P_x(x, j)$ is identically zero. This establishes the theorem for the squared eigenvalues and the elements of the scaled right eigenvectors, and by extension through dependence, the squared lengths of the right eigenvectors and the normalized right eigenvectors, and the scale factors SF of (17).

Uniqueness is established with the understanding that the values of square roots are taken as nonnegative.

Thus we have established the theorem for all quantities in the database.

These same methods establish the form of the matrix $[A^T \cdot A]$ and equations for its elements as given in section IV-A.

■

B. Methods Used in Computing the Database

This database was produced in stages:

- A formulation of the matrix A as defined in Section III was found for which the matrix $[A^T \cdot A]$ has eigenvalues that are square roots of integers and right eigenvectors that can be scaled to integers. Since $[A^T \cdot A]$ is real and symmetrical, Jacobi's method ([16] pp 202-211) was used, as implemented by the author using quad precision. Even higher precision arithmetic was used in some investigations.
- Left eigenvectors were produced as $VL = A \cdot V \cdot \Lambda^{\#}$. Integer numerators and squared denominators of both sets of eigenvalues were found using the methods of Appendix B. An initial database was computed for orders 3 through 35 using these methods.
- A full SVD of A was found by a the numerical methods of [16] pp 134-151, with an implementation by the author that uses quad precision, and which produces VL, Λ, and V directly. Ratios of integers were again extracted

TABLE X
ZIP ARCHIVES IN THE DATABASE

File Name	File Size, MiB
Costas_Condition_SVD_Order=3-53.zip	5.8
Costas_Condition_SVD_Order=54-75.zip	20.2
Costas_Condition_SVD_Order=76-86.zip	20.7
Costas_Condition_SVD_Order=87-94.zip	21.6
Costas_Condition_SVD_Order=95-100.zip	20.6

for the elements of the eigenvalues by the methods of Appendix B. The initial database for orders 3 through 50 was validated by comparison with the first database from orders 3 through 35.

- The methods of Section IV-D, specifically the key equation (39) that gives the left eigenvector matrix numerator integers as $IVL{\cdot}I^* = k{\cdot}A{\cdot}IV{\cdot}I^*$ to provide initial values for the elements of IVL, and removing common factors in each left eigenvector. This method is used to produce the database from orders 3 through 100. Quad precision was used for floating point operations and, in some of the computations that implemented the results presented in Section IV 64-bit integers were used to avoid overflow during computation.
- Because the numbers of elements in IVL and A increase as n^4 and system performance criteria matching uses the right eigenvectors and the eigenvalues, the methods of Section IV was used to produce a database of right eigenvectors and squared eigenvalues for orders 3 through 1030. This database was validated against the full SVD database for orders 3 through 100.

C. Computer Word Lengths and Large Arguments

Most computers and compilers support IEEE floating point (standard IEEE 754) which has 23 bits of accuracy in single precision and 52 bits in double precision. Compilers and computers often support a quad precision or 16-byte (128-bit) floating point that will have 104 bits or more that may or may not be exactly as specified in IEEE 754, and 256-bit floating point is emerging. Multiple precision software packages are available that compute with user-defined precision but with a speed penalty become severe as precision expands beyond 128 bits, but this capability offers a way to verify critical computations. Quad precision is sufficient for orders to about 100, and was used to compute both the SVD database and the right-eigenvectors-eigenvalues database.

Some of the equations in Sections IV and IV-D require integers larger than the usual 32 bits to avoid overflow during computation, even if the final result is less than the maximum value of $2^{31} - 1$. This has been observed for orders above about 45. We use 64-bit integers when needed to avoid errors caused by integer overflow in computing the database.

D. Format of the Database

The actual files of the database are combined into six ZIP archives, five for the full SVD to order 100 and one for the right-eigenvectors-only database. The file names and sizes for the full SVD database are given in Table X. The right-eigenvectors-only database for orders 3 through 1030 is all in one ZIP archive, Costas_Condition_RtEigen_Order=3-1030.zip, which is 17.8 MiB in size. The ZIP archive stores the database in about 5% of the download file sizes of the uncompressed data. The ZIP format was selected as compatible with nearly all computing environments. Data files were allocated to archives to keep the maximum file size to about 20 MB. Total uncompressed size of the full SVD database files is $1,885.4$ MiB. The ZIP archives for the full SVD database total 93.3 MiB.

The format of the database is one comma-separated value (CSV) file for each order. The format of each row is a series of comma-separated integers, followed by a comma and a comment in plain text. All rows have a number of integers equal to the order, except the first row which has four integers. For the full SVD database to order 100, the CSV file content is:

1) (Order n),(No. of rows of A and IVL), (No. of order 1 rows),(No. of order 2 rows),(comment)
2) (Squared lengths of the right eigenvectors separated by commas),(comment)
3) (Squared lengths of the left eigenvectors separated by commas),(comment)
4) (Squared eigenvalues separated by commas),(comment)
5) (Scale factors as given by (15) and used by (17)),(comment)
6) (n rows, n columns; the matrix IV),(comments including row number)
7) ($(n-1)\cdot n\cdot(n+1)/6$ rows, n columns, the matrix IVL),(comments including row number)
8) ($(n-1)\cdot n\cdot(n+1)/6$ rows, n columns, the matrix A),(comments including row number)

The rows of A designated as order 1 have only one $+1$ and one -1, the remainder of the elements being zero. The rows of A designated as order 2 are the difference of two level 1 rows, and will either have two values of $+1$ and two of -1, or one of $+2$ and two of -1.

The SVD database was verified numerically by implementing (17) to reconstruct A and subtracting the matrix A as stored in the database from the result, and tracking the maximum peak value in the difference. The result is shown in Fig. 2

The second database that omits the left eigenvectors has a different file format. Contents of the SVD files is:

1) (Order n),(No. of rows of A and IVL), (No. of order 1 rows),(No. of order 2 rows),(comment)
2) (Squared lengths of the right eigenvectors separated by commas),(comment)
3) (Squared eigenvalues separated by commas),(comment)
4) (Common factors removed from columns of IV separated by commas),(comment)
5) (n rows, n columns; the matrix IV),(comments including row number)

Note that a breakdown of the types of rows of A is included in the first row; this is simply to agree with the format for the first line of the full SVD CSV files.

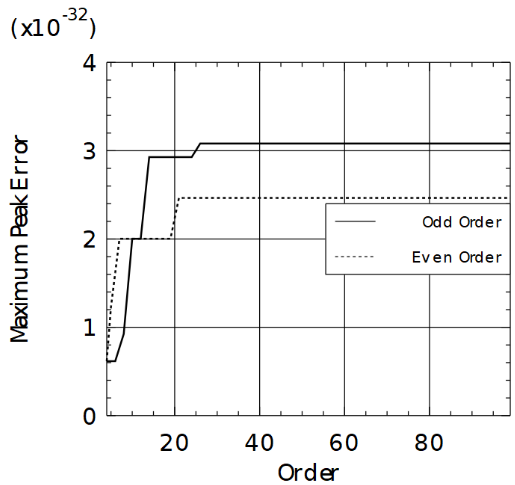

Fig. 2. Numerical Accuracy of Reconstruction of A.

TABLE XI
MATRIX IV FOR ORDER 4

Order 4	V 1	V 2	V 3	V 4
Row 1	1	-6	-1	1
Row 2	1	-3	1	-3
Row 3	1	3	1	3
Row 4	1	6	-1	-1

TABLE XII
MATRIX IV FOR ORDER AND 5

Order 5	V 1	V 2	V 3	V 4	V 5
Row 1	1	-12	-3	3	0
Row 2	1	-6	2	-6	-1
Row 3	1	0	2	0	2
Row 4	1	6	2	6	-1
Row 5	1	12	-3	-3	0

E. Right Eigvectors by Recursion

In Section III-C3 we showed that the form of the right eigenvector matrix is a series of pairs of eigenvectors for each repeated eigenvalue, one odd about the center point and one even about the center point. The rightmost eigenvector has only 4 nonzero elements for even orders and just 3 nonzero elements for odd orders. Each eigenvector has end elements that have values that make them orthogonal to longer eigevenctors that are either even and of constant value or are odd and increment by a constant amount over their nonzero elements. With this knowledge, the matrix IV can be constructed directly. The squared eigenvalues can then be found from (23), the lengths of the scaled eigenvectors as the diagonal elements of $IV^T \cdot IV$, providing all that is needed in most applications. The left eigenvectors can be found by constructing A as described in Section III-B and IVL computed using (39), the squared lengths of the left eigenvectors as the diagonal of $IVL^T \cdot IVL$ to complete the SVD.

The matrix $[A^T \cdot A]$ may also be computed directly using the properties described in Section V-A2.

The IV matrix may be constructed from the equations provide in Section IV without the need for recursion.

The construction of IV is stated as a theorem.

Theorem 5.2 (The Matrix IV is Defined Recursively): For even order $n \geq 4$ the right two eigenvectors are as given in Table XI. We denote the count of eigenvalue pairs, counting from the right, as m, with $m = 1$ being the rightmost two eigenvectors. The next pair of eigenvectors to the left are constructed as $2m$ nonzero elements each, a linear ramp odd about the center of the matrix (between rows $\pm(n/2)$) in steps of 6 as illustrated in column 2 in Table XI, and all ones, as illustrated in column 1 in Table XI. There is one more nonzero element in each eigenvector at row $\pm(n/2 + 1)$ of value selected to make that eigenvector orthogonal to longer eigenvectors that are either all constant or a linear ramp. For the first two eigenvectors, rows $\pm(n/2+1)$ are out of range and these elements are omitted, making the first two eigenvectors simple constant value and linear ramp.

For odd order $n \geq 5$ the construction is similar, but begins, counting from the right, with an even column followed by an odd column. The right two eigenvectors are as given in Table XII.

Proof: The proof follows from the equations and narratives of Sections III-C and IV, and Theorem 5.1 in Section V. ■

Note that most columns share a common factor of 3 that is removed in the database files, and may be removed in your constructed IV matrix. In one of 6 odd columns, the end element will not be divisible by 3, forcing the increment of 6 in the general statement of construction of odd eigenvectors in the theorem.

VI. USE OF THE DATABASE FOR SELECTION OF COSTAS ARRAYS FOR WAVEFORMS

A. Problem Statement

For the purposes of illustrating how we may use the SVD database, we will look at selections of subsets of Costas arrays of a given order to minimize cross-interference between systems that use the same band at the same time. The waveform for our example is a simple frequency jump burst (FJB) waveform using a Costas array of order n to assign frequency bins within a given bandwidth of n adjacent channels for n consecutive short pulses.

We leverage the availability of Costas array vectors, linearly mapped to Costas eigenspace by observing the distribution of desirable Costas arrays and constructing a metric that we use to select Costas arrays that exhibit that property. Since mapping to Costas eigenspace and weighting are simple enough for real-time system implementation, this can enable improved mutual interference performance in a dynamic environment as the scenario evolves in real time.

Since use of a Costas array of order n for frequency assignment in an FJB waveform results in n times the bandwidth and n times the duration of each pulse in the waveform, the time-bandwidth product of the FJB waveform is n^2 times the

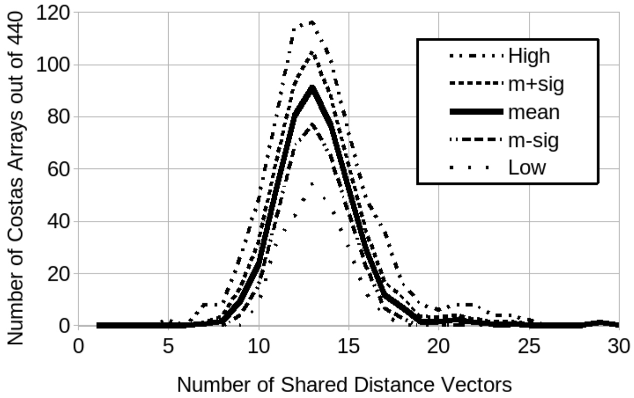

Fig. 3. Cross Interference of Costas Arrays of Order 8.

time-bandwidth product of a single pulse. In the receiver, a digital matched filter must have a minimum number of digitizes samples of two times the time bandwidth product of the signal, so we will look at low to moderate order n in our example.

We will look at the 440 Costas arrays of order $n = 8$ in the IEEE DataPort database[6]. We use shared distance vectors between Costas arrays as a metric of cross-interference. The distance vectors of a Costas array are the vectors between dots; they are easily visualized by inspecting the Costas array matrix in Table I. The number of shared distance vectors between two Costas arrays is a count of pulses that correlate simultaneously in the matched filter for a waveform using a different Costas array, for all offsets in time and frequency. A plot of how this works out among all Costas arrays of order 8 in the database is shown in Fig. 3. A Costas array will have $n \cdot (n-1)/2$ distance vectors, so an order 8 Costas array has a total of 28 distance vectors. In the figure, the abscissa is the count of shared distance vectors, and for each number of shared distance vectors and each of 440 Costas arrays, we count the number of other Costas arrays that share exactly this number of shared distance vectors. This gives us 439 numbers for each count of shared distance vectors. In Fig. 3, we plot the mean, mean plus and minus sample variance, sample maximum, and sample minimum.

B. Using the SVD

The concept of using our SVD for Costas array selection is that operating on a vector in Costas array eigenspace will provide a tool for selection of Costas arrays with a given property, such as low cross-correlation with other Costas arrays. For each Costas array vector $\vec{c}$ the profile vectors

$$\overrightarrow{cm} = V^T \cdot \vec{c} \tag{47}$$

$$\overrightarrow{cmi} = IV^T \cdot \vec{c} \tag{48}$$

$$\overrightarrow{cml} = \Lambda^{\#} \cdot V^T \cdot \vec{c} \tag{49}$$

provides a profile of the Costas array by mapping the Costas array vector into the Costas condition eigenspace; the profile is unique for each $\vec{c}$ because V is full rank.

By mapping the Costas array vector into Costas eigenspace, it is expected that attributes of "Costasness" that make the use of Costas arrays appropriate in waveform design will map "clouds" of Costas arrays that are associated with a performance attribute into distributions that align with the axes aligned with the right eigenvectors. Weighting the mapped vectors will then produce more productive metrics for selection of Costas arrays out of the total populiation of all Costas arrays of a given order.

Since V is orthogonal, the first form (47) preserves vector length and dot products in mapping from the Costas array vector space to the profile vector space, so one of the weighted profiles (48) or (49) may be easier to correlate with a system performance metric. But for unweighted vectors mapped using only (47), Euclidean distances are identical to those in raw Costas array vectors, and most advantages will accrue after weighting the elements of the mapped vectors.

The profile vector $\vec{p}$ of (47) provides independent measures of representation of the Costas array vector $\vec{c}$ for each of the eigenvectors; the fact that V is roughly triangular means that the later values that correspond to larger eigenvalues provide measures about the center of the Costas array vector. The scaled profile $\overrightarrow{pn}$ of (49) provides a measure of "Costasness" and may favor Costas arrays that are generated by number-theoretic means such as the Welch generator[2]; weighting $\vec{p}$ instead by the reciprocals of the eigenvalues (except the first, which is zero) may favor "randomness" or spurious Costas arrays that are found only by search and not by any number-theoretic generator.

C. Defining a Model Subset

We see from Fig. 3 that if we ask for less than about 10 shared distance vectors between a Costas array and all the other 439 Costas arrays of order 8, we will get only a few. Here we will perform a search that begins with designating a "pivot" Costas array, and specify a maximum number of shared distance vectors that will be seen with any of the other Costas arrays of order 8. Among the set that is produced using this filter, we will look at cross-correlations and eliminate any that have more than the specified number of difference vectors in common with each other. The result will be a small set of Costas arrays that meet specified limits on cross-correlation.

A set of 14 Costas arrays of order 8 have been found by such a search, with a maximum cross-correlation of 13 distance vectors with any other Costas array of order 8. The number of shared distance vectors between them is shown as a cross-matrix in Table XIII. The first row and column of Table XIII are sequence numbers in the database[6] of 440 Costas arrays of order 8 The starting Costas array for the search is the first one in the database, number 1. Choosing another starting Costas array for the search will produce a different set of Costas arrays.

D. Defining a Metric in Costas Array Eigenspace

A scalar measure of a Costas array can be formulated as a dot product of one of the profile vectors with a reference vector, the Euclidean distance from one of the profile vectors

TABLE XIII
CROSS-CORRELATIONS OF SELECTED SAMPLE

	1	6	9	17	19	27	29	69	91	116	167	171	304	392
1	28	12	13	12	12	11	12	11	10	10	12	11	13	13
6	12	28	11	13	13	13	11	13	10	11	11	13	10	13
9	13	11	28	12	13	13	13	10	8	11	13	11	13	12
17	12	13	12	28	13	11	12	10	11	13	10	13	12	12
19	12	13	13	13	28	12	11	11	11	13	9	13	10	13
27	11	13	13	11	12	28	12	11	11	10	13	8	12	13
29	12	11	13	12	11	12	28	11	13	13	11	13	12	13
69	11	13	10	10	11	11	11	28	13	13	11	12	11	12
91	10	10	8	11	11	11	13	13	28	12	13	13	11	12
116	10	11	11	13	13	10	13	13	12	28	12	11	12	11
167	12	11	13	10	9	13	11	11	13	12	28	11	12	12
171	11	13	11	13	13	8	13	12	13	11	11	28	13	11
304	13	10	13	12	10	12	12	11	11	12	12	13	28	11
392	13	13	12	12	13	13	13	12	12	11	12	11	11	28

TABLE XIV
METRICS FOR POLYGON, CUBE, AND SPHERE IN n-SPACE

Type	Distance Metric
Hypercube	$\text{Max}_{1\le i\le n}\left(\frac{\lvert v_i - v_{r,i}\rvert}{\sqrt{\sigma_i^2+\sigma_{r,i}^2}}\right)$
Hypersphere (Mahalanobis distance)	$(\vec{v}-\vec{v}_r)^T \cdot R^{-1} \cdot (\vec{v}-\vec{v}_r)$
Hyperpolygon	$\sum_{i=1}^{n} \frac{\lvert v_i - v_{r,i}\rvert}{\sqrt{\sigma_i^2+\sigma_{r,i}^2}}$

TABLE XV
ENCLOSED VOLUMES OF POLYGON, CUBE, AND SPHERE IN n-SPACE

Type of Metric	Enclosed Volume
Hypercube	2^n
Hypersphere (Mahalanobis distance)	$\frac{\pi^{n/2}}{\Gamma(\frac{n}{2}+1)}$
Hyperpolygon	$\frac{2^n}{\Gamma(n+1)}$

to a reference vector, or a quadratic form using the two vectors with a weighting matrix. Simply summing the elements is a dot product of the profile vector $\vec{p}$ with a reference vector $\vec{1}$ of all 1s, for example. The Euclidean length is the distance of a vector to the origin $\vec{0}$, as another example. A particular system's performance criteria may define more appropriate reference vectors.

In order to leverage the key factor that we are working in Costas eigenspace, we can weight the Costas array vectors as mapped by (47), (48) or (49). Equation (48) or (49) do provide weighting by squared scaling or eigenvalue, respectively.

Data fusion techniques show that the metric used in multivariate spaces has a huge effect on effectiveness of selection of volume in the space. The three most popular are shown in Table XIV. In the table, R is a covariance matrix of the difference between the data vector $\vec{v}$ and the reference vector $\vec{v}_r$.

The volumes of a hypercube with faces a distance 1 from the origin, a hypersphere of radius 1, and a hyperpolygon with its vertices on the hypersphere of radius 1, are given by Table XV. Note that the hypercube and hyperpolygon metrics physical units are linear in distance, but that the hyperpolygon metric is given is quadratic in distance.

The ratios of the volumes contained in spaces defined by these metrics is illustrated versus the number of dimension in

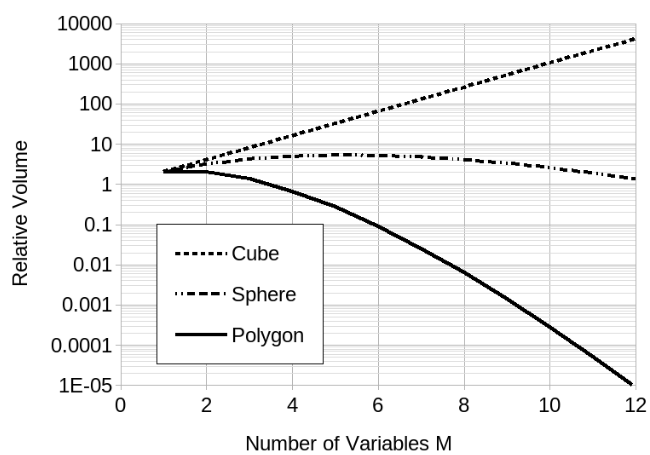

Fig. 4. Comparing Volumes of Hypercube, Hypersphere, and Hyperpolygon.

Fig. 4. Even at 8 dimensions, the relative volumes defined by the three metrics differ by orders of magnitude.

The distance metric we use in our example is weighting, including occasional zeros, based on sample variances of Euclidean distance of the vector in Costas eigenspace from the sample mean from the entire database and the model group.

$$\overrightarrow{cmw}_i = \vec{v}_{W,i} \cdot \overrightarrow{cm}_i,\ 1 \le i \le n \quad (50)$$

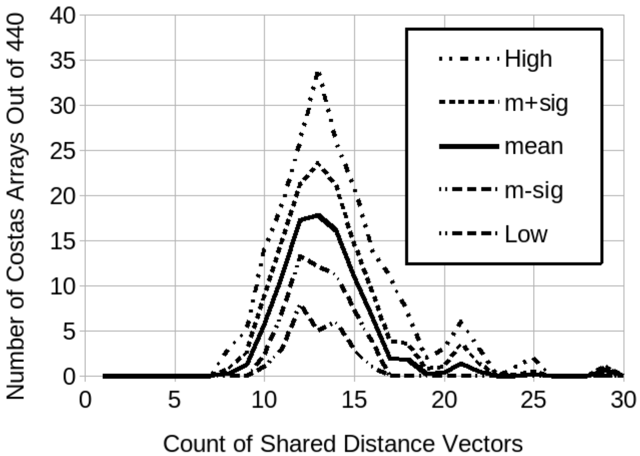

Fig. 5. Cross-Interference of Costas Arrays Selected using SVD.

where $\overrightarrow{cm}$ is the Costas array vector mapped to Costas eigenspace as in (47), $\overrightarrow{v}_W$ is is our weighting vector that may include eigevevalues, sample variances, and zero values, and $\overrightarrow{cmw}$ is the mapped Costas array vector with its elements weighted by the elements of $\overrightarrow{v}_W$. The weighting used in the example below is

- The weight for each eigenspace component is the ratio of the sample variances from the entire database to the sample variance for the model group for that component.
- Eigenvalue weighting is added in each of three powers: -5, $+1/2$, and $+4$.
- When the variance of the model group is greater than that of the entire database, the weight for that eigenspace component is set to zero.

Variance weighting is similar to Mahalanobis distance[18] except that cross-correlations are not included.

E. Defining a Second Set for the Waveform

In our example, we map our first group of desirable Costas arrays defined as explained in Section VI-C. Given the weightings $\overrightarrow{v}_W$ as defined above, Costas arrays are added to a second set when

- The Euclidean distance from $\overrightarrow{vmw}$ to the similarly mapped ensemble mean of the model group is less than 0.8 times the sample variance in excess of the sample mean, and
- The Euclidean distance from $\overrightarrow{vmw}$ to the similarly mapped ensemble mean of the model group is greater than 0.4 times the sample variance below the sample mean.

The selection process scans the entire database for order 8 and adds Costas arrays to the new set that pass the criteria for each eigenvalue weighting, then it applies th selection process critera to the new set and culls Costas arrays from the set that fail the criteria for any one of the eigenvalue weightings.

A sum of 95 Costas arrays passes the selection process. The statistics are shown in Fig. 5. Some improvement over that seen in Fig. 3 is seen, but too many in this group show higher numbers of shared distance vectors than would be acceptable in a system design.

F. Effective Selection of Volume in Costas Vector Space

Counts of shared distance vectors for a first-cut selection using Euclidean distance with weighting modified from (49) is shown in Fig. 5. From the first cut that uses elementary Cartesian distance for selection of volume in Costas vector space shows that

- Selection in Costas eigenspace by elementary methods shows promise for controlling the number of shared distance vectors.
- The Mahalanobis distance must be used to select regions in Costas array space.

An operational real-time Costas array selection will use methods that include but are not limited to

- Mahalanobis distance in n-space as the metric for assessing mapped weighted Costas array vectors; this will account for the model ”cloud” alignment and shape with the right eigenvectors.
- The distributions of distance metrics for the entire Database for a given order, and for the model distribution, form the basis for the weighting vector $\overrightarrow{v}_W$ to maximize the separation between the distributions in n-space.
- Use selection criteria based on the distributions of the metric for the model set versus the full database set in place of the ”hollow ellipsoid shell” critera of Section VI-E.

The Mahalanobis distance uses the covariance matrix between the populations being compared:

$$D_M^2(\overrightarrow{x}) = (\overrightarrow{x} - \overrightarrow{m})^T \cdot S^{-1} \cdot (\overrightarrow{x} - \overrightarrow{m}) \qquad (51)$$

where S is the covariance of $(\overrightarrow{x} - \overrightarrow{m})$. For all the Costas array vectors of a given order n, the mean

$$\overrightarrow{m} = \frac{1}{C(n)} \sum_{k=0}^{C(n)} \overrightarrow{x}_k$$

is always a set of constants equal to the mid-points between the maximum and minimum row indices (1 and n, or 0 and $(n-1)$). In Costas array eigenspace this maps to a zero vector because a vector of all zeros corresponds to the zero eigenvalue of the right eigenvector matrix. But, when a subset is selected for a particular criteria as in section VI-C then the mean is, in general, nonzero. Thus the sample covariance P_C for the entire set of a given order n with $C(n)$ elements is

$$P_C = \frac{1}{C(n)} \sum_{k=0}^{C(n)} \overrightarrow{c}_k \cdot \overrightarrow{c}_k^T.$$

When there is a subset S with N_S elements and a nonzero mean m_S does exist,

$$\overrightarrow{m}_S = \frac{1}{N_S} \sum_{k \in S} \overrightarrow{c}_k$$

the sample variance is

$$P_S = \frac{1}{N_S - 1} \left(\sum_{k \in S} \overrightarrow{c}_k \cdot \overrightarrow{c}_k^T - N_S \cdot \overrightarrow{m}_S \cdot \overrightarrow{m}_S^T \right)$$

and the covariance of $\overrightarrow{m}_S$ is

$$P_{mS} = \frac{1}{N_S} \cdot P_S.$$

The covariance of $(\overrightarrow{c}_k - \overrightarrow{m}_S)$ is, when $k \notin S$,

$$P_{k,S} = \frac{\sum_{k \notin S} \overrightarrow{c}_k \cdot \overrightarrow{c}_k^T - (C(n) - N_S) \cdot \overrightarrow{m}_{nS} \cdot \overrightarrow{m}_{nS}^T}{C(n) - N_S - 1} + P_{mS}$$

where

$$\begin{aligned} \overrightarrow{m}_n S &= \frac{1}{C(n) - N_S} \cdot \sum_{k \notin S} \overrightarrow{c}_k \\ &= \frac{1}{C(n) - N_S} \cdot (C(n) \cdot \overrightarrow{m} - N_S \cdot \overrightarrow{m}_S) \\ &= -\frac{N_S}{C(n) - N_S} \cdot \overrightarrow{m}_S. \end{aligned}$$

In the Mahalanobis distance, the covariance matrix S is the covariance of $(\overrightarrow{x} - \overrightarrow{m})$, which is varies according to whether $\overrightarrow{c}_k$ is in S or not:

$$P(\overrightarrow{x}_k, \overrightarrow{m}_S) \begin{cases} = P_S, \ k \in S \\ = P_{k,S}, \ k \notin S. \end{cases} \tag{52}$$

VII. Computer Resources

A. *Original Basis was Numerical SVD to Order* 35

This work was begun with a numerical SVD to order 35 using the methods of [16] followed by analysis using the methods of Appendix B. Order 35 was selected as the highest order for numerical SVD because

- The dataset to order 35 is more than sufficient to support the formulation and validation of closed forms,
- The run times were becoming an obstacle to progress as orders increased,
- The word length requirements required by these methods require extended precision for higher orders, exacerbating the run time problem.

The numerical SVD computations with production of the integer database for order 35 was accomplished in a run of a few weeks.

B. *Database File Sizes and Word Lengths*

A curve of database file sizes versus order is shown as Fig. 1. The file for order 100 is about 85 MB in size. The database was computed using the equations given in Sections IV because the times needed for numerical SVD followed by finding the scaling that produces integers as explained above in Appendix B Sections B-C and B-D become impractical for larger orders. The methods of Section IV allowed the database to be computed to order 100. The equation fit also shown in Fig. 1 is the approximation for $n > 30$

$$Size(n) \approx \left(\frac{n}{0.83}\right)^{3.82}. \tag{53}$$

These are individual file sizes; total database file size to order n will increase approximately as n to about the 5^{th} power. Total full SVD database uncompressed size to order 100 is about 1.8 GB.

Some terms in the vector of left eigenvalue squared values as given by (13a) or (13b) will overflow a 32-bit integer for order 46 and above, even with common factors removed from the eigenvector elements, so a 64-bit integer must be used for these values in working with higher orders.

C. *Checking an Existing Database Against New Computation*

Checking an existing database by new computation requires about the same resources as creating a new database, because a new database is created for each order to check an existing data file.

D. *Generating New Database Files*

The time for computing new database files from low order to a given order $n > 50$ on a single 64-bit AMD core at 4 GHz is

$$Time(n) \approx \left(\frac{n}{21.95}\right)^{8.25}.$$

The number dividing n is approximately the order to which run time is extrapolated to be about 1 second. Run times to orders below about 50 are higher than those given by the estimate because overhead due to I/O and housekeeping becomes large enough to impact timings for smaller orders.

E. *Checking an Existing Database by Reconstructing* A

This tests uses IVL and IV with the scale n factors to reconstruct the original constraint matrix A, which is subtracted from the result. The maximum peak error versus order is shown in Fig. 2.

This check also looks for negative values for stored values of squared eigenvector lengths, squared eigenvalues, and scale factors. A negative value here is not mathematically possible and is an indicator of overflow during calculations or internal computer representations of those values.

Checking a database by reconstructing A is far quicker, taking about 16 minutes to order 100 on a single 4 GHz core. Total run time to a given order is in polynomial time, proportional to about the sixth power of the order. The time to check database files up to order $n > 50$ is approximated by

$$Time(n) \approx \left(\frac{n}{30}\right)^{5.75}.$$

The database itself can provides data for the user to validate the SVD by implementing

$$A_{Reconst} = \sum_{j=1}^{n} \overrightarrow{IVL_j} * \left(\frac{1}{SF_j}\right) * \overrightarrow{IVR_j}^T. \tag{54}$$

The matrix A is included in the database to order 100, almost doubling the file sizes, specifically to allow checking by this method, as is the vector of scale factors.

The floating point VL and V matrices are obtained by dividing the columns of each matrix by square roots of the appropriate squared lengths that are stored in the second and third row of each database file. This is done for both matrices,

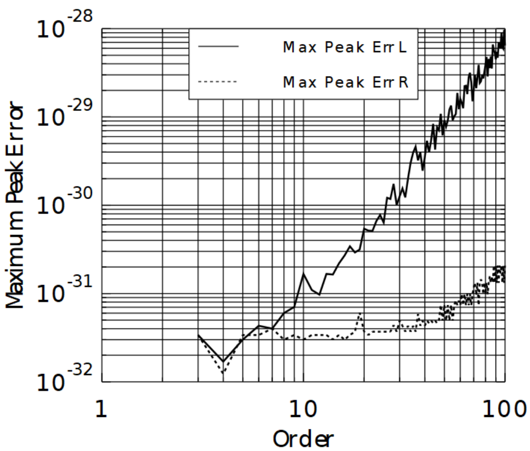

Fig. 6. Numerical Accuracy of VL and V.

an n by n identity matrix is computed by $VL^T \cdot VL$ and $V^T \cdot V$ and the maximum peak error for each is found. The result is shown in Fig. 6.

Since VL is an m by n matrix where m is given by (10), which grows to $166,650$ for $n = 100$, the numerical error in computing $VL^T \cdot VL$ increases approximately as n^3, while the numerical error in computing $V^T \cdot V$ increases more slowly with increasing n.

Time to compute to a given order is approximated by

$$Time(n) \approx \left(\frac{n}{30}\right)^{5.77}.$$

F. Database of Right Eigenvectors and Squared Eigenvalues

Most of the computational resources for large orders is computation of the left eigenvectors. The left eigenvectors are of interest mathematically but only the right eigenvectors and the eigenvalues are used to find a "profile" vector from a Costas array vector; see Section VI. A second database is available on IEEE Dataport [12] of the right eigenvectors and the squared eigenvalues. Compute time to order N is approximated by

$$Time(n) \approx \left(\frac{n}{45.3}\right)^{2.93}.$$

This database includes the common factors removed from the right eigenvectors, so that these can be restored for use in (35) if the user does need the left eigenvectors for very large order.

The closed forms of Section IV enabled this database.

VIII. Conclusion

Four contributions are documented here:

1) A particular formulation of a matrix representation that is an augmentation of the Costas condition matrix has been found, $A \cdot \vec{c} = \vec{b}$, for which the eigenvalues are squares of integers and the elements of the eigenvectors can be scaled to integers. The augmented formulation is presented in Section III.
2) Simple expressions have been found for the eigenvalues and right eigenvectors; these are presented in Section IV.
3) Since the left eigenvectors are linked to the original matrix A, and the order of the rows of A is empirical, a simple equation for rows of the left eigenvector matrix in terms of the matrix A and the other results was found and is presented as (37) in Section IV-D.
4) A simple expression for finding the numerators of the rational forms for elements of the left eigenvectors is found and given as (39) in Section IV-D.
5) A database of these Costas condition matrices and their eigenvalues and eigenvectors as ratios of integers has been computed and is provided on IEEE Dataport[12]. The database is presented in Section V. Full SVD to order 100 and shorter files without the left eigenvectors are available to order 1030.

These are substantiated by proofs.

- For a given order n of Costas arrays, the rank of A is $(n-1)$, proven in Theorem 3.2.
- The singular value decomposition of A is given in terms of the squares of its eigenvalues and right eigenvectors in Section IV; proven in Theorem 5.1.
- The left eigenvectors are given in normalized and in scaled-to-integers form in Section IV-D; proven in Theorem 5.1. Similar results are given for the matrix $[A^T \cdot A]$ in Section V-A2.
- The SVD is presented for orders 3 through 100 as eigenvector matrices and squared eigenvectors in a database in IEEE DataPort[12]. Data is provided as integers to support any accuracy that a user may wish to implement, and the matrix A and scale factors for its reconstruction using 17 to support user verification of their own software. Proven in Theorem 5.1.
- The SVD is presented in another database on the same IEEE DataPort page that omits the left eigenvectors and matrix A for orders through 1030. Proven in Theorem 5.1.
- Proof of validity and uniqueness of the database is provided in Section V-A as Theorem 5.1.
- Proof of elementary recursive construction method for right eigenvector matrices in Section V-E as Theorem 5.2.

The proof of the results provided in Section III-C begins with results from classical numeric singular value decomposition to order 35, and applies to any order. The database provides usable results to order 1030 and Section III-C provides simple polynomial equations that provide these results for any order. The linkage of validation is

- Numerical SVD of the matrix A as given in Section III for orders 3 through 35.
- Production of closed-form SVD to order 100 using the methods of Section IV and IV-D and validation of results against the numerical SVD to order 35, reconstruction of the matrix A to order 100, and verification of orthogonality of the eigenvector matrices to order 100.
- Production of closed-form right eigenvectors and squared eigenvalues to order 1030 using the methods of Section

IV, and verification against the full SVD database to order 100.

The maximum peak errors from validation using the full SVD database to order 100 in quad precision are presented in Fig. 6; file sizes are given in Fig. 1, along with the fit given by (53).

APPENDIX A
SUMS OF POWERS

Sums of powers are given by Bernoulli polynomials in Abramowitz & Stegun ([19] paragraph 23.1.4 p. 804), and in DLMF ([20] paragraph 24.4.7 p. 589, and online at [21] Chapter 24.4(iii) Bernoulli Polynomials/Elementary Properties/Sums of Powers, paragraph 24.4.7),

$$\sum_{k=1}^{n} k^p = \frac{B_{p+1}(n+1) - B_{p+1}}{p+1} \tag{55}$$

where $B_n(x)$ is a Bernoulli polynomial and B_n without an argument is the constant term, $B_n(0)$, defined as a Bernoulli number. In particular,

$$\sum_{k=1}^{n} k = \frac{B_2(n+1) - B_2}{2} = \frac{n \cdot (n+1)}{2} \tag{56a}$$

$$\sum_{k=1}^{n} k^2 = \frac{B_3(n+1)}{3} = \frac{n \cdot (n+1) \cdot (2n+1)}{6} \tag{56b}$$

$$\sum_{k=1}^{n} k^3 = \frac{B_4(n+1) - B_4}{4} = \frac{n^2 \cdot (n+1)^2}{4} \tag{56c}$$

$$\sum_{k=1}^{n} k^4 = \frac{n \cdot (n+1) \cdot (2n+1) \cdot (3n^2 + 3n - 1)}{30} \tag{56d}$$

$$\sum_{k=1}^{n} k^5 = \frac{(n+1)^2 \cdot n^2 \cdot (2n^2 + 2n - 1)}{12}. \tag{56e}$$

A general relationship as well as a recursion for $B_k(n)$ is given by [21] paragraph 24.5.11,

$$\sum_{k=0}^{n} \binom{n+1}{k} \cdot B_k(x) = (n+1) \cdot x^n.$$

APPENDIX B
FINDING INTEGER NUMERATORS AND DENOMINATORS FROM THEIR RATIOS

A. Expanding Real Numbers in a Simple Continued Fraction

We will begin by defining notation for continued fractions, following [19] Section 3.10.1 p. 19,

$$f = b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + ...}}} \tag{57}$$

where a_i and b_i are the coefficients of the continued fraction. A terminating continued fraction, or the n^{th} convergent of a continued fraction f_n is a continued fraction for which a_{n+1} is 0 or taken as 0,

$$f_n = b_0 + \cfrac{a_1}{b_1 + ... + \cfrac{a_n}{b_n}} = \frac{A_n}{B_n} \tag{58}$$

where A_n and B_n, the partial numerator and partial denominator, respectively, are polynomials in the coefficients.

The sequence of partial numerators and denominators, and thus convergents of a continued fraction, can be found recursively using [19]

$$\begin{bmatrix} A_0 & A_{-1} \\ B_0 & B_{-1} \end{bmatrix} = \begin{bmatrix} b_0 & 1 \\ 1 & 0 \end{bmatrix} \tag{59}$$

to initialize the recursion

$$\begin{bmatrix} A_i & A_{i-1} \\ B_i & B_{i-1} \end{bmatrix} = \begin{bmatrix} A_{i-1} & A_{i-2} \\ B_{i-1} & B_{i-2} \end{bmatrix} \cdot \begin{bmatrix} b_i & 1 \\ a_i & 0 \end{bmatrix}. \tag{60}$$

Equations (59) and (60) are easily proven from (58) by induction.

When the coefficients a_i or b_i in (58), (59) and (60) are polynomials in an independent variable, particularly a complex variable, the covergents form the basis for the analytic theory of continued fractions, most prolifically developed in [22]. Formulas for analytic continued fractions are given in most chapters of [19], [20] and [21] etc. that are useful for computing transcendental functions in the complex plane.

When the a_i are all one and the b_i are all positive integers, (57) is called a simple continued fraction. A real number r can be expanded into a simple continued fraction by the recursion

$$b_0 = \lfloor r \rfloor,\ r_1 = r - b_0 < 1 \tag{61}$$

$$b_1 = \left\lfloor \frac{1}{r_1} \right\rfloor,\ r_2 = \frac{1}{r_1} - b_1 < 1$$

$$...$$

$$b_i = \left\lfloor \frac{1}{r_i} \right\rfloor,\ r_{i+1} = \frac{1}{r_i} - b_i < 1 \tag{62}$$

$$...$$

which has several interesting properties ([23] Chapter XI Approximation of Irrationals by Rationals, pp 198-217):

- The partial numerators and denominators A_i and B_i form an increasing sequence of integers.
- The difference between r and A_i/B_i is always less than $1/B_i^2$ ([24] Theorem 3.8 p. 74, [23] Theorem 164 p. 176).
- When r is rational, the continued fraction always terminates with some $r_{N+1} = 0$.

B. Euclid's Method for Finding the GCD of Two Integers

Euclid's method for finding the greatest common multiple of two numbers ([24] p. 17, [23] pp 172-174 and 231-232) is implicit in (61) and (62) through the integer quotient relationship

$$\frac{ix_{i-1}}{ix_i} = b_i + \frac{ix_{i+1}}{ix_i},\ ix_{i+1} < ix_i. \tag{63}$$

The analogy of (63) with (62) can be seen by writing (62) as

$$\frac{1}{r_i} = b_i + r_{i+1},\ 0 \le r_{i+1} < 1.$$

If r_i is rational ix_i/ix_{i-1}, then the integer division equation multiplied through by the denominator ix_i,

$$ix_{i-1} = b_i \cdot ix_i + ix_{i+1},\ ix_{i+1} < ix_i$$

shows that any factors shared by ix_{i-1} and ix_i are also shared by ix_{i+1}, and that when the process terminates as it must for rational numbers when the last $ix_{N+1} = 0$, the ix_N (or b_N if $ix_N = 1$) divides every ix_i.

Equation (60) shows that A_i and B_i, for $a_n = 1$, are polynomials in the b_k, $0 \leq k \leq i$ and that one of the polynomials A_i or B_i always has a constant term of 1, so that partial numerator or denominator cannot be divisible by any b_k for which $a_k > 0$ and $GCD(a_k, b_k) = 1$ for $1 \leq k \leq n$. Thus B_N is not divisible by b_N, so that the GCD of A_i and B_i is always 1 until $a_N = 0$.

C. Approximating a Real Number with the Ratio of Two Integers

The accuracy of the value of a convergent of the continued fraction, as expressed by the ratio of integers A_i/B_i, increases as i increases. Quantitatively ([23] Theorem 171 p. 180),

$$\left| r - \frac{A_i}{B_i} \right| < \frac{1}{B_{i-1} \cdot B_i} < \frac{1}{B_i^2}. \tag{64}$$

Equation (64) shows that we can use (61) and (62) with (59) and (60) to define a sequence of ratios of integers A_i and B_i to approximate a real number r to any degree of accuracy as A_n/B_n. We mention that (64) is not the tightest possible bound when expanding irrational numbers; see [23] Theorems 193-195 pp 209-211. Using any degree of precision of floating point numbers, there exists a value of n for which

$$\log_2(r \cdot B_{n-1} \cdot B_n) > \text{available precision in bits}, \tag{65}$$

beyond which (62) is operating on random bits produced by numerical errors. This practical issue in finding the correct ratio of integers from a rational number is given in floating point notation is addressed by ending (62) when a value of b_n is encountered which is so large that (65) is met; this is recognized by noting when $log_2(B_{i-1} \cdot B_i) \sim log_2(B_i^2)$ approaches or exceeds the number of bits of accuracy of the mantissa in the floating point that is in use. This is most simply recognized by noting a very small r_{n+1} in (62). When this is not clear in the computation, the simplest verification is to recompute with higher precision and duplicate the results.

D. Common Factors in the Elements of the Eigenvectors

The integers from scaled eigenvector elements as initially produced by SVD and rational approximation methods often have common factors. In particular, the even-numbered eigenvectors for even order matrices, given by (32), are all divisible by 3 whenever $(n - j + 2)$ is divisible by 3, which clearly happens every third eigenvector, and in half of those occurrences, the even-numbered eigenvectors are divisible by 6. Our approach in developing the equations for the elements of the eigenvectors is to use numerical methods to high accuracy to provide the eigenvector matrices in binary floating point, then to use a method based on Euclid's method ([24] p. 17, [23] pp 172-174 and 231-232) to provide a sequence of rational approximations to ratios of elements; this sequence is terminated when the continued fraction terminates, as evidenced within the accuracy of the arithmetic used.

This method inherently produces a ratio of integers that have no common divisors other than 1. The scaled eigenvectors with integral elements and the common divisor that normalizes the eigenvector to unity length, when only the direction cosines are known, follows here. We denote the order as n. We are given a set of real numbers r_i, $1 \leq i \leq n$, that are rational multiples of one another, and whose sum of squares is 1,

$$\frac{r_i}{r_1} = \frac{a_i}{b_i} \tag{66a}$$

$$\sum_{i=1}^{n} r_i^2 = 1 \tag{66b}$$

where a_i and b_i are integers. We substitute for r_i, $i > 1$ in the second form,

$$r_1^2 \cdot \left(1 + \sum_{i=2}^{n} \left(\frac{a_i}{b_i} \right)^2 \right) = 1$$

and left-multiply by the product of all the b_i, $i > 1$,

$$B = \prod_{i=1}^{n} b_i$$

to obtain the result

$$r_1^2 \cdot \left(B^2 + \sum_{i=2}^{n} \left(\frac{B}{b_i} \right)^2 \cdot a_i^2 \right) = B^2$$

we find that we can solve for r_1 as

$$r_1 = \frac{B}{\sqrt{B^2 + \sum_{i=2}^{n} \left(\frac{B}{b_i} \right)^2 \cdot a_i^2}}.$$

Given that $r_i = a_i/b_i$, the rest of the r_i are

$$r_i = \frac{\frac{B}{b_i} \cdot a_i}{\sqrt{B^2 + \sum_{i=2}^{n} \left(\frac{B}{b_i} \right)^2 \cdot a_i^2}}.$$

Thus the r_i can be represented by the sequence of integers ir_i, and the r_i can be recovered by dividing each ir_i by a single denominator for each eigenvector,

$$ir_1 = B \tag{67a}$$

$$ir_i = \left(\frac{B}{b_i} \right) \cdot a_i, \; i > 1 \tag{67b}$$

$$r_i = \frac{ir_i}{D} \tag{67c}$$

$$D^2 = \sum_{i=1}^{n} ir_i^2. \tag{67d}$$

To produce eigenvectors with integer elements that equal those obtained by these numerical methods, the common factors must be eliminated when they occur. Some can be avoided by using the least common multiple (LCM) of the b_i instead of simply the product of all the b_o in (67) but the factors of a_i in the ir_i for $i > 1$ allows for more common factors; these must be eliminated when they occur to get results that match the numerical results. This is done for the database by using Euclid's method on all the ir_i to obtain the greatest common divisor (GCD) that applies to all elements, then dividing the vector by this GCD, and dividing the separately-stored sum of squared elements by the square of this GCD.

The elimination of common factors in all the eigenvectors in the database has the advantage of producing the smallest possible integers. The absence of common factors is important in some possible uses of the database, and minimizing the magnitudes of the numbers can help avoid numerical problems in its use with large orders.

The Fortran program module included with the database has a procedure that removes common factors from each eigenvector. The default is for this to be done, but it can be omitted if the user wants to observe the behavior of the eigenvectors as given directly by the equations in Sections IV-C and IV-B. The eigenvectors as given in the database do have common factors removed.

ACKNOWLEDGMENTS

This work was produced in several efforts beginning in 2012, when most of it was done. At that time, publication was not considered because the author did not consider results to that date a milestone of significance to the community. Revisiting the work in 2014 produced the left eigenvector matrices directly and provided more accurate IVL, which was used to validate the earlier SVD and integer data to order 35. In late 2019, a revisit produced the simpler forms of (15), and the insight documented here in (38) and (39), and the availability of IEEE Dataport led to the completion of this work.

The author owes a debt of gratitude to the reviewers, whose comments provided significant insight into communication issues and led to a change in the presentation of the mathematical concepts presented in the presentation of the matrix representation of the Costas condition. In particular, the reviewers pointed out that the paper was incomplete without proofs, and adding proofs required incorporating results for the matrix $[A \cdot A^T]$ that are also interesting. In addition, comments on specific items resulted in better accuracy in the presentation.

The author would like to thank Jon Russo, Keith Erickson, Greg Coxson, Bill Correll, Chris Swanson and others too numerous to name here for discussions and interactions regarding Costas arrays over the years. And, the landmark paper[2] by Solomon Golomb and Herbert Taylor has provided a pivot point for many of us in our work in Costas arrays. Guang Gong[25][5] and Herbert Taylor's later work[26], including the final generators for unbounded order[3], the Mac Challenge for December 1999[27] which, with the work of Scott Rickard and Konstantinos Drakakis and others at University College Dublin, Oscar Moreno of University of Puerto Rico Río Piedras Campus, and many others have stimulated Costas array research and applications in communications and other areas.

We regret the passing of Solomon Golomb on July 14, 2015, and the passing of Oscar Moreno on May 1, 2016.

REFERENCES

[1] J. P. Costas, "A study of detection waveforms having nearly ideal range-Doppler ambiguity properties," *Proc. IEEE*, vol. 72, pp. 996–1009, 1984.
[2] S. Golomb and H. Taylor, "Constructions and properties of Costas arrays," *Proc. IEEE*, vol. 72, pp. 1143–1163, 1984.
[3] S. W. Golomb, "The T_4 and G_4 constructions for Costas arrays," *IEEE Trans. Inf. Theory*, vol. 38, no. 4, pp. 1404–1406, 1992.
[4] J. K. Beard, J. C. Russo, K. G. Erickson, M. C. Monteleone, and M. T. Wright, "Costas array generation and search methodology," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 2, pp. 522–538 DOI: 10.1109/TAES.2007.4 285 351, April 2007.
[5] S. W. Golomb and G. Gong, "The status of Costas arrays," *IEEE Trans. Inf. Theory*, vol. 53, no. 11, pp. 4260–4265, November 2007.
[6] J. K. Beard, "Costas arrays and enumeration to order 1030," IEEE Dataport, 2017. [Online]. Available: http://dx.doi.org/10.21227/H21P42
[7] N. Levanon and E. Mozeson, *Radar Signals*, 1st ed. ISBN 978-0-471-47378-7: John Wiley & Sons, Inc., 2004.
[8] P. E. Pace and C. Y. Ng, "Costas CW frequency hopping radar waveform: peak sidelobe improvement using Golay complementary sequences," *Electronics Letters*, vol. 46, no. 2, pp. 169–170, January 2010.
[9] B. Correll, J. K. Beard, and C. N. Swanson, "Costas array waveforms for closely spaced target detection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 2, pp. 1045–1076, 2020.
[10] E. L. Titlebaum, "Time-frequency hop signals part I: Coding based upon the theory of linear congruences," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 17, no. 4, pp. 490–493, July 1981.
[11] M. Sterling, "A linear dependence based construction related to Costas arrays," in *Proceedings of the 2016 ACM International Conference on Communication and Information Systems (ICCIS2016)*, Bangkok, Thailand, Dec 2016, pp. 21–14 DOI 10.1145/3 023 924.3 023 934.
[12] J. Beard. (2020) Database of singular value decompositions of matrix representations of the costas condition. [Online]. Available: http://dx.doi.org/10.21227/h498-px29
[13] J. P. Costas, "Medium constraints on sonar design and performance," GE Co., Technical Report Class 1 Rep. R65EMH33, 1965.
[14] L. Barker, K. Drakakis, and S. Rickard, "On the complexity of the verification of the Costas property," *Proc. IEEE*, vol. 97, no. 3, pp. 586–593, March 2009.
[15] K. Drakakis, "Some results on the degrees of freedom of Costas arrays," in *Proceedings of Forty-Fourth Conference on Information Sciences and Systems*, 2010, pp. 1–5.
[16] J. Wilkinson and E. C. Reinsch, *Handbook for Automatic Computation, Vol. 2: Linear Algebra*, 1st ed. ISBN 0-387-05414-6 (New York), 3-540-05414-6 (Berlin)7: Springer-Verlag, 1971.
[17] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," in *Numerische Mathematik*, vol. 14, no. 5, April 1970, pp. 403–420.
[18] P. C. Mahalanobis, "On the generalized distance in statistics," *Proceedings of the National Institute of Sciences of India*, vol. 2, no. 1, pp. 49–55, 1936. [Online]. Available: http://insa.nic.in/writereaddata/UpLoadedFiles/PINSA/Vol02_1936_1_Art05.pdf
[19] M. Abramowitz and E. Irene Stegun, *Handbook of Mathematical Functions*, ser. 55, M. Abramowitz and I. Stegun, Eds. U.S. Government Printing Office, National Bureau of Standards Applied Mathematics Series, 1972, tenth Printing, December 1972, with corrections. [Online]. Available: http://people.math.sfu.ca/~cbm/aands/intro.htm
[20] F. W. J. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, Eds., *NIST Handbook of Mathematical Functions*. Cambridge University Press, 2020.
[21] Nist digital library of mathematical functions. [Online]. Available: https://dlmf.nist.gov/
[22] H. Wall, *Analytic Theory of Continued Fractions*, ser. AMS Chelsea Publishing Series. American Mathematical Society, 2000. [Online]. Available: https://books.google.com/books?id=eE8PpgM3ucMC
[23] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers*, 5th ed. Oxford: Oxford University Press, 1979.
[24] C. D. Olds, *Continued Fractions*, 1st ed. Library of Congress 61-12185: Random House New Mathematical Libary, Volume 9, 1963.
[25] G. Gong. Electrical and computer engineering. University of Waterloo. [Online]. Available: https://uwaterloo.ca/electrical-computer-engineering/profile/ggong
[26] H. Taylor, *Handbook of Combinatorial Designs*, 2nd ed. Chapman & Hall/CRC, 2007, ch. Chapter 9, Costas Arrays, pp. 357–361.
[27] (1999, December) December 1999 programmer's challenge: Costas arrays. Mac Tech. [Online]. Available: http://www.mactech.com/progchallenge/9912Challenge.html

James K. Beard (M’64 LM’04 LSM’05), ORCID 0000-0003-4430-8704 was born in Austin, TX in 1939. He received a BS degree form 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. Employers include precursors or current divisions of Northrop Grumman, Raytheon, and Lockheed Martin, most recently Lockheed Martin in Moorestown, NJ. He is currently an individual consultant based near Philadelphia, PA. He is the author of a number of papers and a boot, ”‘The FFT in the 21st Century,”’ (Springer-Verlag, 2003). Dr. Beard in the inventor of record for US Patent No. 6,870,501 awarded March, 2005, ”‘Digital Radio Frequency Tag,”’ assigned to Raytheon Company. Current research interests include system engineering solutions to homeland defense issues, estimation and decision theory, radar and communications concept and waveform design, BMC3I, and digital radar concepts. He was Publications Chairman for FUSION2005. 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 completed his PH. D. under an NSF fellowship.