Difference Bases and Sparse Sensor Arrays

Darel A. Linebarger, Member, IEEE, and Ioannis G. Tollis, Member, IEEE

Abstract—Difference bases are discussed and their relevance to sensor array design is described. Several new analytical difference base structures that result in near optimal low-redundancy sensor arrays are introduced. Algorithms are also presented for efficiently obtaining sparse sensor arrays and/or difference bases. Lastly, new bounds, related to arrays that have both redundancies and holes in their coarray are presented. Also, some extensions to the idea of difference bases that may yield useful results for sensor array design are discussed.

Index Terms—Sensor array, restricted difference base, coarray, minimum redundancy array, minimum hole array.

I. INTRODUCTION

In situations where one must obtain maximum spatial resolution from a limited number of sensors, array configurations known as minimum redundancy arrays [1]–[4] or minimum hole arrays [5] are often employed. For a given number of sensors, these arrays allow for increased aperture by reducing the number of redundant spacings in the array.

The justification for reducing redundancy in sensor arrays arises from assumptions usually made regarding the signals to be monitored with the array. Generally, the acoustic field containing the array and the propagating signals of interest is assumed to be a wide-sense stationary random field. Thus, correlation between various points in the field is a function only of the separation between the points. The quantities we are interested in are the bearings of these propagating signals. Also, these bearings are known to be equivalent to spatial frequency [6]. We assume that all propagating signals are well approximated by plane waves.

If the sensor array is linear, the sensor separations can be represented as a set of numbers—the intersensor separations. Furthermore, it is common to require the sensor spacings to be integer multiples of some fundamental distance and thus the sensor separations can be represented by these integers. Therefore, throughout this paper we will assume that the sensors lie on the marks of a Cartesian grid.

An array of sensors samples the field that it lies in. In our case, we assume this field to be wide sense stationary, and since we are actually interested in a spatial power spectrum density, we can focus on the manner in which the array samples the spatial correlation function of the field as opposed to how it samples the field itself. The spatial power spectrum density will contain information regarding the direction of propagation for any signals in the field. For an introduction to the theory of spatial spectral estimation see [7, ch. 6].

Based on the assumption that one is primarily interested in how an array samples the spatial correlation function, jargon has been coined.

Manuscript received August 19, 1991.

This work was supported in part by the Texas Advanced Technology Program Project No. 009741-038, Texas Advanced Research Program Project No. 3972, and Texas Instruments. This work was presented in part at the Twenty-Fourth Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, Nov. 1990.

The authors are with the Erik Jonsson School of Engineering and Computer Science, The University of Texas at Dallas, Richardson, TX 75083-0688. IEEE Log Number 9203879.

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 39, NO. 2, MARCH 1993

The coarray [8]–[10] of an array refers to the set of points at which the spatial correlation function can be estimated with that array. If the array is multidimensional, the coarray must be represented with a set of vectors. The coarray is the set

\[ D = \{ z_i - z_j \} \quad i, j = 0, 1, \ldots, M - 1, \]

where \( z_i \) (a vector) is the location of the \( i \)th sensor and \( M \) is the number of sensors. If the array is linear, the coarray can be thought of as a set of integers that are a lag at which estimates of the spatial correlation function can be obtained with that array. If the array has more than one pair of sensors separated by the same distance, these pairs produce redundant estimates of the correlation function at that lag. In this case, the coarray of that array is said to have redundancies. If there is no pair of sensors separated by some distance (lag) that is smaller than the aperture of the array, the array is said to have a hole in its coarray at that location. We use the term redundancy array to describe an array with redundancies but no holes in its coarray. An optimal redundancy array or minimum redundancy array is one that has no more redundancies than any other redundancy array with the same number of sensors. Alternatively, such an array has the largest possible aperture for a redundancy array with a given number of sensors. Similarly, a hole array is one that has no redundancies, only holes. A minimum hole array is a hole array that has no more holes than any other hole array with the same number of sensors. Such an array has the minimum aperture possible without introducing redundancies.

II. DIFFERENCE BASES AND THEIR APPLICATION TO SENSOR ARRAY DESIGN

Difference bases are entities that were originally defined and studied by number theorists for their intrinsic appeal and elegance. However, they have recently also been studied by engineers, who have found applications for these number theoretic entities [1]–[4], [11], [12]. The application of difference bases that motivated our work was the design of sparse sensor array geometries.

There are several variations on the difference base problem. The type of difference base with which we are primarily concerned with is known as a restricted difference base [13]–[15]. The problem of restricted difference bases can be described by considering a ruler that has some missing marks. Suppose one has a ruler that is \( L \) units long, but there are only \( K(K < L) \) marks on the ruler. If the marks are arranged such that all distances 1, 2, \ldots, \( L \) can be measured with the ruler, then we say that we have a \( K \) element restricted difference basis for \( L \). Let the set \( \{ a_i \} \) represent the locations of the marks on the ruler. Then, we have

\[ 0 = a_1 < a_2 < \cdots < a_K = L. \]

For example, consider a length 6 ruler with marks at 0, 1, 4, and 6. All distances 1 to 6 can be measured with this ruler. Sometimes it is more convenient to represent a difference base by its spacings as opposed to the absolute mark locations. Thus, the aforementioned 6 unit ruler could be represented by the spacings (1, 3, 2). Difference cycles and unrestricted difference bases are closely related to restricted difference bases, but as we will not be using them, we refer the interested reader to [14].

The concept of Golomb rulers [16]–[18] is closely related to that of restricted difference base. Using the same kind of “ruler” framework,
The length $L$ of a Golomb ruler of length $L$ with $K$ marks is a minimum length ruler that measures as many distances as possible from 1 to $L$, without being able to measure any of the distances in more than one way. The length 6 ruler with spacings {1, 3, 2} satisfies this condition. This particular ruler is a "perfect" ruler. It measures each distance once and only once. Perfect rulers with more than 4 marks do not exist.

Here, we consider linear redundancy arrays primarily. The problem of obtaining linear redundancy arrays is completely equivalent to that of obtaining restricted difference bases. Design of linear redundancy arrays can be recast in the following way: Find a set of integers $\{a_i\}$ such that $\sum a_i = A$ and any integer $1 \leq I \leq A$ can be formed by summing adjacent $a_i$’s. The $a_i$’s correspond to intersensor spacings and $A$ is the aperture of the array. This problem is obviously equivalent to that of finding restricted difference bases. Similarly, the problem of obtaining minimum hole arrays is equivalent to that of obtaining Golomb rulers.

One of the main difficulties associated with discovery of redundancy arrays and/or restricted difference bases is that it is difficult to develop constructive approaches. We would like to have a closed form solution of the following form: given $M$ sensors (or $M$ marks on a ruler), 1) what is the maximum aperture allowable such that no holes are induced in the coarray and 2) what are the actual sensor locations achieving this aperture. Although closed form solutions have been obtained for redundancy arrays, there seems to be no efficient way to verify whether they have the maximum aperture for a given number of sensors. In the following, we introduce several analytical forms for redundancy arrays (or restricted difference bases) that generate low redundancy arrays. In some cases, they are known to be maximum aperture (or minimum redundancy), in others they are not. To the best of our knowledge, the only way to verify one way or the other is via an exhaustive search on a computer. With more than roughly 20 sensors, this is impossible with current computers. We will introduce several structures that although cannot be proven to be minimum redundancy, are nevertheless low redundancy, and can yield actual array geometries with very large numbers of sensors in nearly zero computer time.

### Table I

**Known LMR Arrays**

<table>
<thead>
<tr>
<th>$M$</th>
<th>$R$</th>
<th>$A$</th>
<th>$\frac{M^2}{4}$</th>
<th>$\frac{M(M-1)}{2}$</th>
<th>Array(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>3</td>
<td>3.0</td>
<td>1.0</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>6</td>
<td>2.67</td>
<td>1.0</td>
<td>132</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>9</td>
<td>2.78</td>
<td>1.11</td>
<td>332</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>13</td>
<td>2.77</td>
<td>1.15</td>
<td>341</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>17</td>
<td>2.88</td>
<td>1.24</td>
<td>362 &amp; 15322 &amp; 11443</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>23</td>
<td>2.78</td>
<td>1.22</td>
<td>366232 &amp; 11443 &amp; 111554</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>29</td>
<td>2.79</td>
<td>1.24</td>
<td>10872 &amp; 12577441 &amp; 111245332</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>36</td>
<td>2.76</td>
<td>1.25</td>
<td>12710441</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>43</td>
<td>2.81</td>
<td>1.28</td>
<td>217241</td>
</tr>
<tr>
<td>12</td>
<td>16</td>
<td>50</td>
<td>2.88</td>
<td>1.32</td>
<td>1237-141 &amp; 111(20)5:4:33</td>
</tr>
<tr>
<td>13</td>
<td>20</td>
<td>58</td>
<td>2.91</td>
<td>1.34</td>
<td>111(24)5:4:33</td>
</tr>
<tr>
<td>14</td>
<td>23</td>
<td>68</td>
<td>2.88</td>
<td>1.34</td>
<td>11671(10)3:4:32</td>
</tr>
<tr>
<td>15</td>
<td>26</td>
<td>79</td>
<td>2.85</td>
<td>1.33</td>
<td>1135(11)5:6:6</td>
</tr>
<tr>
<td>16</td>
<td>30</td>
<td>90</td>
<td>2.84</td>
<td>1.33</td>
<td>1135(11)5:6:6</td>
</tr>
<tr>
<td>17</td>
<td>35</td>
<td>101</td>
<td>2.86</td>
<td>1.35</td>
<td>1135(11)5:6:6</td>
</tr>
</tbody>
</table>

Notation $n^2$ means $l$ repetitions of the spacing $n$.

### III. Linear Minimum Redundancy and Minimum Hole Arrays

Given $M$ sensors, there are $M(M-1)/2$ pairwise sensor separations. If each pair were separated by a different distance (no redundancies) and holes were not allowed, the aperture of the array would be $A = M(M-1)/2$. However, as previously discussed, there are no perfect arrays with more than four sensors and thus in general, there will be redundancies and/or holes in arrays. In this case

$$A = M(M-1)/2 - R + H.$$  \(3\)

where $R$ is the number of redundancies and $H$ is the number of holes. For redundancy arrays, $H = 0$ and

$$A = M(M-1)/2 - R.$$  \(4\)

Hence, minimizing the number of redundancies is equivalent to maximizing the aperture. For sparse arrays, a commonly used measure of the amount of redundancy in a particular array is $M^2/A$.

Linear minimum redundancy (LMR) arrays are known for only small numbers of sensors. In some cases LMR arrays for a given number of sensors are unique, and in some cases they are not. See Table I for a summary of all known LMR arrays for up to 17 sensors. Similarly, minimum hole arrays must be verified via exhaustive search. The known minimum hole arrays are presented in Table II.

In the following, we will discuss one known family and introduce two new families of redundancy arrays which achieve a limit of 3 for the ratio $M^2/A$ as $M$ grows large. Note: $M^2/A \rightarrow 3$ as $M \rightarrow 0.3A$. Other redundancy array structures which achieve redundancy ratios close to 3 will also be discussed.

### IV. Array Design—Regular Patterns

Our approach to redundancy array design is based on the recognition of patterns in the known LMR arrays that can be generalized into arrays with any number of sensors. The most successful pattern thus far is illustrated in the 11 and 12 element LMR arrays described in...
These arrays are from a pattern independently discovered by the authors in 1989. This solution was also independently discovered by Pearson et al. [19] and originally discovered by Wichmann in the early 1960's [20]. In the following, let \( r^m \) correspond to \( m \) repetitions of the intersensor spacing \( i \). Then the general form of this pattern is given by

\[
\{1', r+1, (2r+1)', (4r+3)', (2r+2)', 1'\}, \tag{5}
\]

where \( r \) and \( l \) are positive integers. All known LMR arrays with more than eight sensors have a realization of the form described in (5). Although LMR arrays are yet unconfirmed for arrays with more than 17 sensors, the pattern described in (5) yields the largest known aperture for all redundancy arrays with more than 8 sensors. Proofs that this expression yields an array with no holes are found in [14] and [4].

We refer to the above pattern as a Type I regular pattern. Arrays from this class are characterized by the largest spacing being repeated some number of times in the middle of the array. We refer to this largest integer as the base of the regular pattern. Different Type I patterns can be obtained dependent on how the base of the array reduces mod 4. The best known solutions are of the form described above where the base is congruent to 3 mod 4. This pattern can be shown to produce arrays such that \( M^2/A \leq 3 \).

We discovered patterns similar to the previous one for bases congruent to 0 and 1 mod 4. Thus far, no general pattern has been obtained for bases congruent to 2 mod 4. The pattern for bases congruent to 1 mod 4 is derived from the pattern for bases congruent to 3 mod 4. It has form

\[
\{1', r, (2r)^{-1}, (4r+1)^{-1}, (2r+1)^{-1}, 1'\} \tag{6}
\]

for positive integers \( r \) and \( l \). (For example, the redundancy array \( \{1, 2, 4, 9, 5, 3, 1, 1\} \) is from this pattern.) The pattern for bases congruent to 0 mod 4 is given by

\[
\{1', (2r+1)^{-1}, (4r)', (2r-1)^{-1}, 1, 2r^{-1}\} \tag{7}
\]

again for positive integers \( r \) and \( l \). (For example, the redundancy array \( \{1, 2, 5, 8, 3, 1, 2\} \) is from this pattern.) As the number of sensors grows large, the ratio \( M^2/A \) approaches 3 for the three patterns described in (5), (6), and (7). (To minimize this ratio, \( l \) is chosen to be approximately half the relevant base.) Patterns similar to these were considered in [2]. However, the patterns described herein produce arrays with much larger apertures than those in [2].

The nature of Type I patterns is made clearer by understanding that no matter how many times the base is repeated in the middle of the array, a redundancy array results. This is exemplified by the 4 and 5 element LMR arrays \( \{1, 3, 2\} \) and \( \{1, 3, 3, 2\} \). It is easily seen that regardless of how many times the 3 is repeated in the above configuration, the array has no holes. See also the 10, 11, and 12 element LMR arrays with 7 as a base in Table I. This characteristic of Type I arrays is explained in the following theorem and proof.

**Theorem I:** Let \( P \) be the point where the base is to appear in a Type I array described in (5), (6), or (7). (See Fig. 1.) Such arrays have the property that any number of repetitions of the base of the array will lead to a redundancy array.

**Proof of Theorem I:** Our proof depends on the following characteristic of Type I arrays: there are no holes in the array when the base appears only once. This property is easily checked for any candidate pattern.

The structure in these arrays is best seen by breaking the spacings to be covered into three groups. (See Fig. 1.) Assume that the base appears once in the middle of the array. Label the point in the array where the base appears as \( P \). The first group consists of spacings that span \( P \). These are presumably the largest spacings. We denote this set as \( I_1 \). A second group of spacings is composed of those spacings that do not span \( P \) and do not terminate on \( P \). These are presumably smaller spacings. We denote this set as \( I_2 \). The third group consists of those spacings that terminate on \( P \)—we call it \( I_3 \). Example spacings from each of these sets are illustrated in Fig. 1. We now consider the case where additional repetitions of the base \( B \) are inserted at \( P \).
It must be proven that in the new array, the sets of spacings
$I_1 + B/I_1 + B$ is the set of spacings described by $I_1 + B = \{i + B/i \in I_1\}$. $I_2 + B$, $I_3 + B$, and the spacings $1$ to $B - 1$ are all covered.

1) The spacings $I_2 + B$ are obviously covered since inserting a $B$ spacing at $P$ increases each spacing in $I_1$. $I_2$.

2) The spacings $I_3 + B$ are obviously covered since the new spacing of size $B$ can readily be combined with any of the spacings originally in $I_2$.

3) Next we show that the spacings $1$ to $B - 1$ are covered by the new array. First, we note that these spacings were covered in the original array and were contained in either $I_1$ or $I_2$. This is because the spacings originally in $I_1$ are all greater than $B$. Since each of the spacings originally in these two sets ($I_1$ and $I_2$) continue to exist after inserting the $B$ spacing, the spacings $1$ to $B - 1$ must be covered in the new array.

4) The spacing $k = i + B$ for $i \in I_1$ is either an element of $I_1$, $I_2$, or $I_3$ since for any interval $k$, there is always an interval $l \in I_1$ such that $l > k$ and hence $k$ must be covered in the original array. The sets $I_1$, $I_2$, and $I_3$ are clearly covered in the new array. That $I_4$ is covered by the new array can be seen by observing the structure of Type I patterns. For all but the very largest intervals in $I_3$, it is easy to see that if $l \in I_3$, then $l \in I_2 + B$ since as the intervals in $I_3$ are increased by $B$ from the middle, intervals equal to $2r + 1$ and $2r + 2$, which sum to $B$, could be dropped from the ends to cover $l$. The larger intervals in $I_1$ that we neglected could not possibly satisfy $i(l \in I_1) = i(l \in I_1) + B$. (However, these larger intervals in $I_3$ are easily shown to also be covered in the new array.) Thus, since $I_1$, $I_2$, and $I_3$ are covered by the new array, we have $I_1 + B$ covered as well.

There is another interesting characteristic of these Type I patterns. Measuring the distances from $P$ to each of the sensors and reducing mod $B$, all remainders $1$ to $B - 1$, must be represented. We have used this property as we searched for patterns.

The second class of arrays (Type II patterns) is derived in the following manner. Let $S = \{i_k\}$ be a redundancy array with $M - 1$ intersensor spacings, let $m$ be the maximum intersensor spacing, and let $A$ be the aperture of $S$. An aperture $2A + m$ redundancy array with $m + M - 1$ intersensor spacings obtained from $S$ is: $\{1^{m-k} + A, i_1, i_2, \ldots, i_{M-1}\}$. For example, $\{1, 2, 3, 7, 7, 7, 7, 7, 4, 4, 1\}$ is a redundancy array with 10 intersensor spacings. As its maximum element $m$ is 7 and its aperture $A$ is 43, a redundancy array with 17 intersensor spacings and aperture 93 is given by $\{1, 1, 1, 1, 1, 1, 44, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2\}$. It follows that this general construction results in a redundancy array with aperture $2A + m$ from the observations: 1) the spacings from 1 to $A$ are obtained from the sequence $S$, 2) the spacings from $A + 1$ to $2A + m$ are obtained from the intersensor space of size $1 + A$ together, possibly, with some of the spacings to its left and right. Extending this idea, it is also easily seen that the prefix $1^{m-k} + 1 + A$ can be added as often as desired to obtain other redundancy arrays. For example, $\{1^{m-k} + 1 + A, 1^{m-k} + 1 + A, i_1, i_2, \ldots, i_{M-1}\}$ is a redundancy array of $3m + k$ intersensor spacings with aperture $4A + 3m$.

The regular patterns formed by the technique of the previous paragraph have the property that there is one intersensor spacing, possibly repeated, which is much larger than the others. Another pattern of similar form can be constructed. Let $S = \{i_1, i_2, \ldots, i_{M-1}, m', i_j, \ldots, i_{M-1}\}$, for some $m \geq 3$ and some $j$, $r \geq 1$ be a sequence of integers (denoting intersensor spacings) with the following properties:

1) $i_{j-1} = m + 1$.
2) $i_j = m + 1$ and $i_{j+1} = m + 1$.
3) For all $d, m - 1 \leq d \leq r + 1$ and $2m - 1$, there are sensors that are separated by distance $d$.

May not be a redundancy array, as it may not have some of the small spacings and some of the large spacings. However, it does have all of the middle spacings, as indicated. Let $A$ denote the sum of all the integers in the set $\{i_k\}$ for $1 \leq n \leq m - 1$. It follows that, for any $r$, the sequence $\{1^{m-k}, (r + 1) \times m, i_1, i_2, \ldots, i_{M-1}\}$ is a redundancy array with $r + 2m - 1$ intersensor spacings and aperture $4A + 2(2r + 1) \times m - r$. For example, for $m = 4, j = 2$, and $r = 3$, the sequence $5, 4, 3$ satisfies the above properties. So, $\{1, 24, 5, 4, 3, 3\}$ is a redundancy array with 13 sensors and aperture 58. (In fact, it is one of the optimum solutions.) As another example, for $m = 10, j = 5$, and $r = 11$, the sequence $9, 8, 5, 11, 10, 9, 6, 3, 9, 5$ satisfies the properties (1)-(3) indicated. Consequently, $\{1, 120, 9, 5, 11, 10, 11, 9, 6, 3, 9, 5\}$ is a redundancy array with 31 sensors and aperture 304. Yet another general pattern with similar features was found. For any odd base number $m$, one can form the sequence $\{1^{m-k}, (r + 2) \times m - 3, (m - 1)^2, m - 2, m', (m - 3)^2, m - 1, (m - 2)^2\}$. For example, a) when $m = 13$ and $r = 15$, one obtains the redundancy array $\{1, 12, 216, 12^2, 11, 13^2, 21, 12, 11^2\}$ with 41 sensors and aperture 522, and b) when $m = 15$ and $r = 23$, one obtains the redundancy array $\{1, 15, 370, 14^2, 13, 15^2, 21^2, 14, 13^2\}$ with 53 sensors and aperture 859. Moreover, the collection of all redundancy arrays in this final variation of the Type II pattern, where the base grows larger with increasing numbers of sensors, satisfies the property that the ratio $M^2/A$ has a limit of 3.2. While redundancy arrays with redundancy factors which reach 3 in the limit have already been described, these Type II patterns provide a low redundancy alternative that may be of use under certain array geometry constraints.

V. EFFICIENT SEARCH TECHNIQUES

Although early work on derivation of linear minimum redundancy arrays was performed before the ready availability of computers, modern computers make the job much easier. We have efficient algorithms for searching through sets of array spacings for LMR candidates. One of the main ideas incorporated in our search method is discussed in [13], [12], [21]. Our search method is as follows: For an array with $M$ sensors and aperture $A$, if all spacings $1$ to $A$ are to be covered by the array, then the spacing $A - 1$ must surely be covered. However, there are only two ways for the spacing $A - 1$ to be covered. The first spacing on one of the sides of the array must be a 1. Since mirror image arrays can be considered equivalent, we fix the first spacing on the left side of the array to be a 1, and proceed. We represent this array as $1 \times r$. The 1 represents the spacing between the first two sensors on the left, while the $r$ represents the as yet undetermined intersensor spacings. Thus, all LMR arrays have a spacing of 1 as their leftmost (or rightmost) intersensor separation. After fixing this first intersensor spacing, we consider the spacing $A - 2$. There are only three ways for this spacing to be covered: $1 \times r$, $11 \times r$, or $12 \times r$. This idea can be carried on ad infinitum. These candidate subarrays can be generated for all subarrays with a given number of sensors. (These subarrays always place the outer sensors. We do not mean to imply that all possible subarrays with a given number of sensors are considered. This is unnecessary.) These subarrays are collected in a table and used as starting points for the search process. This implementation allows for easy parallelization in the modern computing environment of networked workstations by allowing for many machines to be working on different portions of this table.
Using these candidate subarrays, we greatly reduce the total number of array spacings that must be considered in the search for LMR arrays.

A second idea for reducing the size of the search space concerns how one checks whether a given set of spacings is in fact an array with no holes. The most obvious way to determine this would be to look at all spacings 1 to $A$ and determine if at least one pair of sensors was separated by this distance. However, this method has the disadvantage that all sensors must be placed before it can be determined whether the array is LMR. The alternative is to count redundancies as each sensor is placed. For $S_I$ sensors there are $(S_I-1)/2$ intersensor spacings. If the array aperture is to be $A$, there can be at most $S_I*(M-1)/2 - A$ redundancies. If there are more redundancies, there must surely be holes. Thus, if $M - n$ sensors are placed and the number of redundancies exceeds this number, we must drop back and consider a different subarray before continuing to place sensors.

A third idea involves how one counts redundancies in an array (or subarray). Our idea is to represent the array with a binary word. Each bit in the word represents a potential sensor location. Locations with sensors have 1’s and locations without sensors have 0’s. Using this representation, the coarray can be computed efficiently using binary operations. Also, a count of redundancies is easily maintained with this representation. See [22] for a more detailed discussion of this idea.

VI. NEW BOUNDS ON $M^2/A$ FOR ARRAYS WITH REDUNDANCIES AND HOLES

The following results are obtained in a manner similar to how the bound on $M^2/A$ was obtained in [23]. (Some of the following results were also obtained in [9], [10], although through a flawed derivation.) First, represent an $S_I$-element array by a length $A+1$ sequence of 0’s and 1’s—1’s where there are sensors and 0’s where there are not. The symbol $A$ is to represent the aperture of the array. Let $z_i$ (an integer) represent the location of the $i$th element. Then, the Fourier transform of this array representation can be written as

$$ F(\omega) = \sum_{i=1}^{M} \exp(-j\omega z_i). $$

Computing the magnitude squared,

$$ |F(\omega)|^2 = \sum_{i=1}^{M} \sum_{k=1}^{M} \exp(-j\omega(z_i - z_k)). $$

Recalling that $A = M(M - 1)/2 - R + H$ and assuming that the array is a redundancy array ($H = 0$), it follows that there exists a pair of sensors such that $z_i - z_k = n$ for each integer $|n| \leq A$. Thus, $M^2 - (2A + 1)$ terms remain in

$$ \sum_{i=1}^{M} \sum_{k=1}^{M} \exp(-j\omega(z_i - z_k)) = \sum_{i=1}^{A} \exp(-j\omega A). $$

and hence,

$$ M^2 - 2A - 1 \geq \sum_{i=1}^{M} \sum_{k=1}^{M} \exp(-j\omega(z_i - z_k)) - \sum_{i=1}^{A} \exp(-j\omega A). $$

If we allow the array to have holes as well, there are $M^2 - (2A + 1) + 2H$ terms remaining in

$$ \sum_{i=1}^{M} \sum_{k=1}^{M} \exp(-j\omega(z_i - z_k)) = \sum_{i=1}^{M} \exp(-j\omega A). $$

and hence

$$ M^2 - 2A - 1 + 2H \geq \sum_{i=1}^{M} \sum_{k=1}^{M} \exp(-j\omega(z_i - z_k)) - \sum_{i=1}^{A} \exp(-j\omega A). $$

Noting that

$$ (\sin(\omega/2)) \sin(\omega A/2) = \frac{\sin(\omega + \frac{1}{2})}{\sin(\omega/2)} > \frac{2(2A + 1)}{3\pi}. $$

we have

$$ M^2 - 2A - 1 + 2H \geq \frac{4A + 2}{3\pi}. $$

and thus,

$$ \frac{M^2}{A} \geq \frac{4A + 2}{3\pi}. $$

Setting $H = 0$ (for a redundancy array), we have the lower bound for $M^2/A$ given in [23], which was refined in [13]. It is interesting to substitute $A = M(M - 1)/2 - R + H$ into (17):

$$ M^2 - M^2 + M + 2R - 1 + 2H \geq \frac{2M^2 - 2M - 4R + 2 + 4H}{3\pi} . $$

This expression provides a lower bound for the number of holes and redundancies that must be present in an array with a given number of sensors. For LMR arrays, we have

$$ R \geq \frac{M^2}{2 + 3\pi} \frac{M - 1}{2} . $$

and for LMH arrays we have

$$ H \geq \frac{M^2}{3\pi - 2} \left( \frac{M - 1}{3\pi + 2} \right) . $$

For large $M$, we can approximate these by

$$ R \geq \frac{M^2}{3\pi + 2} . $$

and

$$ H \geq \frac{M^2}{3\pi - 2} . $$

Substituting (23) into the expression

$$ \frac{M^2}{A} = \frac{M^2}{2} \left( \frac{M - 1}{2} + H \right) . $$

for a hole array, we have

$$ \frac{M^2}{A} \leq \frac{M^2}{2} \left( \frac{M - 1}{2} + \frac{M^2}{3\pi} \right) . $$

which for large $M$ yields

$$ M^2 \leq \frac{2(3\pi - 2)}{3\pi} \approx 1.58. $$

for hole arrays (Golomb rulers). (Note: It was pointed out by one of the reviewers that P. Erdos has a result indicating that $M^2/A \rightarrow 1.0$ for LMH arrays as $M \rightarrow \infty$. This result is stated without proof in the postscript of [24].)
where $H_{\text{min}}$ is the number of holes for an $M$-element LMH array and $R_{\text{min}}$ is the number of redundancies for an $M$-element LMR array.

For a given number of sensors $M$, any array that has larger aperture than the minimum redundancy array with $M$ sensors will, by definition, have holes. Similarly, any array with aperture smaller than that of the $M$-element minimum hole array will have redundancies. Therefore, any array whose aperture is between that of the minimum redundancy and minimum hole arrays with $M$ sensors must have both holes and redundancies. These are the arrays that we propose to investigate. For a summary of the possible apertures of such arrays, see Fig. 2.

We have preliminary results indicating that these arrays have both holes and redundancies. An example is available for an 8-element array. The eight-element LMR array has aperture 23 and 5 redundancies. The eight-element LMH array has aperture 34 and 6 holes. However, the eight-element array with spacings $\{1.6,4.9,3.2,3\}$ has aperture 28 and only 2 holes and 2 redundancies. Thus, it is closer to a perfect array than either the LMR or LMH arrays with eight elements.

VIII. CONCLUSION

Our understanding of the structure present in known LMR arrays has led to the discovery of several classes of optimal and near optimal array geometries. One question that remains is whether there is a Type I regular pattern for bases congruent to 2 mod 4. Our suspicion is that there is no such Type I regular pattern with the same efficient aperture usage as is available with other bases. Further, we have introduced interesting generalizations to LMR and LMH arrays that are leading to useful array structures. Our efficient search techniques are benefiting efforts in each of these areas.

We have also introduced several new relationships regarding bounds on the number of holes and/or redundancies in arrays. These should prove useful in future research on sparse array geometries.

REFERENCES