Properties of the Beampattern of Weight- and Layout-Optimized Sparse Arrays

Sverre Holm, Bjørnar Elgetun and Geir Dahl
Department of Informatics, University of Oslo
P. O. Box 1080, N-0316 Oslo, Norway

Abstract—**Theory for random arrays predicts a mean sidelobe level given by the inverse of the number of elements. In practice, however, the sidelobe level fluctuates much around this mean. In this paper two optimization methods for thinned arrays are given: one is for optimizing the weights of each element, and the other one optimizes both the layout and the weights. The weight optimization algorithm is based on linear programming and minimizes the peak sidelobe level for a given beamwidth. It is used to investigate the conditions for finding thinned arrays with peak sidelobe level at or below the inverse of the number of elements. With optimization of the weights of a randomly thinned array, it is possible to come quite close and even below this value, especially for 1D arrays. Even for 2D sparse arrays a large reduction in peak sidelobe level is achieved. Even better solutions are found when the thinning pattern is optimized also. This requires an algorithm that uses mixed integer linear programming. In this case solutions with lower peak sidelobe level than the inverse number of elements can be found both in the 1D and the 2D cases.**

I. Introduction

The 3D ultrasound imaging is one of the main innovations in medical ultrasound in this decade. It has applications in all clinical areas where ultrasound is applied. To take the field of cardiology as an example, the advantages are improved surgical planning due to better diagnosis of complex anatomy like valves and septal defects, unrestricted "any-plane" 2D imaging, and improved volume quantification [1]. In most of the demonstrated 3D systems, the data acquisition has been based on mechanical scanning in at least one of the dimensions. One of the main problems of 3D ultrasound is the limited frame rate achievable due to the slow data acquisition, but 2D arrays with electronic scanning in both dimensions have the greatest potential for acceptable frame rates. This is due to the greater beam agility and the possibility for parallel beams [2], [3].

The topic of this paper is the study of the beam pattern of 2D arrays. The 2D arrays in ultrasound represent a technological challenge not the least because of the high channel count [4]. For this reason sparse 2D arrays, where elements are removed by thinning, are considered to be necessary [5]. Steinberg [6] has given a comprehensive theory for the unweighted randomly thinned array. The main results for the far-field continuous wave (CW) beampattern are:

- The probability distribution of the elements' position determines the main lobe's shape and the nearby sidelobes in exactly the same way as they are determined by the weighting in a full array.
- The sidelobe level can be described in a statistical sense and away from the main lobe, the ratio of the mean sidelobe power to the main lobe peak power is $1/K$ where K is the number of remaining elements. This result is independent of the statistical distribution of the elements.
- The sidelobe amplitude away from the main lobe is Rayleigh distributed and unaffected by the element distribution. The peak sidelobe may be as high as 10 dB above the average sidelobe level.

As an example a sparse random array with 256 elements will have an average sidelobe level of -24 dB and peak sidelobes up to -14 dB. This should be compared to the requirement of high-quality ultrasound imaging where a peak sidelobe level better than -30 dB (one-way beampattern) is usually desirable. In the far-field CW case, the two-way beampattern is obtained by multiplying the transmit and receive beampatterns. This is also the case at the focal point for a focused array, and to a first order approximation it is also valid for the pulsed case [3]. Thus a two-way sidelobe level of -60 dB is desirable.

A. Sparse Array Optimization

Several different approaches to overcome the high sidelobe level of random arrays have been proposed. They can be distinguished by whether the element distributions are random or random-like, or periodic, but it is even more useful to distinguish by whether the receiver and transmit element configurations are the same or different.

There is a long history in the radar literature for analysis of beampatterns for sparse arrays when the receiver and transmitter elements are the same. In the far-field CW case, this is equivalent to analysis of the one-way beampattern. In ultrasound imaging, this was the approach used in [7] where it was partially confirmed that Steinberg's results for average one-way sidelobe levels can be squared to estimate the levels for the two-way beampattern for pulsed 2-D arrays. In [8] an optimization of element placement was reported. The optimization criterion was to find the best approximation to the full array's two-way CW beampattern. The solution was an optimal thinning pattern with random-like appearance.

When one allows the transmit and receive thinning patterns to be different, there is some more freedom. Davidsen *et al.* [9] did a search for the random-like thinning patterns that optimized the beam profile in the focal range of interest by minimizing the peak sidelobe level and the beamwidth. Instead of a search or optimization, it has been proposed to use periodic thinning patterns on both receive and transmit. The desired two-way beampattern is obtained by letting the transmit zeros cancel the receive grating lobes and vice versa, based on the first order approximation to the far-field CW case. This principle has previously been proposed for design of 1D array systems and was applied to 2D arrays in [3]. The method has been further developed by Lockwood *et al.* [10], [11] and good results from imaging of a phantom using a 1D array have been presented. For the pulsed case and away from the focal point, this approach gave even better results than a random sparse array. Others [12] have reported that in 2D array simulations the random-like and periodic thinning approaches have given comparable performance.

Although much interest has been generated by the periodic thinning approach, it is believed that there is still more potential for performance improvements by optimizing random-like element distributions. This work is an attempt to find the properties of the random-like thinning patterns. It is based on optimization of the one-way response by either changing the element apodization or the element positions or both.

There have been previous attempts to apodize sparse arrays. This has been reported to have no effect, but this was because regular apodization functions were sampled [7]. In [13] we showed that it is possible to find apodization functions or element weights for a given thinning pattern that give the beampattern optimal properties. An important point is that these functions have little or no resemblance with the corresponding full array's apodization function. A limitation of this work was that is was not possible to optimize the full angular extent of the sidelobe region for a sparse array. This was due to the algorithm used (Remez exchange algorithm). In [14] this approach was extended from 1D to 2D arrays, and improved results were reported. By using the linear programming algorithm for optimization, it was possible to optimize the whole sidelobe region. Due to the properties of 2D array elements (high impedance, low sensitivity) it is undesirable to apodize the elements of a 2D transducer array. The goal of this work is, therefore, not primarily to propose practical weighting functions, but rather the optimization methods are used to find properties of the beampattern of such arrays. Of special interest is to determine the minimum peak sidelobe level and compare it with the predictions from random theory. Finally a method is also described for optimizing the element positions of a random-like sparse array. This optimization gives results that are more directly useful in an array design. Other related work on joint optimization of thinning pattern and weights has been reported in the context of sonar arrays in [15] and [16]. Like all of the previously cited papers our approach is based on allowing elements only on a fixed underlying grid of positions as opposed to what was done in [17].

The optimization criterion used is also very important. In many cases a minimization of the maximum sidelobe is used. This is a criterion which is related to imaging of a strong reflecting point target in a nonreflecting background containing other point targets. An alternative criterion is to minimize the integrated sidelobe energy. In an imaging system, this is related to imaging of a nonreflecting area like a cyst or a ventricle in a background of reflecting tissue. Some results on weight optimization for 1D arrays using this criterion and the quadratic optimization algorithm of [18] have been reported in [19]. Both of the optimization criteria are relevant to clinical ultrasound imaging. In this paper we find the properties of arrays based on minimization of the peak sidelobe, because this has been the most common criterion until now, and it is straightforward to formulate optimization algorithms for it.

B. The Beam Pattern of a Planar Array

The far-field continuous wave (CW) beampattern of an array with an even number, $L = 2N$, of omnidirectional elements is given as [20]:

$$W(\vec{k}) = \sum_{n=1}^{2N} w_n e^{j\vec{k}\cdot\vec{x}_n} \tag{1}$$

where the array element locations are $\vec{x}_n \in \mathbb{R}^3$ with the corresponding weights $w_n \in \mathbb{R}$. The wavenumber vector $\vec{k} \in \mathbb{R}^3$ has amplitude $|\vec{k}| = 2\pi/\lambda$ where λ is the wavelength.

Let the unit direction vector be $\vec{s}_{\phi,\theta} = (\sin\phi\cos\theta, \sin\phi\sin\theta, \cos\phi)$ in rectangular coordinates (Fig. 1). Then the wavenumber vector is $\vec{k} = 2\pi\vec{s}_{\phi,\theta}/\lambda$.

The elements of a 2D planar array are located in the xy-plane with element n at $\vec{x}_n = (x_n, y_n, 0)$. Thus the beampattern is:

$$W(k_x, k_y) = \sum_{n=1}^{2N} w_n e^{j(k_x\cdot x_n + k_y\cdot y_n)} \tag{2}$$

The beampattern has the following properties:

- For real weights, the beampattern is conjugate symmetric, i.e. $W(k_x, k_y) = W^*(-k_x, -k_y)$.
- Symmetric arrays with symmetric weights give a real beampattern.

When the two properties are combined, the beampattern for an array with an even number of elements is real and equal to:

$$\begin{aligned} W(\phi,\theta) &= 2\sum_{n=1}^{N} w_n \cos\left(\frac{2\pi}{\lambda}\vec{s}_{\phi,\theta}\cdot\vec{x}_n\right) \qquad (3)\\ &= 2\sum_{n=1}^{N} w_n \cos\left(\frac{2\pi}{\lambda}\sin\phi(x_n\cos\theta + y_n\sin\theta)\right) \end{aligned}$$

which gives the array response to a monochromatic wave from direction (ϕ, θ). A similar expression for an odd number of elements can easily be found.

Using matrix notation one gets

$$W(\phi,\theta) = \mathbf{v}(\phi,\theta)\mathbf{w} \tag{4}$$

where $\mathbf{w} = [w_1 \ \cdots \ w_N]^T$ are the element weights and the kernel row vector is given as:

$$\mathbf{v}(\phi,\theta) = \left[2\cos(\frac{2\pi}{\lambda}\vec{s}_{\phi,\theta}\cdot\vec{x}_1)\ ,\ \cdots,\ 2\cos(\frac{2\pi}{\lambda}\vec{s}_{\phi,\theta}\cdot\vec{x}_N)\right]. \tag{5}$$

II. Optimization of Beampattern

Two optimization problems will be formulated as linear programming problems. The first is a minimization of the maximum sidelobe level by varying element weights. The second problem gives rise to a mixed integer linear programming problem which is considerably harder to solve. It is a minimization of the number of active elements and an optimization of the weights in order to achieve a specific maximum sidelobe level.

A. Optimization of Element Weights

The objective is to minimize the sidelobe level in a continuous region $\mathcal{R}$ of the $\phi\theta$-plane (Fig. 2), which is the equivalent of a filter's stopband. Since the beampattern is symmetric about the θ-axis; $W(-\phi,\theta) = W(\phi,\theta)$, only the right halfplane is necessary for the optimization. The passband in this case is minimal in the sense that it consists only of the θ-axis.

The element weight optimization problem is to minimize the beampattern in the stopband by varying the element weight vector subject to the constraint of having a normalized mainlobe. This problem may be formulated and solved as a linear programming problem as discussed next.

A linear programming (LP) problem is the minimization of a linear objective function subject to a (finite) set of linear inequalities and linear equations [21], [22]. In matrix form an LP problem may be written as

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} \le \mathbf{b} \end{array} \tag{6}$$

where $\mathbf{x}$ is a vector of n variables, and the data is given by the $m \times n$-matrix A and the vectors $\mathbf{c}$ and $\mathbf{b}$. Today large-scale LP problems can be solved efficiently on standard computers with good algorithms and implementations.

The stopband region $\mathcal{R}$ is discretized into a set of M gridpoints $R = \{(\phi_m,\theta_m) : m = 1,\ldots,M\}$. Introduce the $M \times N$ matrix V with the mth row being the N-dimensional row vector $\mathbf{v}(\phi_m,\theta_m)$ given in (5), or $\mathbf{v}_m$ for short. For a given element weight vector $\mathbf{w}$ the maximum sidelobe level δ_{sl} on the discrete set R is defined as:

$$\begin{array}{l} \delta_{sl}(\mathbf{w}) = \max\{|\mathbf{v}_m\mathbf{w}| : 1 \le m \le M\} = \\ \max\{|\sum_{n=1}^{N} v_{m,n}w_n| : 1 \le m \le M\}. \end{array} \tag{7}$$

Thus the element weight optimization problem is to minimize δ subject to $|\mathbf{v}_m\mathbf{w}| \le \delta$ for $1 \le m \le M$, and a normalization corresponding to a unit response for zero azimuth angle, $\mathbf{v}_0\mathbf{w} = 1$, where $\mathbf{v}_0 = [1,\ldots,1]$. This is a nonlinear optimization problem, but a standard reformulation [22], may be used to turn it into an LP problem. Consider the LP problem:

$$\begin{array}{llll} \text{minimize} & & \delta & \\ \text{subject to} & & & \\ \text{(i)} & \mathbf{v}_0\mathbf{w} = 1; & & \\ \text{(ii)} & \mathbf{v}_m\mathbf{w} \le \delta & \text{for } m \le M; & \\ \text{(iii)} & -\mathbf{v}_m\mathbf{w} \le \delta & \text{for } m \le M. & \end{array} \tag{8}$$

The variables are $\mathbf{w}$ and δ. This problem is of the form (6) with

$$\mathbf{x} = \begin{bmatrix} \mathbf{w} & \delta \end{bmatrix}^T, \qquad \mathbf{c} = \begin{bmatrix} \mathbf{0} & 1 \end{bmatrix}^T,$$

$$A = \begin{bmatrix} \mathbf{v}_0 & 0 \\ -\mathbf{v}_0 & 0 \\ V & -1 \\ -V & -1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ -1 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

where $\mathbf{0}$ is a vector of all ones.

In an optimal solution of (8) the variable δ equals the minimum value of $\delta_{sl}(\mathbf{w})$ in (7). This problem is, therefore, a min-max problem. Thus the element weight optimization problem may be solved as the LP problem (8).

A very fast and reliable LP solver, CPLEX [23], was used. This is an optimization library for solving linear programming and integer linear programming problems. The problem (8) may be solved for problems corresponding to 2D arrays with thousands of elements where several hundreds of them are active. Thus, the main work of the implementation is to generate the correct entries in the coefficient matrix and vectors. The simplex algorithm (a part of CPLEX) was used for solving the problem. In order to speed up the algorithm (roughly by a factor of 3, it turned out) one should solve the dual problem of (8) or, equivalently, use the dual simplex algorithm. Briefly, the dual problem is an associated LP problem where the role of variables and constraints are interchanged in a certain sense (for a more accurate description, see [21], [22]). The main point is that, while the problem (8) has $2M + 1$ constraints, the dual problem has only N constraints (recall that M is considerably larger than N). Now, it is the general experience (also confirmed in theory) that the main contribution to computational time of the simplex algorithm is caused by the number of constraints; the number of variables are not that important. Thus, one can solve the dual problem a lot faster than (8), and the optimal solution of (8) may also be found directly out of this.

The routines in the Optimization toolbox in MATLAB [18] were also used to solve the weight optimization problem and worked well for 1D arrays, except that for larger problems CPLEX was considerable faster, as expected.

It should be mentioned that more specialized algorithms for solving the element weight optimization problem may be developed. It is, for instance, straightforward to change the norm from the min-max to the sum of absolute values. This may be used for minimization of the average sidelobe level. Another natural extension is to assign different sidelobe requirements to different angles. One could for instance take element directivity into account by allowing the angles far away from the mainlobe to have higher sidelobes than the smaller angles. This can be achieved by using an angle-dependent error weighting in (8). It is also possible to develop more efficient algorithms by eliminating some of the variables. It should be pointed out, however, that for the problems in this study the algorithms described above were all suitable.

B. Optimization of Element Layout and Weights

The simultaneous weighting and thinning problem is a natural extension of the element weighting problem in the previous section. Since the objective is to minimize the number of array elements, a binary variable $x_n \in \{0,1\}$ is introduced for each element. The purpose is to let $x_n = 1$ indicate that the element is present, and $x_n = 0$ indicates that the element is removed by thinning.

Note that the objective is now to minimize the number of array elements, rather than minimizing the sidelobe level δ. The sidelobe level is consequently a fixed parameter $\bar{\delta}$ in this problem as in [15]. Thus one considers the element weighting and thinning optimization problem: minimize the number of array elements subject to constraints assuring a normalized mainlobe and fixed sidelobe level. Consider the following problem

$$\begin{array}{lll} \text{minimize} & \sum_n x_n & \\ \text{subject to} & & \\ \text{(i)} & \mathbf{v}_0\mathbf{w} = 1; & \\ \text{(ii)} & \left[\begin{array}{c} V \\ -V \end{array}\right]\mathbf{w} \le \bar{\delta}\mathbf{1}; & \\ \text{(iii)} & \gamma_1 x_n \le w_n \le \gamma_2 x_n & \text{for } n \le N; \\ \text{(iv)} & x_n \in \{0,1\} & \text{for } n \le N. \end{array} \tag{9}$$

Here the constraints (i) and (ii) are as before except that $\bar{\delta}$ is given, while (iii) gives a logical link between the layout variable x_n and the weight variable w_n. In order to vary the weight w_n between the two bounds γ_1 and γ_2 one has to set x_n to 1. The actual values of the parameters γ_1 and γ_2 may be set depending on the specific problem studied. For instance, interesting choices are $\gamma_1 = 0$ (nonnegative weights), $-\gamma_1 = \gamma_2 > 0$ (symmetric bounds), or positive weights with γ_2/γ_1 restricted to the maximum allowed dynamic range of the apodization hardware.

The problem (9) is a mixed integer linear programming problem, i.e., a linear programming problem where some or all variables are required to be integral. This particular problem may be written in matrix form similarly to what was shown in the previous section. In general, mixed integer LP problems are computationally very difficult optimization problems. Even this particular problem is difficult, i.e., to find an optimal solution seems hard also for moderately-sized problems. This is mainly due to the complex structure of the matrix V which comes in combination with the integrality constraints on the layout variables x_n. In practice it turns out that it is only realistic to solve problems of size corresponding to 1D arrays so far. At present only simplified heuristic methods may be used to solve for the larger problems [8], [16]. One important use of the mixed integer linear programming algorithm is that it may be used to compare the quality of different simplified heuristic methods for the same problem.

Small-scale problems may be solved by the branch and bound method in CPLEX [23]. This is a general method for solving mixed integer linear programming problems in which the feasible region is gradually divided into finer subregions for which a linear programming problem is solved. To (hopefully) control the combinatorial explosion of these subdivisions, one cuts off in subdivisions that cannot lead to a further improvement of the current best solution.

For larger problems CPLEX will run “forever,” but even early in this process it may find good solutions satisfying (9), that may be of interest. The problem, however, is to prove that these solutions are the optimal ones.

The standard algorithms in CPLEX were used for solving the element weighting and thinning optimization problem. Thus a main purpose here was to point out the usefulness of formulating this problem as a mixed integer programming problem. Further algorithmic work is required to solve larger problems, or to solve variations of this problem. Such variations may be to solve for the minimum maximum sidelobe level for a fixed number of active elements when the weights and the thinning pattern is allowed to vary, or when only the thinning pattern is varied and the active elements are unweighted. Some work along these lines is in progress.

III. Examples

A. 1D Sparse Array

A 3.5 MHz array with half wavelength spacing, 64 elements and Gaussian thinning to 48 elements was optimized. An example of the beam patterns before and after optimization are shown in Fig. 3.

Several such optimizations were performed for various beamwidths and thinning patterns. For each thinning pattern, the start angle ϕ_1 (the boundary between the mainlobe and sidelobe regions) was varied and an optimization was performed. The resulting peak sidelobe and -6 dB beamwidth is plotted in Fig. 4. Each curve is the result of between 5 and 18 such optimizations. Fig. 4 shows two dash-dot lines which are the results of optimizing the weights to give uniform sidelobe levels for the full arrays. The left-hand one is the performance for a full 64-element array, and the right-hand one for a full 48-element array. Only thinned arrays with performance better than the 48-element curve are of interest. All the remaining curves are for a 64-element array thinned to 48 elements. The upper solid line shows performance for the worst symmetric thinning that could be found, giving a minimum sidelobe level of about -13 dB. The two dashed lines are two realizations of random Gaussian thinning. Both of them start leveling off at -17 to -18 dB sidelobe level. This is in the vicinity of the mean sidelobe level predicted for a random array given as the inverse of the number of elements which is -16.8 dB. However, with the optimization used here this value is achieved as a peak value instead.

Finally the two lower solid curves are the results from optimizing the weights for two near-optimal thinning patterns. They were obtained with the combined weight and layout optimization algorithm with sidelobe targets of -18 and -19.5 dB. The other values in their curves were obtained by keeping the layout and then optimizing the weights only for different values of start-angles in the optimization. With such thinnings the peak sidelobe level can be improved down to the range -17 to -20 dB.

All the thinning patterns are shown in Table I. Examples of the weights required are shown in Fig. 5. They are quite different from the much smoother weight functions that are obtained for full arrays (see the Dolph-Chebyshev weights of Figs. 46-49 of [24]).

B. 2D Sparse Arrays

A 2D array for 3.5 MHz with 12 by 12 elements with half wavelength spacing in both dimensions was then considered. The array is inscribed in a circle giving 112 elements. Random thinning to 64 elements (57%) and optimization of the weighting gives a beampattern with a sidelobe level of -12 to -15 dB. The procedure for finding the optimal thinning and weighting was then used with a sidelobe target of -19.5 dB. The optimized layout was then input with varying start-angles in the weight optimization algorithm. The peak sidelobe level can now be reduced down to -20 to -22 dB (Fig. 6). Each curve is the result of between 5 and 8 optimizations with different start values for the azimuth angles. The sidelobe value should be compared to the value predicted for mean sidelobe level of $1/64 = -18.1$ dB, and shows that there is a potential of getting a peak value which is 3 dB lower than that predicted for the mean if optimized thinning patterns can be found. This is about the largest array size where optimized element layouts can be found with reasonable use of computer resources (less than about 4 hours CPU time and some hundreds of Mbytes of RAM). The four element layouts are shown in Fig. 7.

Finally, in order to show the ability of the weight optimization algorithm to deal with large arrays, a 64 by 64 element array with half wavelength spacing at 3.5 MHz is considered. The array is inscribed in a circle giving a total of 3228 elements. The array is randomly thinned to 404 elements (87.5 % thinning) as shown in Fig. 8 and the response is optimized. The size of the problem is to find 202 weights plus the sidelobe level using 11520 control points for sidelobe control. The result is shown in Fig. 9. The peak sidelobe is reduced from -9.5 dB to -17.4 dB with only a slight increase in the beamwidth. The four peaks before optimization were located at approximately $\phi = 64°$ and at $\theta = \pm 26°$ and $\theta = \pm 62°$. Compared to the predicted mean value of $1/404 = -26.1$ dB, there is still some way to go before the peak is down to that level. Based on the previous example, this is due to the properties of the thinning pattern, which was selected at random and could not be optimized due to the large number of elements.

The optimization has led to an increase in the average from -28.8 dB to -26.1 dB. This indicates that the sidelobe distribution has become more compact: lower peak value and higher mean. The increase in the mean seems to be the price to pay for getting the peak value down at least as long as a large increase in beamwidth is not allowed. As indicated previously, the LP algorithm could have been formulated as a minimization of the mean rather than the peak.

IV. Discussion

A. Focusing and Pulsing

In the case of focusing, the equations for the beampattern (1 - 4), must be expanded with terms that include the focal depth. The optimization region will no longer only be described by a region in angles as in Fig. 1, but also a region in depth. However, near the focus, the farfield assumption is valid. It is our experience that, if there are sidelobe peaks in the farfield beampattern, then they will also be evident over a range of depths. However, the focusing will change the delays or phases applied to the array elements, and this will influence the optimality of the solutions presented here. As a parenthesis it may be mentioned that it is possible to obtain optimal solutions at a single frequency by only changing the phases [25].

The issue of pulsing with broad bandwidth pulses is also important. In this paper, the assumption is that there is only a single frequency present. In general pulsing will tend to smear out sidelobe peaks. To some extent this happens to the sparse arrays also, but the peaks are still there, although smaller, after pulsing.

B. Optimization of Steered Arrays

For optimization over the elevation, azimuth space the sidelobe level over the region defined by all elevation angles and with the azimuth angle in the range $[\phi_1, \pi/2]$ should be minimized, where ϕ_1 is the boundary between the mainlobe and sidelobe regions. This corresponds to an annular region in $\vec{k}$-space of radius $2\pi/\lambda$ centered at the origin (Fig. 10). Due to the sampled nature of the aperture, the beampattern will be repeated for argument of k_x and k_y larger than $2\pi/\lambda$ when the pitch is $\lambda/2$.

When steering is applied to the array, the beampattern is $W(k_x - k_x^0, k_y - k_y^0)$ [20]. The visible region will shift to have its center at the steering direction (k_x^0, k_y^0), while the optimized region from the array is still centered at the origin. There is, therefore, no longer full overlap between the optimized region and the visible region. In order to deal properly with steering, one must, therefore, optimize a larger region.

For a 1D array this is greatly simplified. The only relevant variable is k_x, and when there is steering, the argument in the beampattern is

$$k_x - k_x^0 = 2\pi/\lambda \cdot (\sin\phi - \sin\phi^0) = 2\pi/\lambda \cdot u \qquad (10)$$

First there is always symmetry with respect to $u = 0$. When, in addition the element locations are all on a grid with distance $\lambda/2$, there will also be symmetry with respect to $u = 1$. In this case optimization over the region $u \in [0, 1]$ ensures that the array can be steered to any azimuth angle [16]. If the pitch is less than $\lambda/2$, it is simple to find from this argument that the optimization region must be larger than $u \in [0, 1]$.

It turns out that the achievable minimum sidelobe levels for 2D arrays are comparable independent of whether the array is optimized for steering or not, although the actual element layouts or weights may be different. The results of this paper are therefore representative of those that can be obtained when steering is taken into account also.

V. Conclusion

A method based on linear programming for finding the optimum weights for minimum peak sidelobe level and a method using mixed integer linear programming for finding both the weights and the element layout have been presented. They have been used to find properties of sparse arrays with random thinning and arrays with optimized thinning. The properties are found by trading off sidelobe level for beamwidth.

Theory for random arrays predicts a mean sidelobe level given by the inverse of the number of elements. In practice however, the sidelobe level fluctuates much around this mean. With optimization of the weights, it is possible for the peak value to come quite close and even below the predicted value for the mean, especially for 1D arrays. Even for 2D sparse arrays a large reduction in peak sidelobe level is achieved. However, when the thinning pattern is optimized also, solutions which have lower peak sidelobe level than the inverse number of elements can be found both in the 1D and the 2D cases.

It is also shown that for 2D sparse phased arrays with $\lambda/2$ grid spacing, steering requires a larger region in the wavenumber domain where sidelobes should be optimized than for an unsteered array. This is different from a 1D array where steering does not add any new constraints.

Acknowledgment

We would like to thank the reviewers for important comments that helped improve this paper.

REFERENCES

[1] A. Salustri and J. R. T. C Roelandt, "Ultrasonic three-dimensional reconstruction of the heart," *Ultrasound Med. Biol.,* vol 21, no. 3, pp. 281-293, 1995.

[2] D. P. Shattuck, M. D. Weinshenker, S. W. Smith, and O. T. von Ramm, "Explososcan: A parallel processing technique for high speed ultrasound imaging with linear phased arrays," *J. Acoust. Soc. Amer.,* vol. 75, no. 4., pp. 1273-1282, Apr. 1984.

[3] S. W. Smith, H. G. Pavy, and O. T. von Ramm, "High-speed ultrasound volumetric imaging system. Part I: Transducer design and beam steering," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.,* vol. 38, no. 2, pp. 100-108, 1991.

[4] B. A. J. Angelsen, H. Torp, S. Holm, K. Kristoffersen and T. A. Whittingham, "Which transducer array is best?" *Eur. J. Ultrasound,* vol. 2, pp. 151-164, 1995.

[5] S. Holm, "Medical ultrasound transducers and beamforming," *Proc. Int. Cong. Acoust.,* Trondheim, Norway, June 1995, pp. 339-342.

[6] B. Steinberg, *Principles of aperture and array system design.* New York: Wiley, 1976.

[7] D. H. Turnbull and F. S. Foster, "Beam steering with pulsed two-dimensional transducer arrays," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.,* vol. 38, no. 4, pp. 320-333, 1991.

[8] P. K. Weber, R. M. Schmidt. B. D. Tylkowski, and J. Steck, "Optimization of random sparse 2-D transducer arrays for 3-D electronic beam steering and focusing", *Proc. IEEE Symp. Ultrason.,* Cannes, France, 1994, pp. 1503-1506.

[9] R. E. Davidsen, J. A. Jensen, and S. W. Smith, "Two-dimensional random arrays for real time volumetric imaging", *Ultrason. Imaging,* vol. 16, pp. 143-163, 1994.

[10] G. R. Lockwood, P.-C. Li, M. O'Donnell, and F. S. Foster, "Optimizing the radiation pattern of sparse periodic linear arrays", *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.,* vol. 43, no. 1, pp. 7-14, 1996.

[11] G. R. Lockwood, F. S. Foster, "Optimizing the radiation pattern of sparse periodic two-dimensional arrays", *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.,* vol. 43, no. 1, pp. 15-19, 1996.

[12] S. W. Smith, R. E. Davidsen, C. D. Emery, R. L. Goldberg, and E. D. Light, "Update on 2-D array transducers for medical ultrasound," *Proc. IEEE Symp. Ultrason.,* Seattle, WA, 1995, pp. 1273-1278.

[13] J.O. Erstad, S. Holm, "An approach to the design of sparse array systems", *Proc. IEEE Symp. Ultrason.*, Cannes, France, 1994, pp. 1507-1510.

[14] S. Holm, B. Elgetun, "Optimization of the beampattern of 2D sparse arrays by weighting," *Proc. IEEE Symp. Ultrason.,* Seattle, Washington, 1995, pp. 1345-1348.

[15] R. M. Leahy and B. D. Jeffs, "On the design of maximally sparse beamforming arrays," *IEEE Trans. Antennas Propagat.,* vol. AP-39, No. 8, pp. 1178 - 1187, Aug. 1991.

[16] V. Murino, A. Trucco, and C. S. Regazzoni, "Synthesis of unequally spaced arrays by simulated annealing," *IEEE Trans. Signal Processing,* vol. 44, pp. 119-123, Jan. 1996.

[17] H. Schjær-Jacobsen and K. Madsen, "Synthesis of nonuniformly spaced arrays using a general nonlinear minimax optimization method," *IEEE Trans. Antennas Propagat.,* vol. AP-24, pp. 501-506, July 1976.

[18] A. Grace, *Optimization Toolbox for Use with MATLAB.* Natick, MA: The Mathworks, 1992.

[19] S. Holm, "Maximum sidelobe energy versus minimum peak sidelobe level for sparse array optimization," *Proc. IEEE Nordic Signal Processing Symp.,* (NORSIG 96), Espoo, Finland, Sept. 1996, pp. 227-230.

[20] D. H. Johnson, D. E. Dudgeon *Array Signal Processing.* Englewood Cliffs, NJ: Prentice Hall, 1993.

[21] G. Strang, *Linear Algebra and Its Applications,* 2nd ed. New York: Academic, 1980.

[22] V. Chvátal, *Linear Programming.* San Francisco: CA, Freeman, 1983.

[23] *Using the CPLEX callable library.* Incline Village: NV, CPLEX Optimization, 1994.

[24] F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proc. IEEE,* vol. 66, pp. 61-83, Jan. 1978.

[25] J. F. DeFord and O. P. Gandhi, "Phase-only synthesis of minimum peak sidelobe patterns for linear and planar arrays," *IEEE Trans. Antennas Propagat.,* vol. AP-36, pp. 191-201, Feb. 1988.

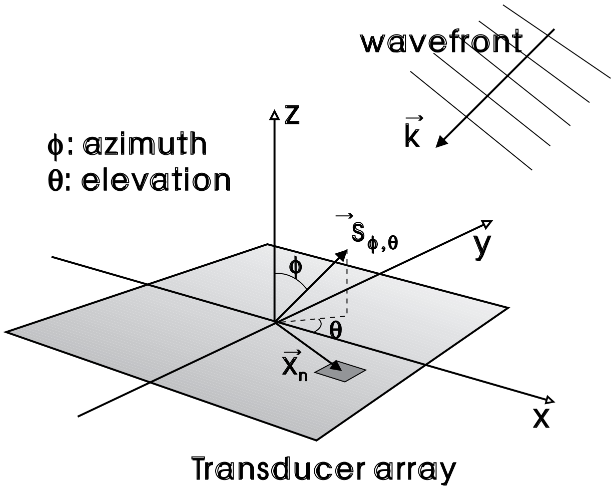

Fig. 1. A 2D planar array with coordinate system.

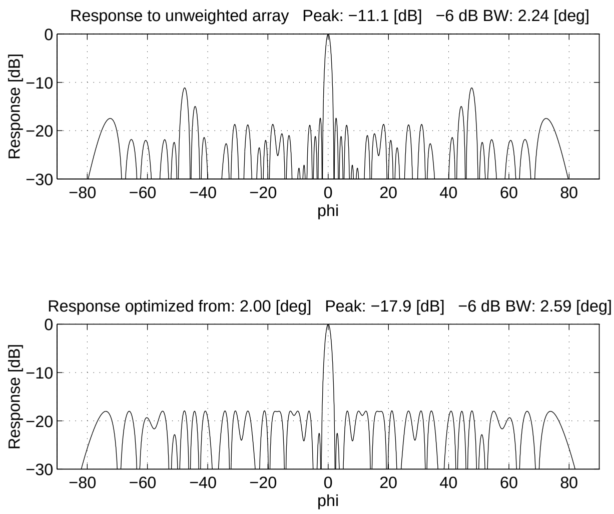

Fig. 3. Beampattern before and after optimization for 64-element array randomly thinned to 48 elements. Thinning and weights are shown in the center panel of Fig. 5.

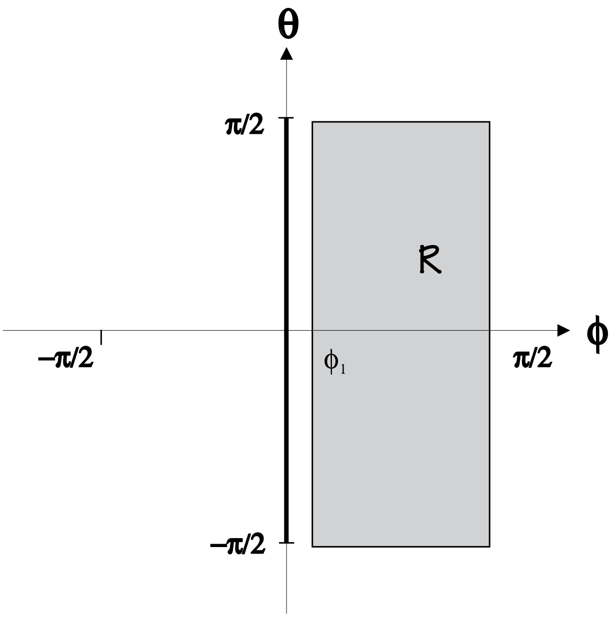

Fig. 2. The optimization region $\mathcal{R}$ in the $\phi\theta$-plane.

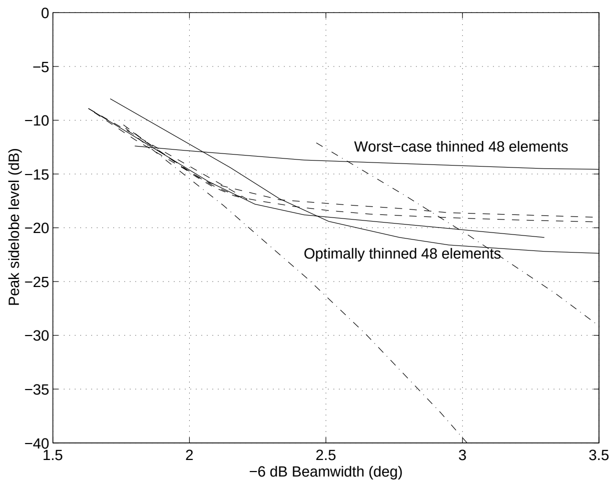

Fig. 4. Sidelobe level as a function of beamwidth for uniform sidelobe level 64-element and 48-element full arrays (dash-dot lines), for two realizations of random 25% thinning of the 64-element array (dashed lines), and for worst-case and optimally 25% thinned arrays (solid lines).

Elements enabled	Comment
11011101110111011101110111011101	Worst-case symmetric array
11010110110110101111101111111011	Random 1 (upper dashed curve)
11011011011111111001010111110111	Random 2 (lower dashed curve)
10111100011001111101101111111111	Optimized 1, (-18 dB) (upper solid curve)
00101001111101111011101111111111	Optimized 2, (-19.5 dB) (lower solid curve)

TABLE I

LEFT-HAND PART (32 ELEMENTS) OF SYMMETRIC 64-ELEMENT ARRAYS. ALL REFERENCES TO RELATIVE POSITION ARE TO THE RIGHT-HAND PART OF CURVES IN FIG. 4

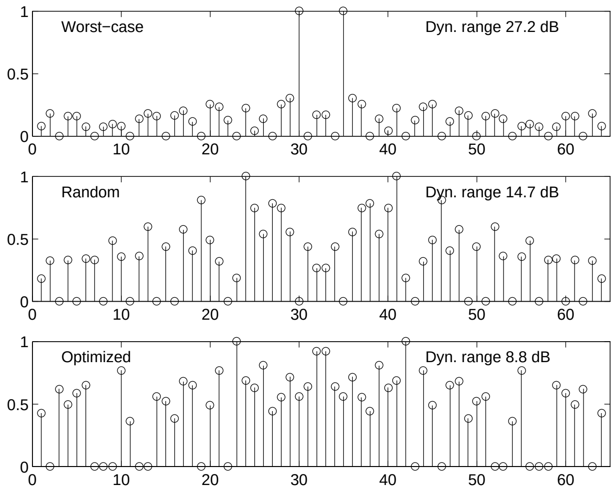

Fig. 5. Weights found after optimization from 2 degrees for three different element layouts. The beampattern of the random layout is shown in the lower panel of Fig. 3.

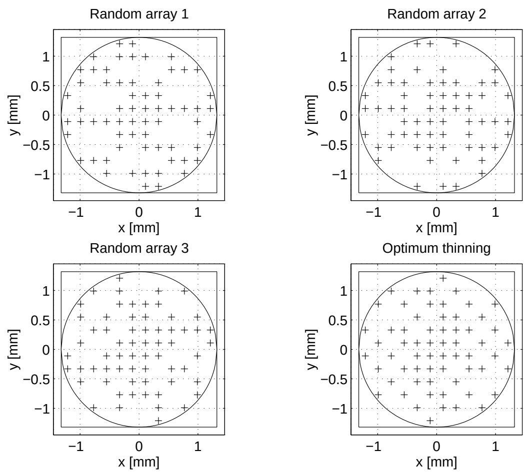

Fig. 7. Element layouts for 112-element full array thinned to three different random 64-element layouts and a 62-element optimized layout. The random arrays are sorted according to the peak sidelobe level in Fig. 6 with Random 1 having the highest peak sidelobe level for large beamwidths.

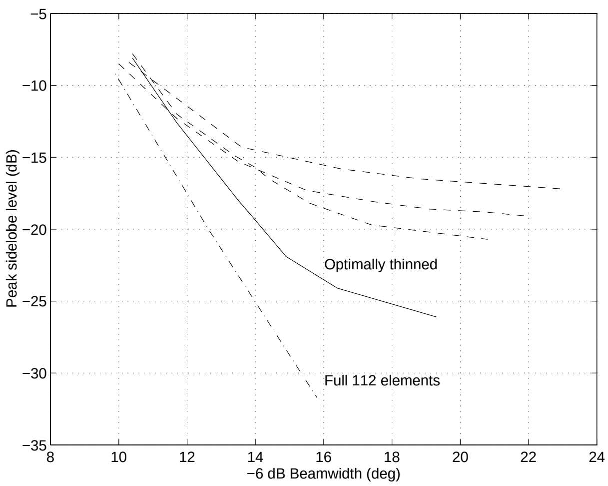

Fig. 6. Sidelobe level as a function of beamwidth for several weight-optimized uniform sidelobe cases: 112-element full array (dash-dot line) and three realizations of random thinning to 64 elements (dashed lines). The best result is obtained for a layout-optimized 62-element thinning (solid line).

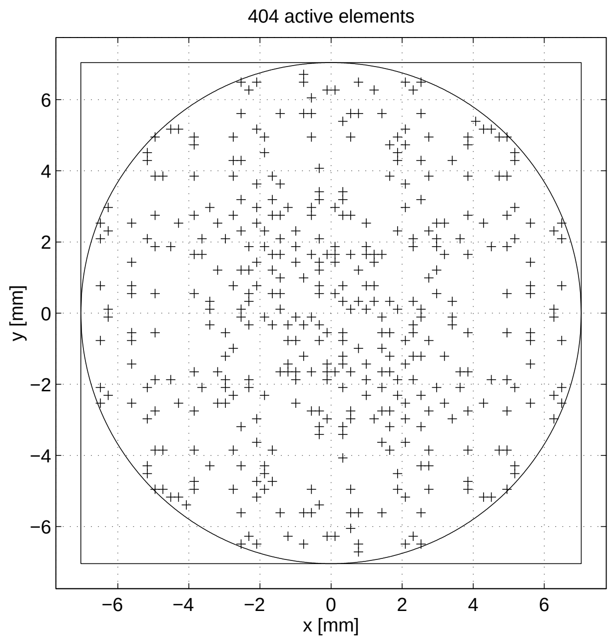

Fig. 8. Element layout for 64 by 64 element array first inscribed in a circle (3228 elements) and then randomly thinned by 87.5% (404 active elements).

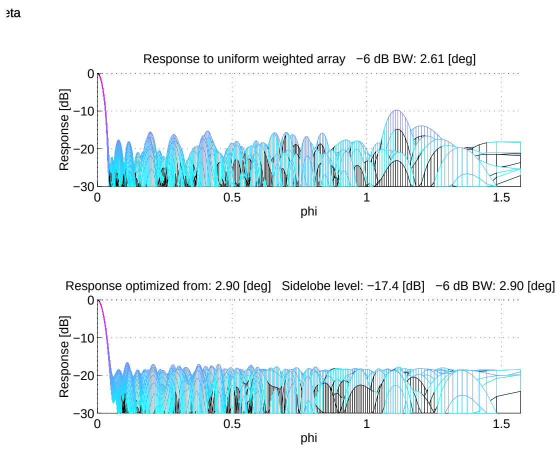

Fig. 9. Beampattern for 64 by 64 element array first inscribed in a circle (3228 elements) and then thinned by 87.5% (404 active elements). The unweighted (top) and the optimally weighted (bottom) responses are shown as a function of azimuth angle and seen from the side in 3D space, i.e. the peak values over all elevation angles are shown.

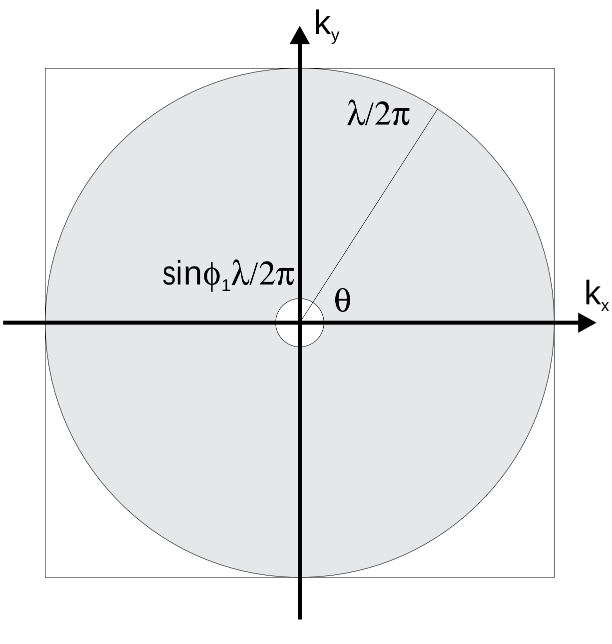

Fig. 10. The optimization region in k-space.

List of Tables

I Left-hand part (32 elements) of symmetric 64-element arrays. All references to relative position are to the right-hand part of curves in Fig. 4 . . . 8

List of Figures

1 A 2D planar array with coordinate system. 7
2 The optimization region $\mathcal{R}$ in the $\phi\theta$-plane. 7
3 Beampattern before and after optimization for 64-element array randomly thinned to 48 elements. Thinning and weights are shown in the center panel of Fig. 5. 7
4 Sidelobe level as a function of beamwidth for uniform sidelobe level 64-element and 48-element full arrays (dash-dot lines), for two realizations of random 25% thinning of the 64-element array (dashed lines), and for worst-case and optimally 25% thinned arrays (solid lines). 7
5 Weights found after optimization from 2 degrees for three different element layouts. The beampattern of the random layout is shown in the lower panel of Fig. 3. 8
6 Sidelobe level as a function of beamwidth for several weight-optimized uniform sidelobe cases: 112-element full array (dash-dot line) and three realizations of random thinning to 64 elements (dashed lines). The best result is obtained for a layout-optimized 62-element thinning (solid line). 8
7 Element layouts for 112-element full array thinned to three different random 64-element layouts and a 62-element optimized layout. The random arrays are sorted according to the peak sidelobe level in Fig. 6 with Random 1 having the highest peak sidelobe level for large beamwidths. 8
8 Element layout for 64 by 64 element array first inscribed in a circle (3228 elements) and then randomly thinned by 87.5% (404 active elements). . 8
9 Beampattern for 64 by 64 element array first inscribed in a circle (3228 elements) and then thinned by 87.5% (404 active elements). The unweighted (top) and the optimally weighted (bottom) responses are shown as a function of azimuth angle and seen from the side in 3D space, i.e. the peak values over all elevation angles are shown. . 9
10 The optimization region in k-space. 9