Design of Sparse Linear Arrays by Monte Carlo Importance Sampling

Steven Kay and Supratim Saha

***Abstract*—The formation of acoustic images in real-time requires an enormous computational burden. To reduce this demand the use of sparse arrays for beamforming is mandated. The design of these arrays for adequate mainlobe width and low sidelobe level is a difficult nonlinear optimization problem. A new approach to the joint optimization of sensor placement and shading weights is discussed. Based on the concept of importance sampling an optimization method is presented and some examples given to illustrate its effectiveness.**

***Index Terms*—Acoustic imaging, global optimization methods, linear arrays, Monte Carlo methods.**

I. Introduction

In this paper, we propose a technique to design nonuniformly spaced, partially filled linear arrays for narrowband signals. A partially filled array is one in which the number of sensors is significantly less than that in a filled array. A filled array is one in which the sensors are placed every half wavelength from each other. If the sensors in a partially filled array are uniformly spaced, grating lobes are produced. These grating lobes are undesirable in most applications. The grating lobes can be reduced by placing the sensors nonuniformly [1]. This is the main motivation behind the design of nonuniformly spaced arrays. Although by nonuniform sensor placement the sidelobe levels can be reduced, there is no closed form analytical solution for finding the optimum sensor locations and the shading weights. This is partly due to the fact that the cost functions used for optimizing the positions and shading weights are highly nonlinear.

The most commonly used criterion to determine the positions of sensors in a nonuniformly spaced, partially filled array is the minimization of peak sidelobe level [2]. However, the use of this criterion leads to a typical minimax optimization problem, which does not lend itself to a closed form analytical solution. Hence, most of the reported approaches, based on minimization of peak sidelobe level, have been iterative in nature. The major drawback of an iterative approach is that there is no guarantee that the design will yield the global optimum. Also, iterative techniques are generally computationally very intensive.

Dynamic programming [3], simulated annealing [4], genetic algorithms [5] are some of the iterative techniques which have been used for the design of sparse arrays by minimizing the peak sidelobe level. Dynamic programming results in a -8.8-dB peak sidelobe level for a 3-dB beamwidth of approximately 0.014, for a 25-element array spaced over a 50λ aperture. Simulated annealing results in about a-13-dB peak sidelobe level for a 3-dB beamwidth of 0.0125 and the same number of array elements and aperture. Simulated annealing [4] achieves a lower sidelobe level than most other techniques [3], [6] but it requires a judicious choice of temperatures (used as parameters during the iterative optimization) for best results. This is another common drawback of all iterative techniques. They require a good initial choice of the variables involved in the optimization and a judicious choice of the algorithm parameters for quick convergence. The most promising approach, based on reported results is simulated annealing, but there is no guarantee that the global minimum will be attained.

Our aim in this paper is to develop a *noniterative* scheme that jointly optimizes a cost function with respect to both sensor locations and shading weights to produce a globally optimum solution. The main application of the proposed array design method is for acoustic imaging. In this scenario, the background noise incident on the sensor array is assumed to be spatially white due to reverberation over all directions. Keeping these points in mind, we do not resort to the commonly used criterion of minimizing the peak sidelobe level. Instead, we minimize the average sidelobe energy, which minimizes the contribution of the noise power received by the sensors. This is in contrast to the radar problem of a jammer interference for which the worst case approach of minimizing the peak sidelobe level is appropriate. We show that the problem of minimizing the average sidelobe energy with respect to sensor positions and shading weights involves a mixed linear and nonlinear parameter optimization. However, it can be reduced to a separable parameter optimization problem where the shading weights, which form the linear parameter vector, become a function of the sensor positions. The sensor positions are optimized first, and then from those optimized sensor positions, the optimal shading weight vector is obtained. A globally optimum solution to this problem requires a multidimensional grid search for the sensor locations. However, a grid search can be avoided by resorting to the closed form solution proposed in [7] to find the coordinates of a global optimum of a multidimensional function. The method requires an evaluation of a multidimensional integral, which at first appears impractical. But such types of integrals can be well approximated by Monte Carlo techniques [8], [9]. In particular, importance sampling has been shown to be a very powerful Monte Carlo technique, allowing multidimensional integrals to

Manuscript received June 19, 2000; revised April 28, 2002. This work was supported in part by DARPA under the sponsorship of Dr. Theo Kooij.

The authors are with the Department of Electrical and Computer Engineering, University of Rhode Island, Kingston, RI 02881 USA (e-mail: kay@ele.uri.edu).

Digital Object Identifier 10.1109/JOE.2002.804325

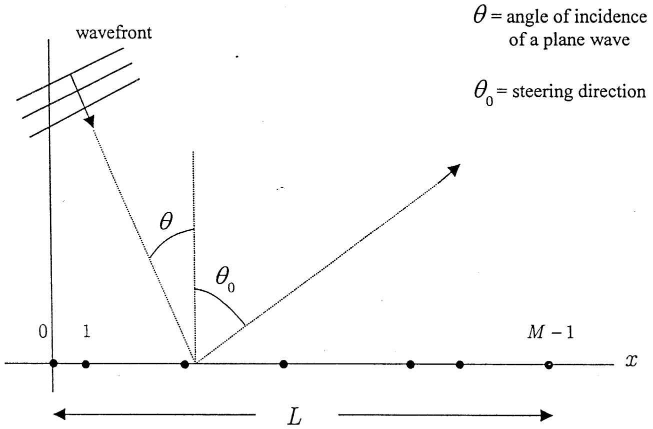

wavefront
θ = angle of incidence of a plane wave
θ_0 = steering direction
θ
θ_0
0
1
$M-1$
x
L

Fig. 1. Nonuniformly spaced array geometry.

be evaluated efficiently. We will use the importance sampling approach to obtain the globally optimum locations and weights of the sensors.

The paper is organized as follows. In Section II, we discuss the beampattern formulation for the sparse array and the constraints to be used. In Section III, minimization of average sidelobe energy jointly with respect to sensor locations and shading weights is described. In Sections IV and V, the use of importance sampling to efficiently obtain the sensors locations is described. Section VI contains some simulation results. Finally, in Section VII, we give conclusions and future directions.

II. Beampattern Formulation

Consider a linear array consisting of M omni-directional sensors placed along the x axis as shown in Fig. 1. The narrowband beampattern $B(u)$ can be expressed as

$$B(u) = \left| \sum_{i=0}^{M-1} w_i \exp\left(-j\frac{2\pi}{\lambda} x_i u\right)\right|^2 \tag{1}$$

where the x_is and w_is are the positions and real shading weights corresponding to the ith sensor, and $u = \sin\theta - \sin\theta_0$, where θ is the angle of arrival of the plane wave and θ_0 is the steering direction. The shading weights are constrained to satisfy $\sum_{i=0}^{M-1} w_i = 1$ so that $B(0) = 1$. For all possible combinations of θ and θ_0, u can assume only real values between -2 and 2. The beampattern is symmetric with respect to u so that the region of interest becomes $0 \le u \le 2$. We put another constraint on the sensor locations which has been used in some of the reported techniques [3], [4]. The constraint is that sensors can occupy only positions which are integral multiples of $\lambda/2$. From a practical standpoint this is indeed reasonable in that too closely spaced sensors result in manufacturing difficulties. As a result of this constraint on the positions of the sensors, $B(1+u) = B(1-u)$, and thus for optimizing any function of $B(u)$, the range of u that needs to be tested is $0 \le u \le 1$. It is also possible to use the design method described in Section III without a constraint on the minimum spacing between sensors. However, in this case, the minimization of sidelobe energy must be done over the interval $0 \le u \le 2$. We place two sensors at the ends of the array to maintain the aperture width and hence the desired beamwidth, and determine the locations of the remaining $M-2$ sensors and the shading weights for each of the M sensors by minimizing the average sidelobe energy. Thus, $x_0 = 0$ and $x_{M-1} = L$ and we optimize with respect to $\{x_1, \ldots, x_{M-2}\}$ and $\{w_0, \ldots, w_{M-1}\}$.

III. Joint Optimization of Positions and Shading Weights

As discussed in the introduction, our main area of application is acoustic imaging, where the background noise is assumed to be spatially white due to the presence of spatially distributed reverberation. Thus, we choose the minimization of average sidelobe energy rather than peak sidelobe level as the optimization criterion. The average sidelobe energy is defined as

$$J' = \int_{u_0}^{1} B(u)\, du \tag{2}$$

where u_0 is the starting point of the sidelobe region. If the integral in (2) is approximated by a sum, then it becomes

$$J' \approx \delta \sum_{n=p}^{N} B(n\delta) \tag{3}$$

where $p = u_0/\delta$ and $N = 1/\delta$ are assumed to be integers. Thus, from (1) and (3), we need to minimize

$$J = \sum_{n=p}^{N} \left| \sum_{i=0}^{M-1} w_i \exp\left(-j\frac{2\pi}{\lambda} x_i n\delta\right)\right|^2. \tag{4}$$

In order for the signal arriving from the look direction to remain unattenuated, we constrain $B(0) = \sum_{i=0}^{M-1} w_i = 1$. Also, recall that we have constrained two sensors to lie at the ends of the array so that $x_0 = 0$ and $x_{M-1} = L$. Thus, using $w_0 = 1 - \sum_{i=0}^{M-1} w_i$, (4) reduces to

$$J = \sum_{n=p}^{N} \left| \left(1 - \sum_{i=1}^{M-1} w_i\right) + \sum_{i=1}^{M-1} w_i \exp\left(-j\frac{2\pi}{\lambda} x_i n\delta\right)\right|^2 \tag{5}$$

$$= \sum_{n=p}^{N} \left| 1 - \sum_{i=1}^{M-1} w_i \left(1 - \exp\left(-j\frac{2\pi}{\lambda} x_i n\delta\right)\right)\right|^2. \tag{6}$$

Now, if $\mathbf{1} = [1 \cdots 1]^T$, which is of dimension $(N-p+1) \times 1$, $\boldsymbol{\theta} = [w_1 \cdots w_{M-1}]^T$, $\mathbf{x} = [x_1 \cdots x_{M-2}]^T$, and $\mathbf{H}(\mathbf{x})$ is a $(N-p+1) \times (M-1)$ matrix defined as shown in (7) at the bottom of the next page, then the approximate average sidelobe energy can be expressed as

$$J = (\mathbf{1} - \mathbf{H}(\mathbf{x})\boldsymbol{\theta})^H (\mathbf{1} - \mathbf{H}(\mathbf{x})\boldsymbol{\theta}). \tag{8}$$

Minimizing J in (8) over $\boldsymbol{\theta}$ and $\mathbf{x}$ is a mixed linear–nonlinear parameter optimization problem, where $\boldsymbol{\theta}$ is the linear parameter and $\mathbf{x}$ is the nonlinear parameter. However, this can be reduced to a separable optimization problem, where the optimal $\boldsymbol{\theta}$ is a function of the optimal $\mathbf{x}$ [10], the latter being found first. Since $\boldsymbol{\theta}$ and $\mathbf{1}$ are real and $\mathbf{H}(\mathbf{x})$ is complex, the $\boldsymbol{\theta}$ and $\mathbf{x}$ which jointly minimize J in (8) are given by

$$\hat{\boldsymbol{\theta}} = \left[\text{Real}\left(\mathbf{H}^H(\hat{\mathbf{x}})\,\mathbf{H}(\hat{\mathbf{x}})\right)\right]^{-1} \text{Real}\left(\mathbf{H}^H(\hat{\mathbf{x}})\right)\mathbf{1} \tag{9}$$

and

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} L(\mathbf{x}) \tag{10}$$

where

$$L(\mathbf{x}) = \mathbf{1}^T \text{Real}(\mathbf{H}(\mathbf{x})) \left[\text{Real}\left(\mathbf{H}^H(\mathbf{x})\mathbf{H}(\mathbf{x})\right)\right]^{-1} \cdot \text{Real}\left(\mathbf{H}^H(\mathbf{x})\right)\mathbf{1} \tag{11}$$

(see the proof in the Appendix).

To find $\hat{\mathbf{x}}$ requires a multidimensional grid search, which is not practical. Iterative techniques require a good initial guess for the parameter $\mathbf{x}$ and are also not guaranteed to converge to the global optimum. However, Pincus [7] showed that for such problems, it is possible to obtain a closed form solution for the parameter $\mathbf{x}$ that yields the *global* optimum. Based on the theorem given by Pincus the $\mathbf{x}$ that yields the *global* maximum of $L(\mathbf{x})$, is given by

$$\hat{x}_i = \lim_{\rho\to\infty} \frac{\int\cdots\int x_i \exp(\rho L(\mathbf{x}))\, d\mathbf{x}}{\int\cdots\int \exp(\rho L(\mathbf{x}))\, d\mathbf{x}} \qquad i = 1, 2, \ldots, M-2. \tag{12}$$

If we let $L'(\mathbf{x}) = \exp(\rho L(\mathbf{x}))$ and the normalized version of $L'(\mathbf{x})$ be

$$\overline{L}'(\mathbf{x}) = \frac{\exp \rho L(\mathbf{x}))\, d\mathbf{x}}{\int\cdots\int \exp(\rho L(\mathbf{x}))\, d\mathbf{x}}$$

then, the function $\overline{L}'(\mathbf{x})$ is nonnegative and has all the properties of a probability density function (PDF), although strictly speaking, it is not a PDF since $\mathbf{x}$ is not random. We term $\overline{L}'(\mathbf{x})$ a pseudo-PDF in $\mathbf{x}$. With this definition, the optimal $\mathbf{x}$ which minimizes $L(\mathbf{x})$ in (11) is

$$\hat{x}_i = \int\cdots\int x_i \overline{L}'(\mathbf{x})\, d\mathbf{x}, \qquad i = 1, 2, \ldots, M-2. \tag{13}$$

for some large value of ρ. Intuitively, as $\rho \to \infty$, the function $\overline{L}'(\mathbf{x})$ becomes a multidimensional Dirac delta function centered at the location of the maximum of $L(\mathbf{x})$. Thus, (13) yields the location of the maximum.

Now, the optimal $\mathbf{x}$ requires the evaluation of an $(M-2)$-dimensional integral, which is difficult to implement in practice. However, since $\overline{L}'(\mathbf{x})$ is a pseudo-PDF, we can interpret $\hat{x}_i$ as the expected value of x_i, where the expectation is calculated with respect to the pseudo-PDF $\overline{L}'(\mathbf{x})$. It has been shown that for this type of problem, Monte Carlo approximation techniques can achieve good results without using direct integration [11]. A straightforward Monte Carlo integration approximation can be defined as

$$\hat{\mathbf{x}} = \frac{1}{M}\sum_{k=1}^{M} \mathbf{x}_k \tag{14}$$

where $\mathbf{x}_k$ is the kth realization of the vector $\mathbf{x}$ distributed according to $\overline{L}'(\mathbf{x})$. Computing $\hat{\mathbf{x}}$ by (14) requires generation of $\mathbf{x} \sim \overline{L}'(\mathbf{x})$. The symbol $\sim$ means "is distributed according to" and indicates that $\mathbf{x}$ has the pseudo-PDF $\overline{L}'(\mathbf{x})$. For the problem of interest in this paper, generation of the vector $\mathbf{x} \sim \overline{L}'(\mathbf{x})$ may not be easy, as $\overline{L}'(\mathbf{x})$ is a highly nonlinear function of $\mathbf{x}$. So, even though direct integration can be bypassed by using (14), generation of $\mathbf{x} \sim \overline{L}'(\mathbf{x})$ may again demand integration. As a result, we do not use (14) to compute $\hat{\mathbf{x}}$. Rather, we use importance sampling [11], as described in the next section.

IV. Importance Sampling

To compute a multidimensional integral of the type given in (13), importance sampling has been shown to be a powerful tool. The approach is based on the observation that integrals of the form $\int h(\mathbf{x})\overline{L}'(\mathbf{x})\, d\mathbf{x}$ can be expressed as

$$\int h(\mathbf{x})\overline{L}'(\mathbf{x})\, d\mathbf{x} = \int h(\mathbf{x}) \frac{\overline{L}'(\mathbf{x})}{\overline{g}(\mathbf{x})} \overline{g}(\mathbf{x})\, d\mathbf{x} \tag{15}$$

where $\overline{g}(\mathbf{x})$ is assumed to possess all the properties of a PDF. Then, the right-hand side of (15) can be expressed as the expected value of $h(\mathbf{x})(\overline{L}'(\mathbf{x})/\overline{g}(\mathbf{x}))$, with respect to the pseudo-PDF $\overline{g}(\mathbf{x})$. The function $\overline{g}(\mathbf{x})$ is called the normalized importance function. Unlike $\overline{L}'(\mathbf{x})$, which in general is a complicated function of $\mathbf{x}$, $\overline{g}(\mathbf{x})$ can be chosen to be some simple function of $\mathbf{x}$, so that realizations of $\mathbf{x}$ can be easily generated. Then, the value of the integral in (15) can be found by the Monte Carlo approximation

$$\frac{1}{M}\sum_{k=1}^{M} h(\mathbf{x}_k)\frac{\overline{L}'(\mathbf{x}_k)}{\overline{g}(\mathbf{x}_k)} \tag{16}$$

where $\mathbf{x}_k$ is the kth realization of the vector $\mathbf{x}$ distributed according to the pseudo-PDF $\overline{g}(\mathbf{x})$. The value of M needed for a good approximation depends on the choice of g. Typically, $\overline{g}(\mathbf{x})$ should be chosen similar to $\overline{L}'(\mathbf{x})$, as this reduces the variance of the estimate given by (16). However, another important point to keep in mind when choosing $\overline{g}(\mathbf{x})$ is that it should be simple enough so that $\mathbf{x} \sim \overline{g}(\mathbf{x})$ can be easily generated [12], [13] .

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} 1-\exp\left(-j\,\frac{2\pi}{\lambda}\,x_1 p\delta\right) & \cdots & 1-\exp\left(-j\,\frac{2\pi}{\lambda}\,x_{M-2} p\delta\right) & 1-\exp\left(-j\,\frac{2\pi}{\lambda}\,Lp\delta\right) \\ 1-\exp\left(-j\,\frac{2\pi}{\lambda}\,x_1 (p+1)\delta\right) & \cdots & 1-\exp\left(-j\,\frac{2\pi}{\lambda}\,x_{M-2} (p+1)\delta\right) & 1-\exp\left(-j\,\frac{2\pi}{\lambda}\,L(p+1)\delta\right) \\ \vdots & \vdots & \vdots & \vdots \\ 1-\exp\left(-j\,\frac{2\pi}{\lambda}\,x_1 N\delta\right) & \cdots & 1-\exp\left(-j\,\frac{2\pi}{\lambda}\,x_{M-2} N\delta\right) & 1-\exp\left(-j\,\frac{2\pi}{\lambda}\,LN\delta\right) \end{bmatrix} \tag{7}$$

We explain in the next section how to choose $\overline{g}(\mathbf{x})$ for the position and shading weight optimization problem described in this paper.

V. Optimization by Importance Sampling

For the problem of interest, $L'(\mathbf{x})$ was shown to be

$$L'(\mathbf{x}) = \exp\left[\rho\mathbf{1}^T\text{Real}(\mathbf{H})\left(\text{Real}\left(\mathbf{H}^H\mathbf{H}\right)\right)^{-1}\text{Real}\left(\mathbf{H}^H\right)\mathbf{1}\right] \tag{17}$$

where $\mathbf{H}$ now depends on $\mathbf{x} = [x_1 \cdots x_{M-1}]^T$ as per (11). Since the length of the array is L, it follows that $0 \leq x_i \leq L$. As a result, the x_is are restricted to lie in a finite interval. Thus, the x_is can be considered to have the properties of a circular random variable [14]. The expected value of x_i with respect to the *circular* pseudo-PDF $\overline{L}'(\mathbf{x})$ is obtained using the circular mean definition

$$\hat{x}_i = \frac{1}{2\pi}\angle\int\cdots\int\exp(j2\pi x_i)\overline{L}'(\mathbf{x})\,d\mathbf{x} \tag{18}$$

where $\angle$ denotes the operation of finding the angle of the complex number. The use of (18) instead of (13) eliminates a potential bias in $\hat{x}_i$ [15], [16]. The key idea in defining a circular mean is to average position vectors. Hence, if $\theta_1, \ldots, \theta_M$ are realizations of a random point $\exp(j\theta)$ on the circumference of a circle of unit radius in the complex plane, then the sample mean of the data is defined as [14]

$$\overline{\theta} = \angle\frac{1}{M}\sum_{k=1}^{M}\exp(j\theta_k). \tag{19}$$

Thus, $\hat{x}_i$ is defined from (16) and (19) as

$$\hat{x}_i = \frac{1}{2\pi}\angle\frac{1}{M}\sum_{k=1}^{M}\frac{\overline{L}'(\mathbf{x}_k)}{\overline{g}(\mathbf{x}_k)}\exp(j2\pi[\mathbf{x}_k]_i) \tag{20}$$

for $i = 1, \ldots, M-2$, where $\mathbf{x}_k$ is the kth realization of the sensor position vector. Note that since we need only find the angle of the complex quantity in (20), an equivalent estimator is

$$\hat{x}_i = \frac{1}{2\pi}\angle\frac{1}{M}\sum_{k=1}^{M}\frac{L'(\mathbf{x}_k)}{g(\mathbf{x}_k)}\exp(j2\pi[\mathbf{x}_k]_i) \tag{21}$$

since the normalization factors cancel out. This observation is quite important in that it simplifies the computation greatly. We no longer need to find the normalization constants $\int\cdots\int L(\mathbf{x})\,d\mathbf{x}$ and $\int\cdots\int g(\mathbf{x})\,d\mathbf{x}$.

Finally, we have as our estimate

$$\hat{x}_i = \frac{1}{2\pi}\angle\frac{1}{M}\sum_{k=1}^{M}w(\mathbf{x}_k)\exp(j2\pi[\mathbf{x}_k]_i) \tag{22}$$

where

$$w(\mathbf{x}) = \frac{L'(\mathbf{x})}{g(\mathbf{x})} \tag{23}$$

and $L'(\mathbf{x})$ is given by (17) and for some appropriate $g(\mathbf{x})$ as described next.

Having expressed the estimate of the positions in (22), we need to choose an appropriate importance function $g(\mathbf{x})$ which will allow $\mathbf{x}$ to be generated easily. From (17), we observe that if Real$(\mathbf{H}^H\mathbf{H})$ is replaced by a diagonal matrix whose diagonal elements are the same as the diagonal elements of Real$(\mathbf{H}^H\mathbf{H})$, then $L'(\mathbf{x})$ becomes separable in the x_is. This is the key step in obtaining the importance function. Thus, we replace the argument of the exponential in (17) by

$$\begin{aligned}&\rho\mathbf{1}^T\text{Real}(\mathbf{H})\left[\text{diag}\left(\text{Real}\left(\mathbf{H}^H\mathbf{H}\right)\right)\right]^{-1}\text{Real}\left(\mathbf{H}^H\right)\mathbf{1}\\ &\qquad = \rho\sum_{i=1}^{M-1}\sum_{n=p}^{N}\frac{1}{2}\left(1-\cos\left(\frac{2\pi}{\lambda}x_i n\delta\right)\right)\end{aligned} \tag{24}$$

where $x_{M-1} = L$. Thus, $g(\mathbf{x})$ is chosen as

$$\begin{aligned}g(\mathbf{x}) = \exp&\left[\rho\mathbf{1}^T\text{Real}(\mathbf{H})\left[\text{diag}\left(\text{Real}\left(\mathbf{H}^H\mathbf{H}\right)\right)^{-1}\right]\right.\\ &\left.\cdot\,\text{Real}\left(\mathbf{H}^H\right)\mathbf{1}\right]\end{aligned} \tag{25}$$

and from (24), we obtain the importance function $g(\mathbf{x})$, which is now separable in the x_is or

$$\begin{aligned}g(\mathbf{x}) &= \exp\left[\rho\sum_{i=1}^{M-1}\sum_{n=p}^{N}\frac{1}{2}\left(1-\cos\left(\frac{2\pi}{\lambda}x_i n\delta\right)\right)\right]\\ &= \prod_{i=1}^{M-1}\exp(\rho I(x_i))\end{aligned} \tag{26}$$

where $I(x_i) = \sum_{n=p}^{N}(1/2)(1-\cos((2\pi/\lambda)x_i n\delta))$. It should be noted that $g(\mathbf{x})$ is a function of $x_1, \ldots, x_{M-2}$ only, because x_{M-1} is constrained to lie at the right end of the array or $x_{M-1} = L$. Furthermore, the function $g(\mathbf{x})$ is a function of the discrete variables x_is, as the sensors can occupy only those positions which are integral multiples of $\lambda/2$.

A. Generation of $\mathbf{x}$

Due to separability of $g(\mathbf{x})$ in the x_is as seen in (26), the sensor positions can be considered independent random variables. We denote the function $\exp(\rho I(x))$ of the scalar variable x as $g(x)$ (with a slight abuse of notation). As a result we have that the importance function becomes the separable function $g(\mathbf{x}) = \prod_{i=1}^{M-1} g(x_i)$. This makes the generation of the x_is quite simple. The only constraint on the x_is is that they should be distinct. Such a constraint is necessary, because in generating a sensor location vector, two of the generated locations may turn out to be nearly the same. This violates the implicit assumption that the sensor locations are distinct. If this occurs, the matrix $\mathbf{H}^H\mathbf{H}$ will be singular. In this proposed approach, the condition of distinct sensor locations and hence a full rank $\mathbf{H}$ is required in order to determine the sensor locations. We implicitly assume this in ordering $x_1 < x_2 < \cdots < x_{M-2}$ in the algorithm to be described.

Next, we show how to generate a realization of the vector $\mathbf{x}$. First we generate the position x_1 by generating $u_1 \sim U[0, 1]$, and then using the inverse transformation $x_1 = G^{-1}(u_1) =$

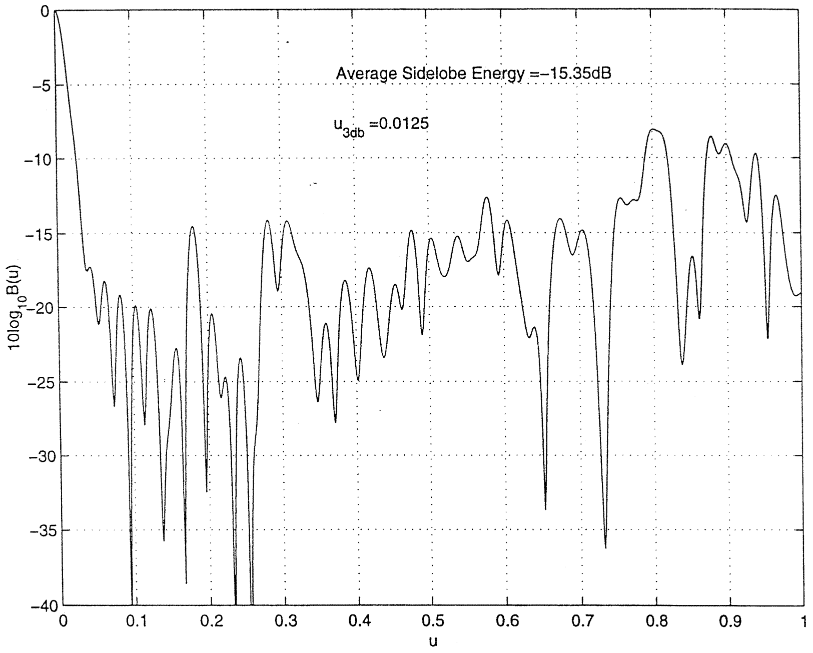


Fig. 2. Beampattern for a 3-dB beamwidth of $u_{3\text{db}} = 0.0125$.

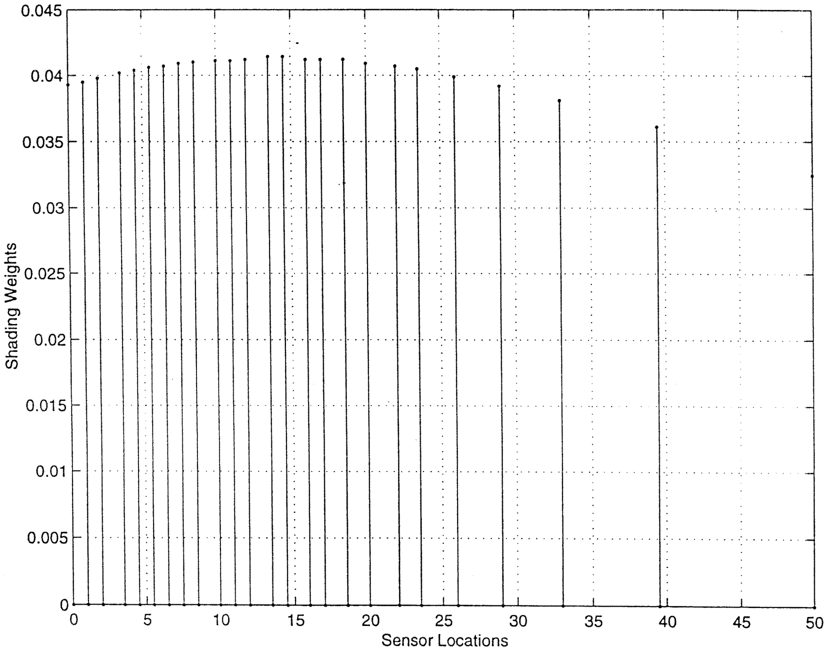


Fig. 3. Shading weights versus sensor locations for the beampattern of Fig. 2.

$\arg_x \min(u_1 - G(x))$, where $G(x)$ is the cumulative distribution function of $\overline{g}$ defined as

$$G(x) = \sum_{i=0}^{[x]} \overline{g}(\xi_i) = \sum_{i=0}^{[x]} \frac{\exp(\rho I(\xi_i))}{\sum_{k=0}^{[2L/\lambda]} \exp(\rho I(\xi_k))} \tag{27}$$

where $[x]$ denotes the largest integer less than or equal to x. This is a standard method for generating a sample distributed according to a given PDF. Once x_1 is generated as an integral multiple of $\lambda/2$, x_2 is generated by first generating $u_2 \sim U[[0, G(x_1 - (\lambda/2))] \cup [0, G(x_1 + (\lambda/2))]]$. Then, $x_2 = G^{-1}(u_2)$. Note that generating u_2 is this way guarantees $x_1 \neq x_2$. In a similar way three disjoint intervals are found based on x_1 and x_2. u_3 is then generated uniformly in the union of the three intervals, and thus, x_3 is generated. The process is continued until x_{M-2} is generated.

B. Steps of the Optimization

1) Find

$$I(x) = \frac{1}{2} \sum_{n=p}^{N} \left(1 - \cos\left(\frac{2\pi}{\lambda} x n \delta\right)\right) \tag{28}$$

for $x=\lambda/2,\lambda,\ldots,L-(\lambda/2)$ and obtain the normalized importance function as

$$\overline{g}(x)=\frac{\exp(\rho I(x))}{\sum\limits_{\xi}\exp(\rho I(\xi))}. \tag{29}$$

Here, δ has been chosen as 0.001.

2) Compute the cumulative distribution function as $G(x)=\sum_{k=0}^{[2L/\lambda]}\overline{g}(k(\lambda/2))$.

3) Generate a realization of the vector $\mathbf{x}$, say $\mathbf{x}_k$, using the procedure described in the previous section, and obtain M such realizations.

4) Compute the ith position x_i, for $i=1$ to $M-2$ using

$$\hat{x}_i=\frac{1}{2\pi}\angle\frac{1}{M}\sum_{k=1}^{M}w(\mathbf{x}_k)\exp(j2\pi[\mathbf{x}_k]_i) \tag{30}$$

where (31), shown at the bottom of the page, holds. The matrix $\mathbf{H}(\mathbf{x})$ is defined in (7). The value of ρ should in theory be infinity for a global optimum solution [7]. However, in practice ρ should be chosen as large as possible. The choice of ρ in (31) has to be made in such a way that there is little computational error, owing to the extremely large numerical values of both the numerator and denominator. Hence it is advisable to obtain $w(\mathbf{x}_k)$ from the following expression rather than (31):

$$w(\mathbf{x}_k)=\exp\left[\rho\mathbf{1}^T\text{Real}(\mathbf{H}(\mathbf{x}_k))\left[\text{Real}\left(\mathbf{H}^H(\mathbf{x}_k)\mathbf{H}(\mathbf{x}_k)\right)\right]^{-1}\right.$$
$$\left.\cdot\text{Real}\left(\mathbf{H}^H(\mathbf{x}_k)\right)\mathbf{1}-\sum_{i=1}^{M-1}\rho I([\mathbf{x}_k]_i)\right]. \tag{32}$$

From experimentation it was found that ρ in the range of $0.1\le\rho\le 0.14$ produced satisfactory results in that there was no further decrease in the average sidelobe energy as ρ increased. For values higher than 0.14, the terms in (31) became too large for machine precision. Thus, we used $\rho=0.14$ in our simulations. It should be noted that the choice of ρ is highly problem dependent.

5) The shading weight vector $\hat{\boldsymbol{\theta}}=[w_1\cdots w_{M-1}]^T$ is found as

$$\hat{\boldsymbol{\theta}}=\left[\text{Real}\left(\mathbf{H}^H(\hat{\mathbf{x}})\mathbf{H}(\hat{\mathbf{x}})\right)\right]^{-1}\text{Real}\left(\mathbf{H}^H(\hat{\mathbf{x}})\right)\mathbf{1} \tag{33}$$

and $w_0=1-\sum_{i=1}^{M-1}w_i$.

VI. Simulation Results

A sparse array was designed for 25 sensors placed over a spatial aperture of 50λ or a 25% filled array. This is a standard example which has been used for designing narrowband sparse arrays [3], [4], [6]. The length of the array used in our simulation example is 50 units. As a result, $\lambda=1$. Also, note that the 3-dB beamwidth for a filled array is about $\lambda/L=0.02$ radians or $u_{3\,\text{dB}}=0.01$.

TABLE I
Positions And Shading Weights For Design of Fig. 2 Beam Pattern

position	weight
0.0	0.0393
1.0	0.0395
2.0	0.0398
3.5	0.0402
4.5	0.0404
5.5	0.0406
6.5	0.0407
7.5	0.0409
8.5	0.0410
10.0	0.0411
11.0	0.0411
12.0	0.0412
13.5	0.0414
14.5	0.0414
16.0	0.0412
17.0	0.0412
18.5	0.0412
20.0	0.0409
22.0	0.0407
23.5	0.0405
26.0	0.0399
29.0	0.0392
33.0	0.0381
39.5	0.0361
50.0	0.0325

A total of $M=1500$ realizations were carried out to obtain the optimal positions from (30) using importance sampling. The beampattern obtained for a 3-dB beamwidth $u_{3\,\text{dB}}$ of 0.0125 is shown in Fig. 2. In order to obtain this beamwidth it was determined by “trial and error” that one should choose $u_0=0.013$. Hence, in (7) $p\delta=0.013$ is used. The average sidelobe energy (in decibels) for the beampattern is -15.35 dB. The beampattern is different from the ones obtained by minimizing the peak sidelobe level as in [4] and [6], where the sidelobes are more uniform. From Fig. 2, it can be observed that there are 3 local sidelobe peaks which exceed -10 dB with the remaining local peaks all below -10 dB. In Fig. 3, a plot of the shading weights versus the sensor locations is shown and they are tabulated in Table I. The average sidelobe energy using the technique proposed in [4] was -15.18 dB for a 3-dB beamwidth of 0.0155,

$$w(\mathbf{x}_k)=\frac{\exp\left[\rho\mathbf{1}^T\text{Real}(\mathbf{H}(\mathbf{x}_k))\left[\text{Real}\left(\mathbf{H}^H(\mathbf{x}_k)\mathbf{H}(\mathbf{x}_k)\right)\right]^{-1}\text{Real}(\mathbf{H}^H(\mathbf{x}_k))\mathbf{1}\right]}{\prod\limits_{i=1}^{M-1}\exp[\rho I([\mathbf{x}_k]_i)]} \tag{31}$$

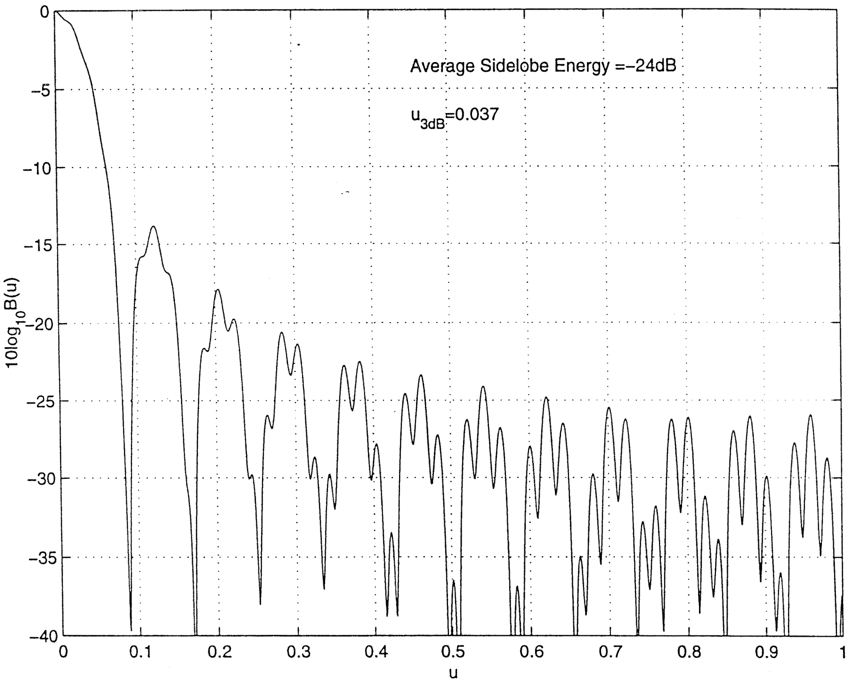


Fig. 4. Beampattern for an increased 3-dB beamwidth of $u_{3dB} = 0.037$.

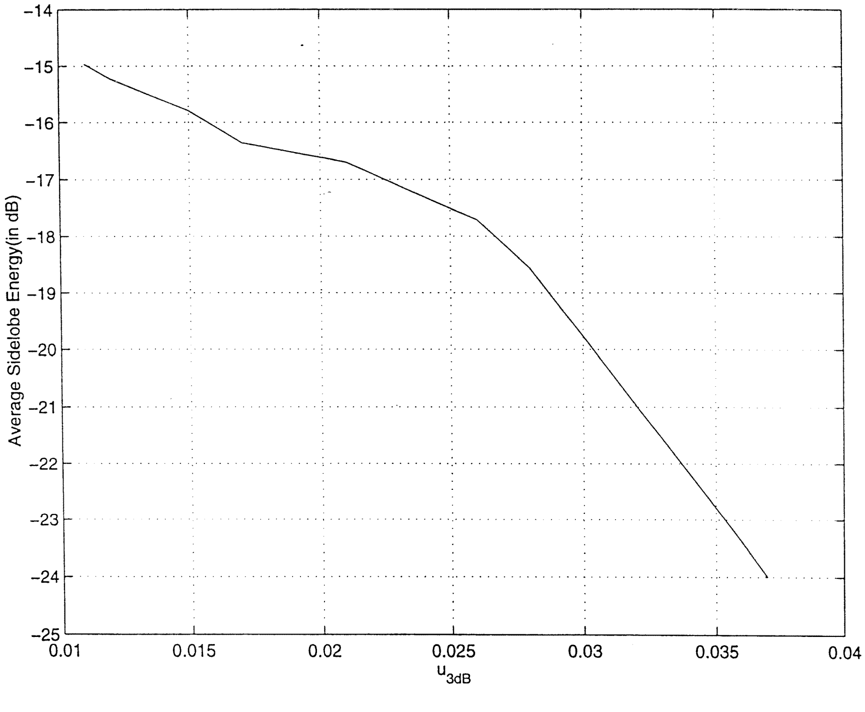


Fig. 5. Plot of average sidelobe energy versus 3-dB beamwidth.

whereas our technique produced −16.01 dB for the same 3-dB beamwidth.

In Fig. 4, we show a beampattern obtained by increasing the 3-dB beamwidth $u_{3\,\mathrm{dB}}$ by about a factor of 3. The value of u_0 required to yield this beamwidth was 0.034, again determined by "trial and error." Note that the number of sensors remain the same as before, being 25, and the length of the array is the same, being $L = 50\lambda$. The increase in beamwidth allows the sidelobe energy to be lower as some of the energy can now be located in the mainlobe. The average sidelobe energy for this beampattern was found to be −24 dB. Thus, with an increase in the mainlobe width, the sidelobe energy goes down by about 10 dB. The peak sidelobes are also reduced compared to the previous example. To get an idea of the variation of the average sidelobe energy with the mainlobe width, we plot average sidelobe energy versus the 3-dB beamwidth in Fig. 5. It is clear from the plot that the average sidelobe energy decreases with increasing mainlobe width, as expected.

Fig. 6 shows the average sidelobe energy for a given 3-dB beamwidth against the number of realizations, for 3 different

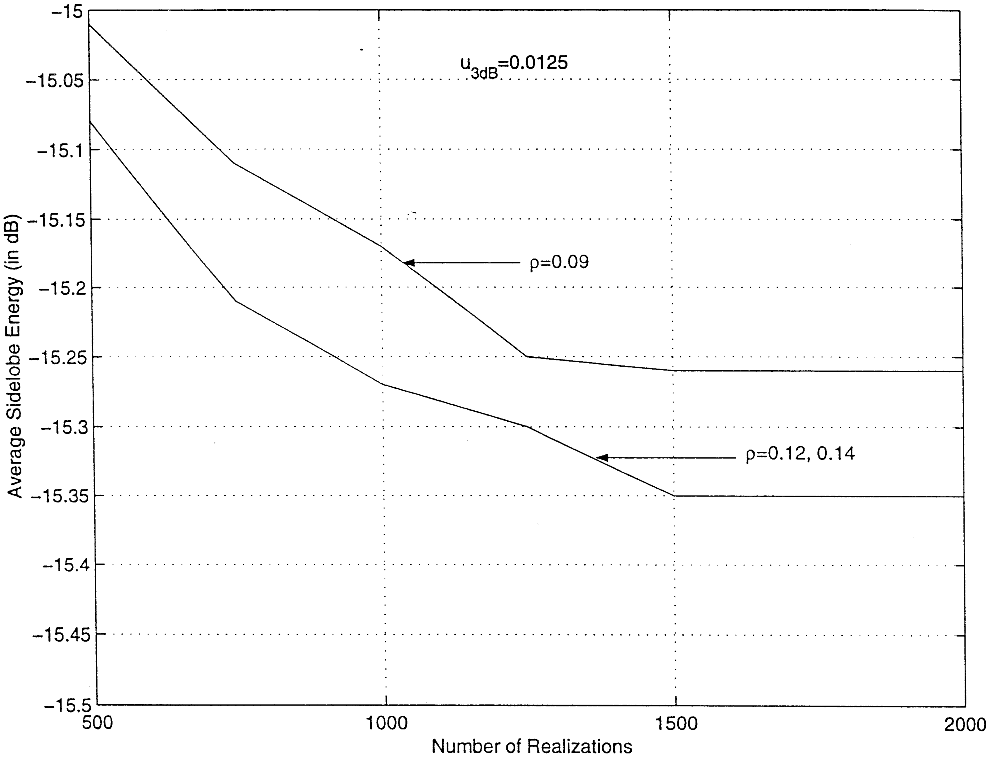


Fig. 6. Plot of average sidelobe versus number of importance sampling realizations.

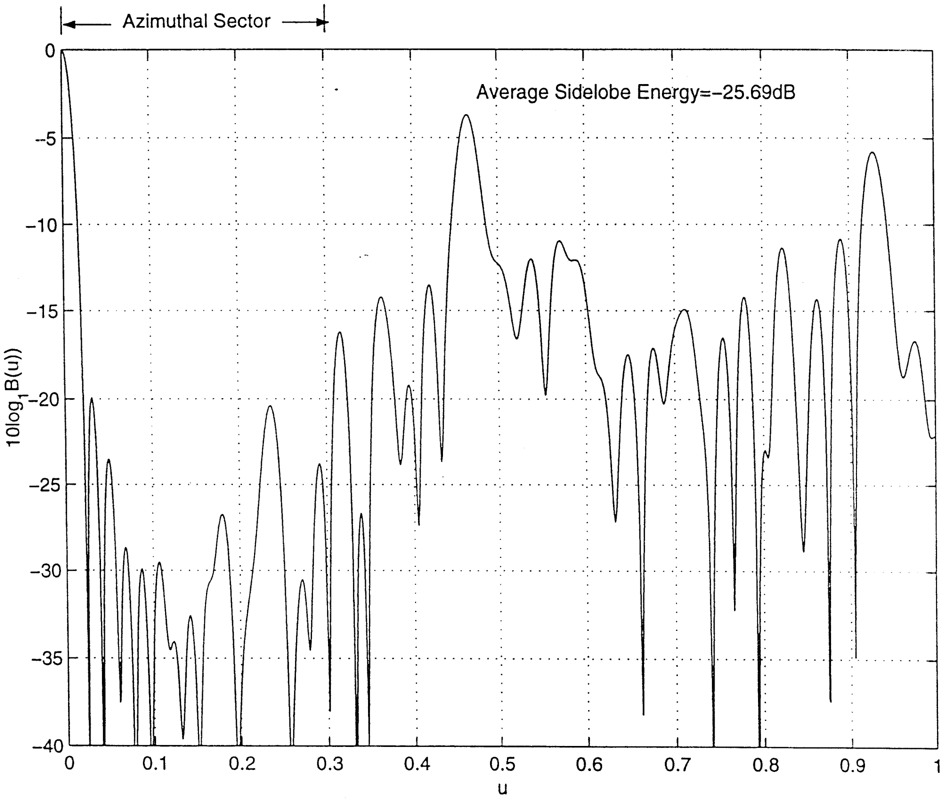


Fig. 7. Beampattern obtained by reducing sidelobe energy for a limited azimuth.

values of ρ. From this plot it is clear that only around 1500 Monte Carlo realizations are necessary for the design to converge to the optimum solution. It can be observed that the minimum average sidelobe energy does not change for ρ between 0.12 and 0.14. Also, it has been observed that by varying ρ in the range (0.11, 0.14) there is no change in the final sensor locations. Thus it can be concluded that the design converges to the global optimum.

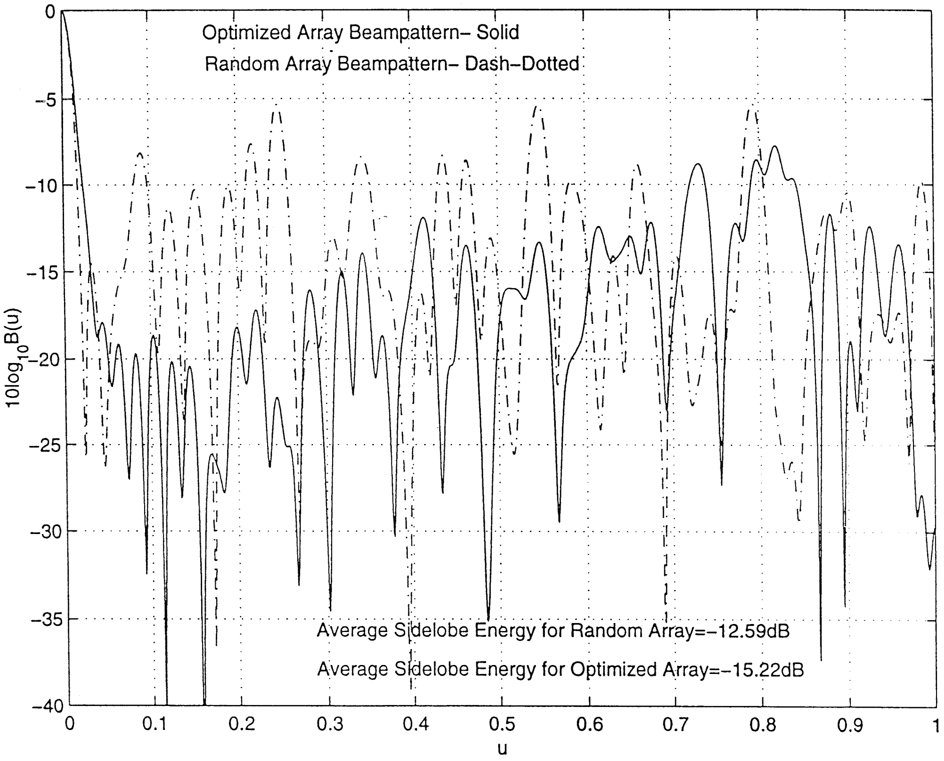
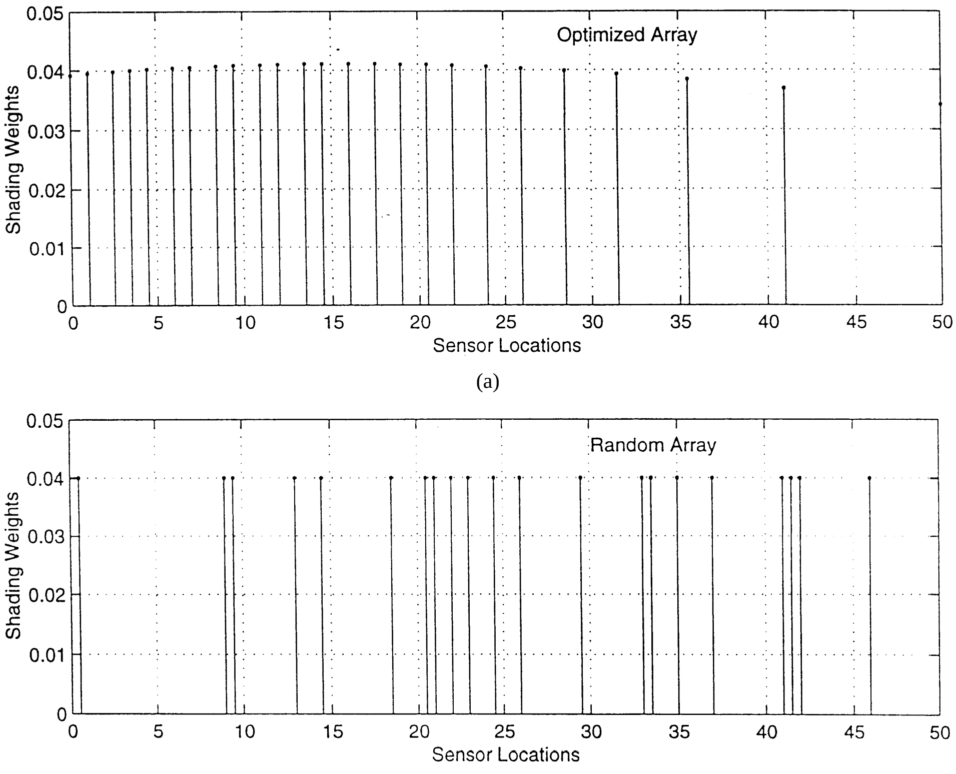


Fig. 8. Design results (beampatterns) comparing random array and importance sampling.



(a)

(b)

Fig. 9. Design resulst (shading weights versus sensor locations) for random array and importance sampling.

To illustrate the utility of the technique for cases when no acoustic energy arrives outside of given azimuthal sector, we show in Fig. 7 the beampattern obtained by minimizing the average sidelobe energy in the range $0.01 < u < 0.3$. The average sidelobe energy is reduced to -25.69 dB, without any increase in mainlobe width.

Finally, to compare the design results of the proposed technique with that of a random array, we show in Fig. 8, a beampattern overlaid with that of a beampattern obtained for a typical random array. In order to make a fair comparison the beamwidth of the random array was first determined. Then, the appropriate value of u_0 was determined for the optimal array so that its beamwidth matched that of the random array. Hence, the resultant beampattern is slightly different than that shown in Fig. 2, where the value of u_0 used was slightly different. The shading weights versus sensor locations for the proposed technique and the uniformly shaded random array and are shown in Fig. 9.

VII. Conclusion

We have developed a method for designing a nonuniformly spaced linear array by minimizing the average sidelobe energy. The design is guaranteed to converge to the global optimum. It jointly optimizes over the shading weights as well as the sensor locations. The technique is also computationally modest, compared to conventional iterative techniques. The proposed approach can be extended to planar arrays and is currently under investigation.

Appendix
Derivation of $L(\mathbf{x})$ in (11)

Let

$$J = (\mathbf{y} - \mathbf{H}(\mathbf{x})\boldsymbol{\theta})^H(\mathbf{y} - \mathbf{H}(\mathbf{x})\boldsymbol{\theta}) \tag{34}$$

where $\mathbf{y} = \mathbf{1}$, and is real, $\boldsymbol{\theta}$ is real, and $\mathbf{H}(\mathbf{x})$ is complex. Expanding J we have

$$J = \mathbf{y}^T\mathbf{y} - \boldsymbol{\theta}^T\mathbf{H}^H(\mathbf{x})\mathbf{y} - \mathbf{y}^T\mathbf{H}(\mathbf{x})\boldsymbol{\theta} + \boldsymbol{\theta}^T\mathbf{H}^H(\mathbf{x})\mathbf{H}(\mathbf{x})\boldsymbol{\theta}. \tag{35}$$

Since $\mathbf{y}^T\mathbf{H}(\mathbf{x})\boldsymbol{\theta} = [\mathbf{y}^T\mathbf{H}(\mathbf{x})\boldsymbol{\theta}^T = \boldsymbol{\theta}^T\mathbf{H}^T(\mathbf{x})\mathbf{y}$, we have

$$J = \mathbf{y}^T\mathbf{y} - \boldsymbol{\theta}^T\mathbf{H}^H(\mathbf{x})\mathbf{y} - \boldsymbol{\theta}^T\mathbf{H}^T(\mathbf{x})\mathbf{y} + \boldsymbol{\theta}^T\mathbf{H}^H(\mathbf{x})\mathbf{H}(\mathbf{x})\boldsymbol{\theta}. \tag{36}$$

Now, $\mathbf{H}^H(\mathbf{x})\mathbf{H}(\mathbf{x}) = \mathbf{A}(\mathbf{x}) + j\mathbf{B}(\mathbf{x})$ where $\mathbf{A}(\mathbf{x}) = \mathbf{A}^T(\mathbf{x})$ and $\mathbf{B}(\mathbf{x}) = -\mathbf{B}^T(\mathbf{x})$. Expressing the last term in (36) as a function of $\mathbf{A}(\mathbf{x})$ and $\mathbf{B}(\mathbf{x})$ we have

$$\begin{aligned} J &= \mathbf{y}^T\mathbf{y} - \boldsymbol{\theta}^T\left(\mathbf{H}^H(\mathbf{x}) + \mathbf{H}^T(\mathbf{x})\right)\mathbf{y} + \boldsymbol{\theta}^T\mathbf{A}(\mathbf{x})\boldsymbol{\theta} + j\boldsymbol{\theta}^T\mathbf{B}(\mathbf{x})\boldsymbol{\theta} \\ &= \mathbf{y}^T\mathbf{y} - 2\boldsymbol{\theta}^T\mathrm{Real}\left(\mathbf{H}^H(\mathbf{x})\right)\mathbf{y} + \boldsymbol{\theta}^T\mathbf{A}(\mathbf{x})\boldsymbol{\theta} + j\boldsymbol{\theta}^T\mathbf{B}(\mathbf{x})\boldsymbol{\theta}. \end{aligned} \tag{37}$$

As $\mathbf{B}(\mathbf{x})$ is skew-symmetric, the last term in (37) is zero since $\boldsymbol{\theta}^T\mathbf{B}(\mathbf{x})\boldsymbol{\theta} = (\boldsymbol{\theta}^T\mathbf{B}(\mathbf{x})\boldsymbol{\theta})^T = \boldsymbol{\theta}^T\mathbf{B}(\mathbf{x})^T\boldsymbol{\theta} = -\boldsymbol{\theta}^T B(\mathbf{x})\boldsymbol{\theta}$. Now

$$J = \mathbf{y}^T\mathbf{y} - 2\boldsymbol{\theta}^T\mathrm{Real}\left(\mathbf{H}^H(\mathbf{x})\right)\mathbf{y} + \boldsymbol{\theta}^T\mathrm{Real}\left(\mathbf{H}^H(\mathbf{x})\mathbf{H}(\mathbf{x})\right)\boldsymbol{\theta}. \tag{38}$$

For a given $\mathbf{x}$ and hence a given $\mathbf{H}(\mathbf{x})$, $\partial J/\partial\boldsymbol{\theta}$ is

$$\frac{\partial J}{\partial\boldsymbol{\theta}} = -2\mathrm{Real}\left(\mathbf{H}^H(\mathbf{x})\right)\mathbf{y} + 2\mathrm{Real}\left(\mathbf{H}^H(\mathbf{x})\mathbf{H}(\mathbf{x})\right)\boldsymbol{\theta}. \tag{39}$$

Setting the derivative of J with respect to the vector $\boldsymbol{\theta}$ to zero, we obtain

$$\hat{\boldsymbol{\theta}} = \left[\mathrm{Real}\left(\mathbf{H}^H(\mathbf{x})\mathbf{H}(\mathbf{x})\right)\right]^{-1}\mathrm{Real}\left(\mathbf{H}^H(\mathbf{x})\right)\mathbf{y}. \tag{40}$$

For this $\boldsymbol{\theta}$, J becomes

$$\begin{aligned} J = \mathbf{y}^T\Big(\mathbf{I} - \mathrm{Real}(\mathbf{H}(\mathbf{x}))\left[\mathrm{Real}\left(\mathbf{H}^H(\mathbf{x})\mathbf{H}(\mathbf{x})\right)\right]^{-1} \\ \cdot\ \mathrm{Real}\left(\mathbf{H}^H(\mathbf{x})\right)\mathbf{y}. \end{aligned} \tag{41}$$

Minimizing J, with respect to $\mathbf{x}$ is thus equivalent to a maximization of

$$\mathbf{y}^T\mathrm{Real}(\mathbf{H}(\mathbf{x}))\left[\mathrm{Real}\left(\mathbf{H}^H(\mathbf{x})\mathbf{H}(\mathbf{x})\right)\right]^{-1}\mathrm{Real}(\mathbf{H}^H(\mathbf{x}))\mathbf{y}.$$

References

[1] B. Steinberg, *Principles of Aperture and Array Systems*. New York: Wiley, 1976.

[2] D. K. Cheng, "Optimization techniques for antenna arrays," *Proc. IEEE*, vol. 59, pp. 1664–1674, Dec. 1971.

[3] M. I. Skolnik, Nemhauser, and J. W. Sherman, "Dynamic programming applied to unequally spaced arrays," *IEEE Trans. Antennas Propagat.*, pp. 35–43, 1964.

[4] V. Murino, A. Trucco, and C. S. Regazzoni, "Synthesis of unequally spaced arrays by simulated annealing," *IEEE Trans. Signal Processing*, vol. 36, Jan. 1996.

[5] K. K. Yan and Y. Lu, "Sidelobe reduction in array-pattern synthesis using genetic algorithms," *IEEE Trans. Antennas Propagat.*, vol. 45, pp. 1117–1121, July 1997.

[6] P. Jarske, T. Saramaki, S. K. Mitra, and Y. Neovo, "On properties and design of nonuniformly spaced linear arrays," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 372–380, Mar. 1988.

[7] M. Pincus, "A closed form solution for certain programming problems," *Oper. Res.*, pp. 690–694, 1962.

[8] L. Stewart, "Bayesian analysis using Monte Carlo integration: A powerful methodology for handling some difficult problems," *Amer. Stat.*, pp. 195–200, 1983.

[9] A. F. M. Smith and A. Gelfand, "Bayesian statistics without tears: A sampling–resampling framework," *Amer. Stat. Assoc.*, 1992.

[10] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[11] J. M. Bernardo and A. F. M. Smith, *Bayesian Theory*. New York: Wiley, 1994.

[12] R. Y. Rubinstein, *Monte Carlo Optimization, Simulation, and Sensitivity of Queuing Networks*. New York: Wiley, 1986.

[13] M. H. Kalos and P. A. Whitlock, *Monte Carlo Methods*. New York: Wiley, 1986.

[14] K. V. Mardia, *Statistics of Directional Data*. New York: Academic, 1972.

[15] B. C. Lowell, P. J. Kootsookos, and R. C. Williamson, "The circular nature of discrete-time frequency estimates," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, May 1991, pp. 3369–3372.

[16] S. Kay and S. Saha, "Mean likelihood frequency estimation," *IEEE Trans. Signal Processing*, vol. 48, pp. 1937–1946, July 2000.

Steven Kay was born in Newark, NJ, on April 5, 1951. He received the B.E. degree from Stevens Institute of Technology, Hoboken, NJ in 1972, the M.S. degree from Columbia University, New York, NY, in 1973, and the Ph.D. degree from Georgia Institute of Technology, Atlanta, GA, in 1980, all in electrical engineering.

From 1972 to 1975, he was with Bell Laboratories, Holmdel, NJ, where he was involved with transmission planning for speech communications and simulation and subjective testing of speech processing algorithms. From 1977 to 1980, he was with the Submarine Signal Division, Portsmouth, RI, where he engaged in research on autoregressive spectral estimation and the design of sonar systems. He is presently Professor of Electrical Engineering at the University of Rhode Island, Kingston, and a Consultant to industry and the Navy. He has written numerous papers and is a contributor to several edited books. He is the author of the textbooks Modern Spectral Estimation (NJ: Englewood Cliffs, Prentice-Hall, 1988), Fundamentals of Statistical Signal Processing, Vol. I: Estimation Theory (NJ: Englewood Cliffs, Prentice-Hall, 1993), and Fundamentals of Statistical Signal Processing, Vol. II: Detection Theory (NJ: Englewood Cliffs, Prentice-Hall, 1998). His current interests are spectrum analysis, detection and estimation theory, and statistical signal processing.

Supratim Saha received the B.Tech degree from Indian Insitute of Technology, Kharagpur, India in 1993, the M.S. degree from Indian Institute of Science, Bangalore, India, in 1996, and the Ph.D. degree from the University of Rhode Island in 2001, all in electrical engineering.

His interests are in probability theory, and communication systems.