A Numerical Pattern Synthesis Algorithm for Arrays

CARL A. OLEN, MEMBER, IEEE, AND R. T. COMPTON, JR., FELLOW, IEEE

Abstract—**A numerical technique for pattern synthesis in arrays is presented. For a given set of elements, this technique allows one to find a set of array coefficients that steer the main beam in a given direction and yield sidelobes meeting a specified criterion, if such a set of array coefficients exists. If the pattern specifications cannot be met with the given elements, the algorithm finds the best attainable pattern. The advantage of this technique is that it can be used with an arbitrary set of array elements. Different elements in the array can have different element patterns, and the array can have arbitrary nonuniform spacing between elements. The synthesis technique is based on adaptive array theory. The given array elements are assumed to be used as the elements of an adaptive array. The main beam is pointed in the proper direction by choosing the steering vector for that direction. The sidelobes are controlled by introducing a large number of interfering signals at many angles throughout the sidelobe region. The algorithm iterates on the interference powers until a suitable pattern is obtained.**

I. Introduction

THE PROBLEM OF synthesizing antenna patterns with low sidelobes has received much attention over the years. In a famous paper, Taylor [1] obtained an aperture distribution that produces equal height sidelobes near the main beam and tapered far out sidelobes. Hyneman and Johnson [2] presented a technique for controlling pattern behavior by moving zeros in the pattern function. Hyneman [3] also considered the problem of synthesizing patterns with a nonuniform sidelobe envelope. Additional contributions to the pattern synthesis problem for continuous apertures include the work of Elliott [4]–[6], and White [7].

For arrays, Dolph's classic paper [8] derived the array weights for a uniformly spaced linear array that yields minimum beamwidth for a given maximum sidelobe level. He showed that the resulting uniform sidelobe patterns are derived from Chebyshev polynomials. Villeneuve [9] showed how Taylor's method [1] can be applied to arrays. Elliot and Stern [10] have also described additional pattern synthesis techniques for arrays.

An important feature of all of these papers is that the results are applicable only to arrays of uniformly spaced, isotropic elements. Some of the methods, such as those in Hyneman [3] and Elliott and Stern [10], allow one to design an array whose sidelobe envelope varies with angle. But none can address design problems in which the elements are nonuniformly spaced, the element patterns are unequal, or the elements do not lie along a straight line.

Our purpose in this paper is to describe a simple numerical synthesis technique that can be used with arrays consisting of a set of arbitrary elements. This technique allows one to find a set of weights that will steer a beam in a given direction and will meet an arbitrary sidelobe specification in other directions.

The underlying approach is to assume that the given array elements are used as elements of an adaptive array [11], [12]. The main beam is steered in the desired direction by choosing the steering vector for that direction. To reduce the sidelobes, a large number of interfering signals is assumed to be incident on the array from the sidelobe region. On a computer, one solves for the resulting adapted weights. The adapted pattern is then computed and compared with the design objectives. At any angle where the sidelobes are too high or too low, the interference power is increased or decreased accordingly and then the weights are recalculated. This process is repeated iteratively until a suitable final pattern is obtained. The final adapted weights are then used as the design weights for the actual (nonadaptive) array.

The use of adaptive array theory for pattern synthesis has been suggested previously in two earlier papers. In the first, Sureau and Keeping [13] used an adaptive array approach to find the weights for an array on a cylinder. They tapered the interference spectrum (its power versus arrival angle) to control the sidelobes. However, they did not provide any systematic way of choosing the interference spectrum to meet a given sidelobe requirement. In another interesting paper, Dufort [14] also suggested using adaptive array theory for pattern design. To obtain the desired sidelobe behavior, Dufort adopted the clever approach of making the interference spectrum equal to the reciprocal of the desired pattern power pattern. This approach was based on the idea that the product of the interference spectrum and the power pattern is constant with angle, an assumption that holds approximately, but not exactly. Using this approach, Dufort obtained the required array weights analytically for uniformly spaced arrays of isotropic elements by minimizing the array output power or maximizing the output signal-to-noise ratio.

The approach we describe here differs from Dufort's in that we obtain the interference spectrum needed to produce the desired sidelobe behavior by using a recursive feedback procedure. We force the pattern to converge to the desired result by iterating on the interference spectrum. Because this

Manuscript received September 20, 1989; revised April 9, 1990. This work was supported by the Joint Services Electronics Program under Contract N00014-89-J-1007.

The authors are with the ElectroScience Laboratory, Department of Electrical Engineering, The Ohio State University, Columbus, OH 43212.

IEEE Log Number 9037643.

0018-926X/90/1000-1666$01.00

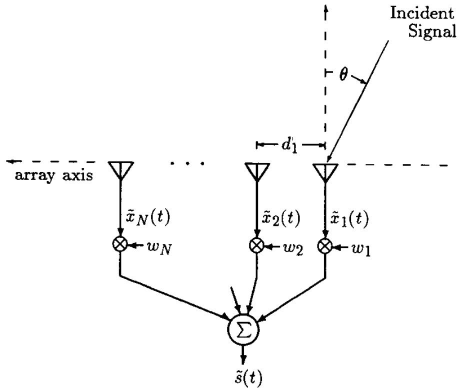


Fig. 1. An N-element linear array.

method is a numerical technique, it does not yield analytic solutions for the weights. However, being a numerical technique, it can be used with much more general types of problems than an analytical approach. The method easily handles arrays in which the element patterns of different elements are different and the element locations are arbitrary. It can be used to obtain patterns whose sidelobe levels vary arbitrarily with angle.

The paper is organized as follows. Section II defines our terminology and discusses the adaptive array model. Section III presents the iterative pattern design technique. Section IV illustrates the use of this method in some different examples of design problems. Finally, Section V contains our conclusions.

II. Formulation

Consider an N-element linear array as shown in Fig. 1. Let $f_j(\theta)$ be the pattern of element j and d_j the distance in wavelengths between element j and element $j+1$. Assume a single frequency (CW) signal is incident on the array from angle θ, where θ is measured from broadside as shown in Fig. 1. Let $\tilde{x}_j(t)$ be the received (analytic) signal on element j and define the signal vector $\mathbf{X}$ as

$$\mathbf{X} = \left[\tilde{x}_1(t), \tilde{x}_2(t), \cdots, \tilde{x}_N(t)\right]^T, \tag{1}$$

where superscript T denotes the transpose. The array output signal $\tilde{s}(t)$ is obtained by multiplying each $\tilde{x}_j(t)$ by a complex weight w_j and then summing,

$$\tilde{s}(t) = \sum_{j=1}^{N} w_j \tilde{x}_j(t) = \mathbf{W}^T\mathbf{X}, \tag{2}$$

where $\mathbf{W}$ is the weight vector,

$$\mathbf{W} = \left[w_1, w_2, \cdots, w_N\right]^T. \tag{3}$$

If the incoming CW signal is at frequency ω_0, $\mathbf{X}$ is given by

$$\mathbf{X} = Ae^{j\omega_0 t}\mathbf{U}, \tag{4}$$

where A is the signal amplitude, $e^{j\omega_0 t}$ is its time dependence, and $\mathbf{U}$ is a vector containing the interelement phase shifts and the element patterns,

$$\mathbf{U} = \left[f_1(\theta), f_2(\theta)e^{-j\phi_2(\theta)}, f_3(\theta)e^{-j\phi_3(\theta)}, \cdots, f_N(\theta)e^{-j\phi_N(\theta)}\right]^T, \tag{5}$$

with

$$\phi_j(\theta) = 2\pi\left[\sum_{k=1}^{j-1} d_k\right]\sin\theta, \qquad j \geq 2. \tag{6}$$

Combining (4) and (2) gives the array output $\tilde{s}(t)$,

$$\tilde{s}(t) = Ae^{j\omega_0 t}\mathbf{W}^T\mathbf{U}. \tag{7}$$

We define the array voltage pattern $p(\theta)$ to be

$$p(\theta) = |\mathbf{W}^T\mathbf{U}|. \tag{8}$$

Given an array pattern with a beam in some direction, let $P = \max_\theta\{p(\theta)\}$ be the magnitude of the beam peak. Denote the first nulls (or minima) to the left and right of the beam peak by θ_L and θ_R, respectively. The beam region is then the region $\theta_L \leq \theta \leq \theta_R$, and the sidelobe region is the region outside the beam region.

The problem we consider is the following. We assume that the number of array elements N, the element patterns, $f_j(\theta)$, and the element spacings d_j are all given. We want to find a weight vector $\mathbf{W}$ for which $p(\theta)$ has a beam maximum at some angle θ_d and meets a given sidelobe specification at other angles.

To do this, we shall imagine that the array is used as an adaptive array [11], [12]. An adaptive array is an antenna system that controls its own pattern by adjusting the weight vector $\mathbf{W}$ in response to the signal environment. The weight vector is adjusted to maximize the array output desired signal-to-interference-plus-noise power ratio (SINR). Maximum SINR is obtained, roughly speaking, by maximizing the pattern in the desired signal direction and minimizing it in interference directions.

The array weights that maximize SINR may be calculated as follows. Suppose a desired signal and an interfering signal are incident on the array and suppose each element signal also contains thermal noise. The total signal vector $\mathbf{X}$ is then

$$\mathbf{X} = \mathbf{X}_d + \mathbf{X}_i + \mathbf{X}_n = \mathbf{X}_d + \mathbf{X}_u, \tag{9}$$

where $\mathbf{X}_d$, $\mathbf{X}_i$ and $\mathbf{X}_n$ are the desired, interference, and noise vectors, respectively, and where $\mathbf{X}_u = \mathbf{X}_i + \mathbf{X}_n$ is the undesired part of $\mathbf{X}$. The optimal weights are then given by [11], [12]

$$\mathbf{W} = \mu\Phi_u^{-1}\mathbf{U}_d^*, \tag{10}$$

where Φ_u is the covariance matrix of the undesired signals,

$$\Phi_u = E\{\mathbf{X}_u^*\mathbf{X}_u^T\}. \tag{11}$$

In these equations, $E\{\cdot\}$ denotes expectation and the asterisk, complex conjugate. The factor μ is an arbitrary nonzero scalar.

Specifically, suppose the desired signal is a CW signal arriving from angle θ_d. $\mathbf{X}_d$ is then given by

$$\mathbf{X}_d = A_d e^{j(\omega_0 t + \psi_d)}\mathbf{U}_d, \tag{12}$$

where A_d is the desired signal amplitude, ψ_d is the carrier phase angle, and $\mathbf{U}_d$ is given by (5) and (6) with $\theta = \theta_d$. The phase ψ_d is assumed to be a random variable uniformly distributed on $(0, 2\pi)$. In addition, suppose the interference is also a CW signal arriving from angle θ_i. $\mathbf{X}_i$ is then

$$\mathbf{X}_i = A_i e^{j(\omega_0 t + \psi_i)} \mathbf{U}_i, \tag{13}$$

where A_i is the interference amplitude, ψ_i is its carrier phase, and $\mathbf{U}_i$ is given by (5) and (6) with $\theta = \theta_i$. The angle ψ_i is also assumed to be a random variable uniform distributed on $(0, 2\pi)$ and independent of ψ_d. Finally, suppose the thermal noise vector $\mathbf{X}_n$ is given by

$$\mathbf{X}_n = \left[\tilde{n}_1(t), \tilde{n}_2(t), \cdots, \tilde{n}_N(t)\right]^T, \tag{14}$$

where the noise voltages $\tilde{n}_j(t)$ are narrow-band random processes, statistically independent of ψ_d, ψ_i, and each other, and each of power σ^2. Then

$$E\{\tilde{n}_i^*(t)\tilde{n}_j(t)\} = \sigma^2 \delta_{ij}, \tag{15}$$

where δ_{ij} is the Kronecker delta. When these signal vectors are used in (9) and (11), it is found that

$$\Phi_u = \sigma^2 \mathbf{I} + A_i^2 \mathbf{U}_i^* \mathbf{U}_i^T. \tag{16}$$

The optimal weight vector $\mathbf{W}$ can then be calculated from (10).

For later use, it is helpful to define the signal-to-noise ratio (SNR) per element ξ_d,

$$\mathrm{SNR} = \xi_d = \frac{A_d^2}{\sigma^2}, \tag{17}$$

and also the interference-to-noise ratio (INR) per element ξ_i,

$$\mathrm{INR} = \xi_i = \frac{A_i^2}{\sigma^2}. \tag{18}$$

These are the signal-to-noise ratios that would exist on an isotropic element (i.e., if $f_j(\theta) = 1$).

The response of an adaptive array pattern to interfering signals depends on the number of interfering signals in relation to the number of degrees of freedom in the array. An N-element array has $N - 1$ degrees of freedom in its pattern [12]. One degree of freedom is needed to form a pattern maximum on the desired signal. The remaining $N - 2$ degrees are available to null interference signals. If $N - 2$ or fewer interference signals are incident on the array, the array usually forms a null on each interference signal. However, if more than $N - 2$ interference signals are incident, the array does not normally null the individual interference signals but instead forms a compromise pattern that minimizes the total interference power at the array output.

The response of an adaptive array to an interference signal also depends on the interference signal strength. The stronger the interference, the lower the adapted pattern level. Figs. 2 and 3 illustrate this behavior. In Fig. 2 a single interference signal is incident on a ten-element adaptive array from $\theta_i = -30°$. (The elements are isotropic and spaced every half-wavelength; the desired signal is at $\theta_d = 0°$.) In Fig. 2(a), the INR is -10 dB, in Fig. 2(b) it is 0 dB, and in Fig. 2(c) it is $+10$ dB. Note how the adapted pattern level at $\theta = -30°$ decreases as the interference power is increased.

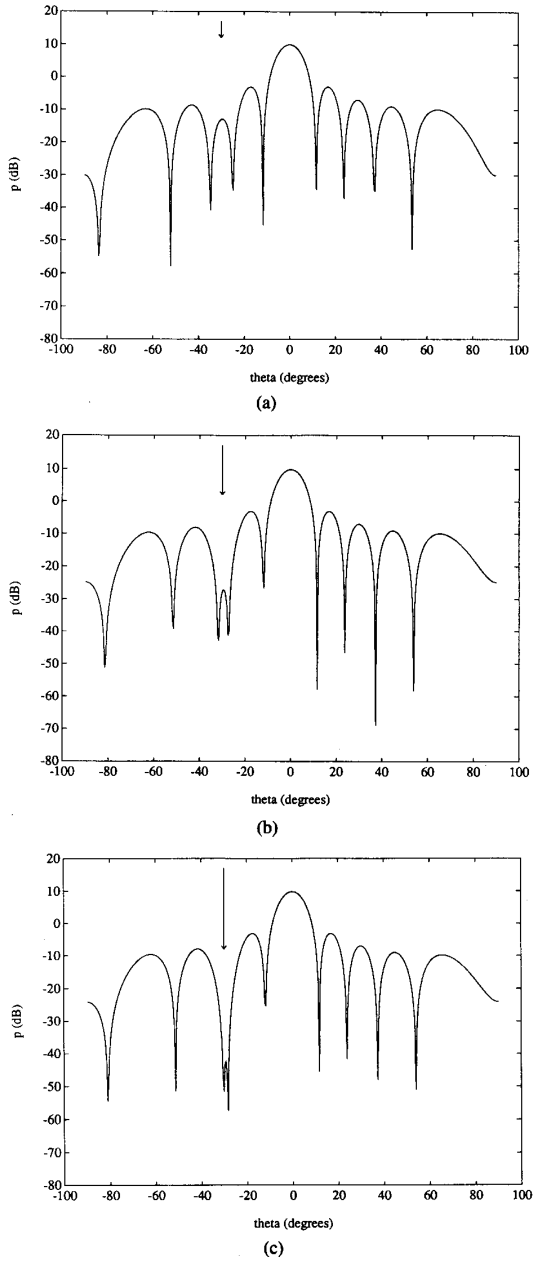
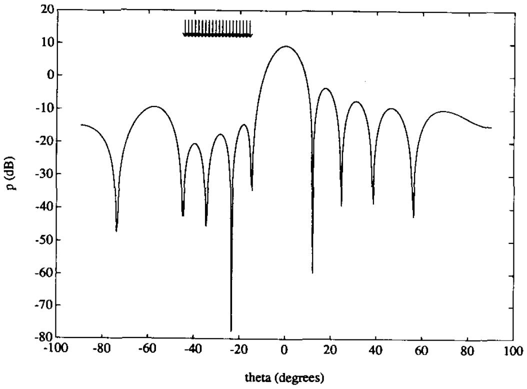
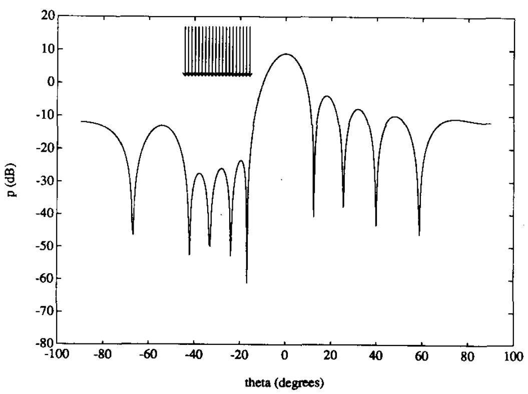


Fig. 2. Adapted patterns with one interference signal. (a) INR = -10 dB. (b) INR = 0 dB. (c) INR = $+10$ dB.

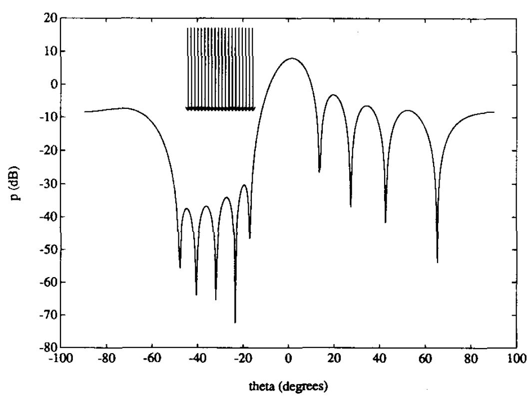
Fig. 3 shows what happens when there are more interference signals incident than the number of degrees of freedom in the array. In Fig. 3, 20 interference signals are incident on the same ten-element array from a 28.5° angular region centered at $\theta = -30°$. The INR (for each interference signal) is again varied from -10 to $+10$ dB. Note again how the sidelobe level drops as the INR is increased. The pattern design algorithm described below takes advantage of this behavior to force the sidelobes down in an array.



(a)



(b)



(c)

Fig. 3. Adapted patterns with 20 interference signals. (a) INR per signal = − 10 dB. (b) INR per signal = 0 dB. (c) INR per signal = 10 dB.

III. The Sidelobe Control Algorithm

A. The Algorithm

To find a weight vector giving an acceptable pattern with the given set of elements, we proceed as follows. First, the main beam is steered in the required direction by choosing the vector $\mathbf{U}_d$ as given by (5) and (6) for that θ_d. Then, to force the sidelobes down in other directions, a large number of closely spaced interfering signals are assumed to be incident on the array from the sidelobe region. The number of interfering signals M is chosen to be two or three times the number of degrees of freedom in the array, so the adaptive array cannot simply place a null on each interfering signal. The powers of these interfering signals are then adjusted iteratively, until the desired sidelobe behavior is obtained.

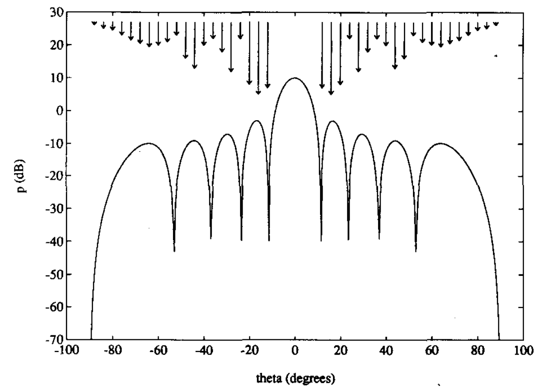


Fig. 4. Multiple interference signals in the sidelobe region.

For example, Fig. 4 shows the pattern of an equally weighted array of ten isotropic elements with half-wavelength spacing and shows how interference signals could be added to control the sidelobes at angles outside the main beam.

Suppose the goal is to have the sidelobe level at angle θ be $D(\theta)$ dB below the beam peak. For the initial step, the powers of all M interference signals are set to zero, so the only undesired terms are the noise signals. The optimal weights are calculated from (10) and the resulting pattern is calculated from (8). (The initial pattern with no interference is called the quiescent pattern.) We refer to this initial step as the $k = 0$ iteration.

Then, for each succeeding iteration, the sidelobe level of the pattern is compared with the desired sidelobe level $D(\theta)$ and the interfering signal powers are adjusted accordingly. Suppose θ_{i_m} denotes the arrival angle for interference signal m, where $m = 1, 2, \cdots, M$. If the sidelobe level of the pattern at angle θ_{i_m} is above the desired sidelobe level $D(\theta_{i_m})$, the interference power at angle θ_{i_m} is increased. If the sidelobe level at θ_{i_m} is below $D(\theta_{i_m})$, the interference power is decreased. Negative interference powers are not allowed, so if an interference power would become negative it is set to zero.

Before assigning interference powers at iteration k, it is necessary to find the current desired voltage envelope $d(\theta_{i_m}, k)$, which is related to the decibel sidelobe level $D(\theta_{i_m})$ and the current beam peak $P(k)$ by

$$d(\theta_{i_m}, k) = \frac{P(k)}{10^{[D(\theta_{i_m})/20]}}. \tag{19}$$

$d(\theta_{i_m}, k)$ must be recalculated for each iteration because the beam peak $P(k)$ is a function of k. (Although interference is never added in the beam region, the magnitude of $P(k)$ changes at every iteration.)

It is also necessary to calculate the limits of the current beam region, $\theta_L(k)$ and $\theta_R(k)$, at each iteration. These angles continually change, because the beam width widens as the sidelobes are reduced. $\theta_L(k)$ and $\theta_R(k)$ are needed so that interference can be kept out of the main beam. The procedure we use is to choose the set of interference arrival angles θ_{i_m} so interference actually arrives from all regions, including the main beam region. But then at each iteration the interference powers are set to zero for each θ_{i_m} inside the region $\theta_L(k) \leq \theta \leq \theta_R(k)$. (In our computer code, we determine $\theta_L(k)$ and $\theta_R(k)$ by estimating the derivative $dp(\theta, k)/d\theta$ with a central difference formula [15]. The zero crossings of the derivative closest to θ_d are used for $\theta_L(k)$ and $\theta_R(k)$.)

After finding $d(\theta_{i_m}, k)$ and the beam region, the interference signal powers can be set for the next iteration. We actually adjust the interference-to-noise ratios rather than the powers. Let $\xi_{i_m}(k)$ denote the INR of interference signal m at iteration k. The INR's for iteration $(k+1)$ are set according to

$$\xi_{i_m}(k+1) = \begin{cases} 0, & \theta_{i_m} \in [\theta_L(k), \theta_R(k)], \\ \max[0, \Gamma_{i_m}(k)], & \text{otherwise} \end{cases} \tag{20}$$

where

$$\Gamma_{i_m}(k) = \xi_{i_m}(k) + K[p(\theta_{i_m}, k) - d(\theta_{i_m}, k)]. \tag{21}$$

K is a scalar constant called the iteration gain.

After the M interference powers are assigned, the covariance matrix Φ_u in (11) can be calculated. With M interfering signals, the signal vector $\mathbf{X}_u$ is given by

$$\mathbf{X}_u = \mathbf{X}_n + \sum_{m=1}^{M} \mathbf{X}_{i_m}, \tag{22}$$

where $\mathbf{X}_n$ is the noise signal vector and $\mathbf{X}_{i_m}$ is the mth interference signal vector as in (13). The matrix Φ_u is found from (11) to be

$$\Phi_u = E\{\mathbf{X}_n^* \mathbf{X}_n^T\} + \sum_{m=1}^{M} E\{\mathbf{X}_{i_m}^* \mathbf{X}_{i_m}^T\}$$
$$= \sigma^2 \left[\mathbf{I} + \sum_{m=1}^{M} \xi_{i_m}(k) \mathbf{U}_{i_m}^* \mathbf{U}_{i_m}^T\right], \tag{23}$$

where $\mathbf{U}_{i_m}$ is found from (5) using the appropriate value of θ_{i_m}. Solving (10) then yields the optimal weight vector for that iteration.

This cycle is repeated iteratively until either the pattern is judged to be satisfactory or $\mathbf{W}$ no longer changes from iteration to iteration.

B. An Example

The following example shows how a typical pattern evolves during this process. Suppose we have a ten-element array of isotropic elements spaced every half-wavelength. Assume we want a weight vector yielding a beam at broadside and uniform sidelobes 30 dB below the peak.

We first set all interference powers to zero and compute the initial (quiescent) pattern. The beam is steered by choosing $\theta = \theta_d = 0°$ in (5) and (6). With only noise in $\mathbf{X}_u$, the covariance matrix in (23) is $\sigma^2 \mathbf{I}$ and the initial weight vector is $\mu/\sigma^2 \mathbf{U}_d^*$. The initial pattern computed from this weight vector is shown in Fig. 5(a). It is a standard $[\sin(N\pi/2 \sin\theta)]/[\sin(\pi/2 \sin\theta)]$ pattern with 13 dB first sidelobes (where N is the number of elements).

Now we iterate on the interference power. Fig. 5(b) shows the interference spectrum at the first iteration as assigned by (20). In this example we use $M = 119$ interference signals and an iteration gain of $K = 0.5$. Note how the initial interference spectrum mirrors the sidelobes in Fig. 5(a).

The first adapted voltage pattern $p(\theta, 1)$ (in response to the interference spectrum of Fig. 5(b)) is shown in Fig. 5(c). Also shown in Fig. 5(c) (as a dashed line) is the new desired voltage envelope, $d(\theta, 1)$. Note that $d(\theta, 1) < d(\theta, 0)$ because $P(1) < P(0)$ in (19) due to the adaptation. Fig. 5(d) shows the $k = 2$ interference spectrum, which is weaker than the $k = 1$ spectrum. Figs. 6(a), 6(b) and 6(c), 6(d) show how the algorithm progresses through iterations $k = 3, 4$ and $k = 9, 10$. Each figure shows the most recent adapted voltage pattern, the desired voltage envelope and the new interference spectrum. Note the evolution of the sidelobes as they approach the desired envelope. Note also the behavior of the beam and the interference spectrum. Eventually, (as long as K is suitably chosen) these figures approach a steady-state where the pattern, spectrum and desired envelope change little from one iteration to the next. The process is stopped when no further changes occur or when the sidelobe behavior is deemed acceptable. As may be seen in the final pattern in Fig. 7, the algorithm performed well for this simple problem.

We will examine the behavior of the algorithm with more interesting problems below. First, however, we mention the issue of stability.

C. Stability

If K is set too large, the algorithm becomes unstable, as one would expect from discrete control theory [16]. The maximum value of K that yields stable operation depends strongly on the desired sidelobe depth $D(\theta)$. Table I shows some examples of the maximum stable K for the uniform sidelobe problem above and for various sidelobe levels. These values were obtained simply by trial-and-error. Note that the lower the sidelobes, the higher K can be. We shall not go into an analysis of the maximum value of K for stable operation here, but simply point out that it is easy to find suitable values of K in practice by trial-and-error. To make the algorithm converge as quickly as possible, one wants the largest value of K for which the algorithm is stable. For a given array, a few trials quickly yield a suitable value for K.

IV. Applications and Examples

Now we show some examples illustrating the usefulness of this algorithm. We begin with an array of uniformly spaced isotropic elements and compare our solution with the classical Dolph–Chebyshev solution. Then we consider arrays with various types of elements and element spacings. Finally, we consider examples with nonuniform sidelobe specifications.

A. The Uniform Sidelobe Problem

The uniform sidelobe problem is a classical problem in array design. The standard method of achieving uniform

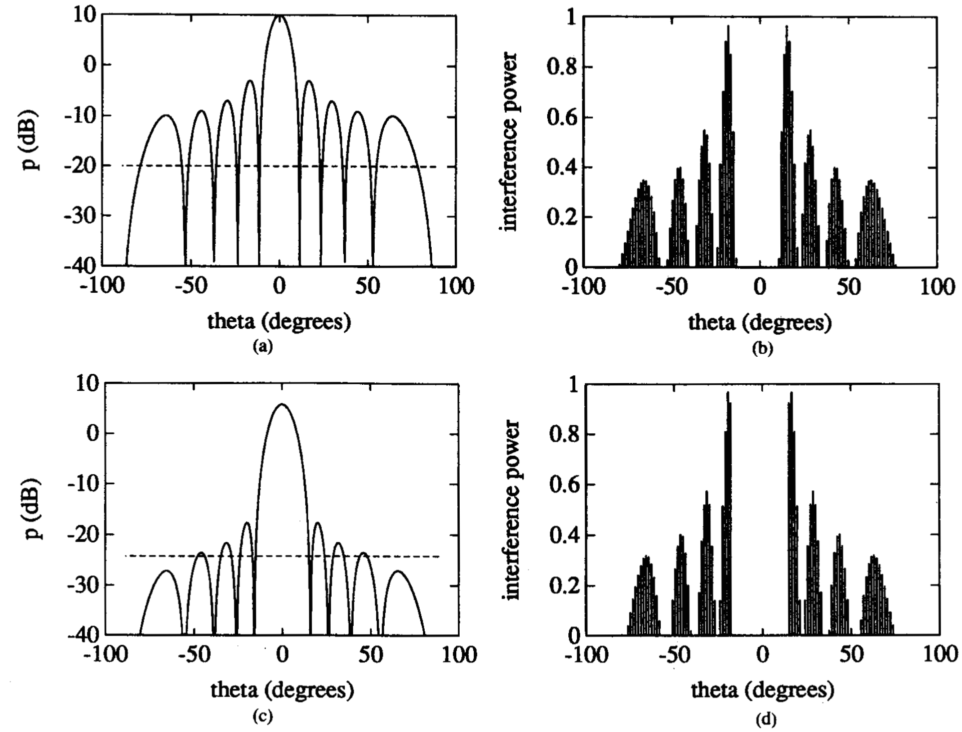


Fig. 5. Patterns and interference powers. (Dotted line is $d(\theta, k)$.) (a) $p(\theta, 0)$ and $d(0)$. (b) Interference spectrum: $k = 1$. (c) $p(\theta, 0)$ and $d(1)$. (d) Interference spectrum: $k = 2$.

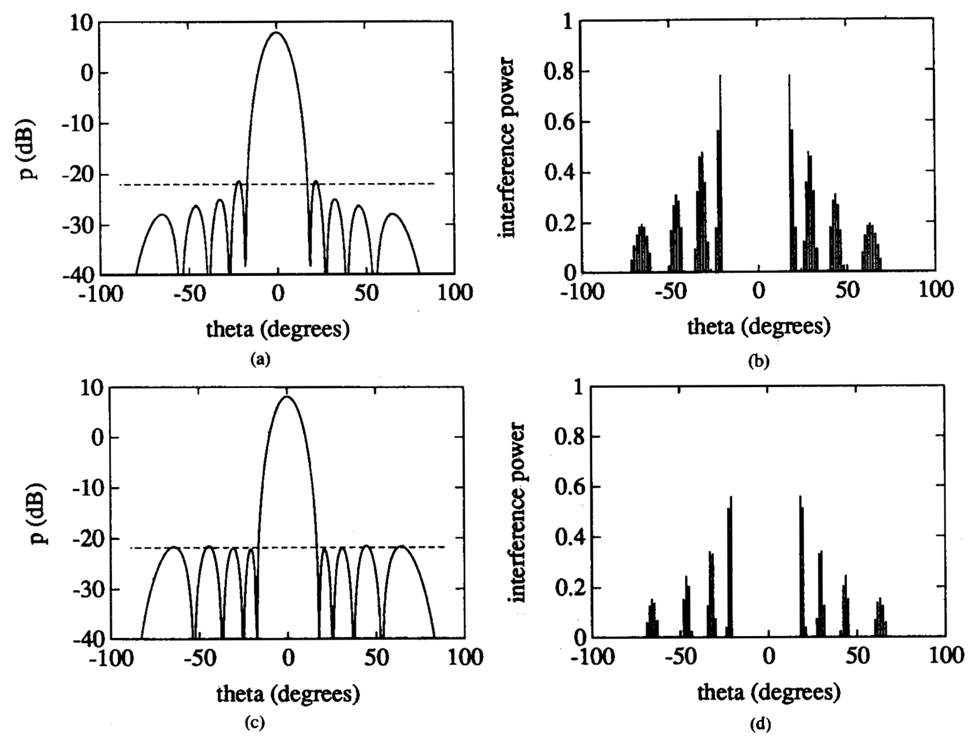


Fig. 6. Patterns and interference powers. (Dotted line is $d(\theta, k)$.) (a) $p(\theta, 3)$ and $d(3)$. (b) Interference spectrum: $k = 4$. (c) $p(\theta, 9)$ and $d(9)$. (d) Interference spectrum: $k = 10$.

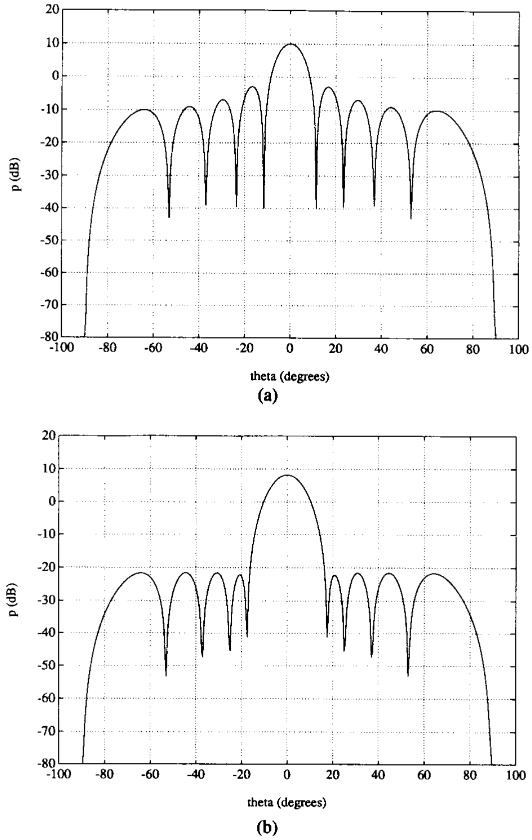


Fig. 7. Initial and final patterns: ten-element array, isotropic elements. (a) Initial pattern. (b) Final pattern.

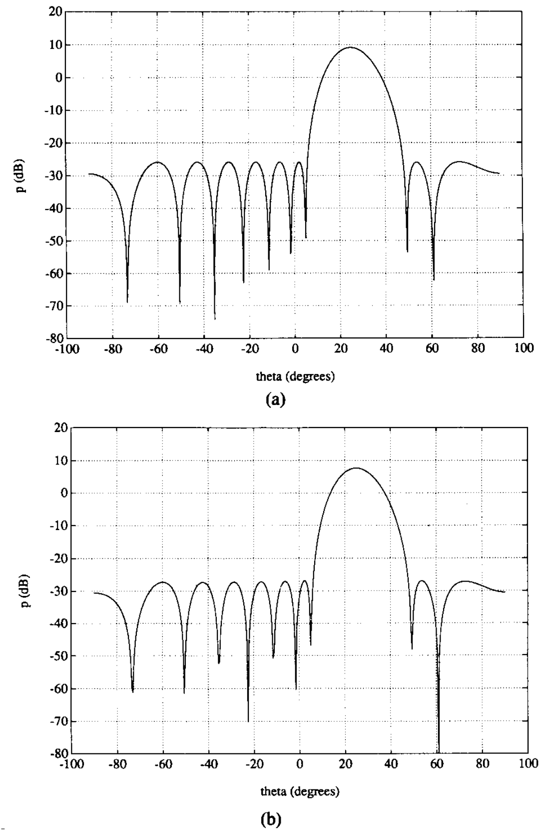


Fig. 8. Patterns: ten isotropic elements, half-wavelength spacing. (a) Dolph-Chebyshev pattern. (b) Pattern obtained with adaptive algorithm.

TABLE I
STABLE LIMITS ON K FOR VARIOUS TARGETS OF UNIFORM SIDELOBE DEPTH

Desired Sidelobe Depth (dB)	Maximum K
30	2.1
35	31.
40	198.

sidelobes with an antenna array is the Dolph-Chebyshev (DC) approach [8]. This method is well known and easy to use. Several fast algorithms for computing the DC weights are available [17]-[20].

Fig. 8 compares the patterns obtained with the DC method and our method for a uniformly spaced array of isotropic elements. These patterns are for a ten-element array with half-wavelength spacing between elements and with the main beam steered to $\theta_d = 25°$. The design objective is to obtain sidelobes 35 dB down from the main beam. Fig. 8(a) shows the pattern obtained with the DC method and Fig. 8(b) shows the pattern from our algorithm after 27 iterations with $K = 25$.

As may be seen, the patterns in Figs. 8(a) and 8(b) are basically identical. In Table II we compare the weight vectors obtained with the two methods. The weight vectors are shown in polar form, with the magnitudes of the fifth and sixth weights normalized to unity. It is clear that the weights obtained with this algorithm are converging to the Dolph-Chebyshev weights when applied to the same problem.

TABLE II
A COMPARISON OF THE DC METHOD AND SC METHOD WEIGHT VECTORS FOR THE UNIFORM SIDELOBE PROBLEM

n	w_{DC_n}	w_{SC_n}
1	$0.176e^{j(0.0000)}$	$0.179e^{j(0.0123)}$
2	$0.367e^{j(-1.3277)}$	$0.371e^{j(-1.3155)}$
3	$0.622e^{j(-2.6554)}$	$0.625e^{j(-2.6470)}$
4	$0.858e^{j(2.3001)}$	$0.859e^{j(2.3053)}$
5	$1.000e^{j(0.9724)}$	$1.000e^{j(0.9741)}$
6	$1.000e^{j(-0.3553)}$	$1.000e^{j(-0.3569)}$
7	$0.858e^{j(-1.6830)}$	$0.859e^{j(-1.6882)}$
8	$0.622e^{j(-3.0107)}$	$0.625e^{j(-3.0191)}$
9	$0.367e^{j(1.9448)}$	$0.371e^{j(1.9326)}$
10	$0.176e^{j(0.6171)}$	$0.179e^{j(0.6048)}$

B. Element Patterns and Spacings

The DC method is strictly applicable only to arrays with uniform spacing and isotropic elements. For arrays with

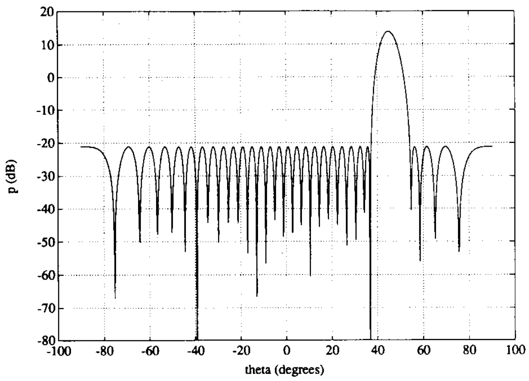


Fig. 9. Dolph-Chebyshev pattern of a 30-element array: isotropic elements, half-wavelength spacing, $\theta_d = 45°$.

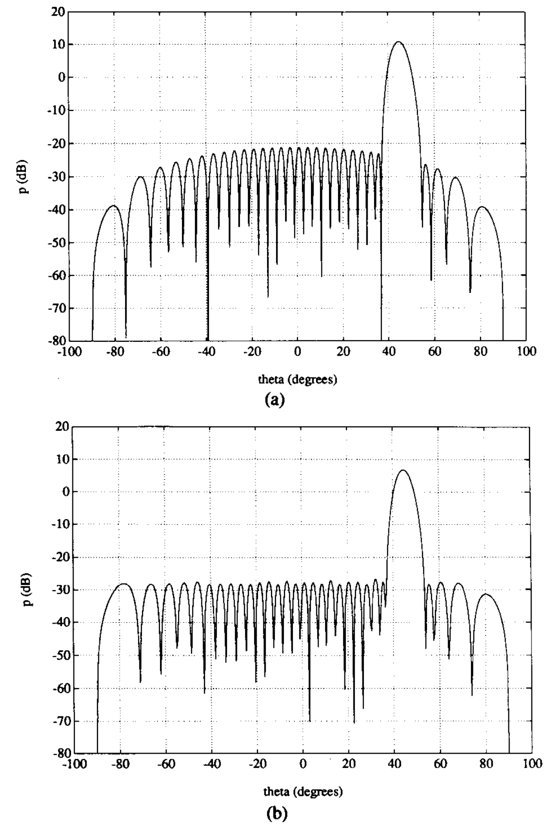


Fig. 10. Patterns for 30 short dipoles, $\theta_d = 45°$. (a) Dolph-Chebyshev weights. (b) Weights obtained from the algorithm.

nonisotropic elements, the DC method can be used only by ignoring the element patterns. Low sidelobes can be achieved in this way, but the sidelobes are no longer uniform, because of the element patterns.

For example, suppose we have an array of 30 short dipoles in which the dipoles are aligned with the array axis and spaced every half-wavelength. Assume the pattern of every element is $f_j(\theta) = \cos\theta$. Suppose we want the beam steered to $\theta_d = 45°$ with uniform sidelobes 35 dB down. To use the DC method, we first calculate the DC weights for a 30-element array. Fig. 9 shows the pattern that would result if these

TABLE III
PARAMETERS FOR THE 33-ELEMENT LINEAR ARRAY; l_j AND d_j ARE IN WAVELENGTHS, τ_j IS IN DEGREES

j	l_j	τ_j	d_j	j	l_j	τ_j	d_j
1	0.25	0.0	0.5	18	0.25	0.0	0.50
2	0.25	0.5	0.5	19	0.24	5.0	0.45
3	0.24	5.0	0.55	20	0.26	4.7	0.55
4	0.20	-32.	0.54	21	0.27	-8.9	0.54
5	0.26	-3.2	0.60	22	0.28	3.0	0.53
6	0.27	10.	0.45	23	0.25	3.2	0.56
7	0.23	1.0	0.46	24	0.25	2.8	0.54
8	0.24	0.0	0.50	25	0.25	2.9	0.56
9	0.25	0.0	0.50	26	0.23	1.5	0.50
10	0.21	7.0	0.51	27	0.27	0.7	0.50
11	0.28	6.0	0.47	28	0.28	0.33	0.50
12	0.30	4.4	0.48	29	0.24	0.0	0.57
13	0.29	0.0	0.61	30	0.24	0.43	0.51
14	0.19	1.0	0.57	31	0.25	-20.	0.52
15	0.22	-2.1	0.65	32	0.26	0.8	0.49
16	0.22	3.0	0.42	33	0.25	-9.6	–
17	0.25	0.0	0.50				

weights were used with isotropic elements. When the DC weights are used with the dipole array, the resulting pattern is shown in Fig. 10(a). Note that the sidelobes in Fig. 10(a) are no longer uniform but are tapered by the element pattern. Also note that the sidelobe level at $\theta = 0°$ is not 35 dB below the beam peak as desired.

On the other hand, when the adaptive array algorithm is used to solve this problem, the element patterns are included in the procedure from the beginning. Fig. 10(b) shows the final pattern obtained with this algorithm. (With the element patterns included, the final weights obtained are different from the DC weights, of course.) Note how much closer the sidelobes come to meeting the original uniform design objective when the adaptive array algorithm is used.

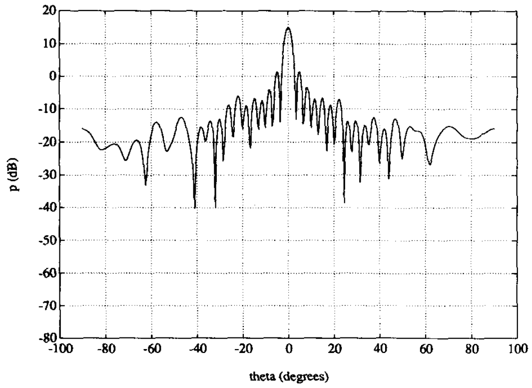
One advantage of this algorithm is thus seen: the element patterns are included directly in the algorithm. There are no restrictions whatsoever on the element patterns. Not only may they be nonisotropic, they may differ from element to element. We show an example of this type below.

The algorithm above also places no restrictions on element spacing. The problem of designing an array with arbitrary element spacing for low sidelobes has no general solution in the literature. The algorithm described here can be used, however.

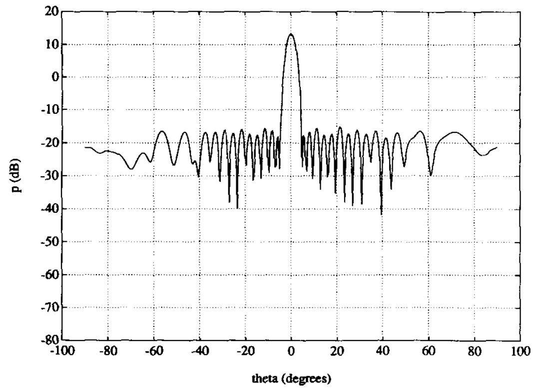
Consider a 33-element linear array of dipoles with nonuniform spacing and orientation. Suppose the element pattern for dipole j is given by

$$f_j(\theta) = \frac{\cos\left[\pi l_j \sin\left(\theta + \tau_j\right)\right] + \cos\left(\pi l_j\right)}{\cos\left(\theta + \tau_j\right)}, \quad (24)$$

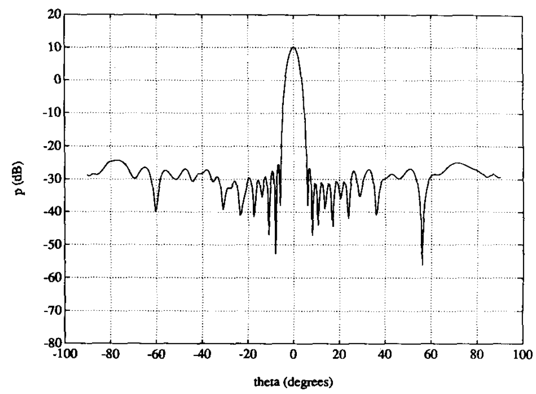
where l_j is the dipole length in wavelengths, τ_j is the dipole



(a)



(b)



(c)

Fig. 11. Patterns for the array of Table III. (a) Initial pattern. (b) Final pattern: 30 dB sidelobe design. (c) Final pattern: 40 dB sidelobe design.

tilt angle from the array axis, and θ is defined as in Fig. 1. Let the dipoles in the array vary in length, tilt angle and spacing. For the example below, we used the (arbitrary) set of array parameters shown in Table III. First, suppose the goal is to achieve 30 dB uniform sidelobes with the beam steered to $\theta_d = 0°$. Fig. 11a shows the initial pattern and Fig. 11(b) shows the final pattern after 5 iterations with $K = 1.0$ for iterations 1 and 2 and $K = 0.5$ for iterations 3–5. Note the improvement in the sidelobes. Second, suppose the goal is to achieve 40 dB uniform sidelobes with this array. In this case the final pattern is shown in Fig. 11(c). Note that for this case the sidelobes do not meet the 40 dB specification. It turned out that these were the best sidelobes that could be achieved with this array.

This example illustrates the fact that it is not necessarily possible to meet an arbitrary pattern specification with a given set of elements. When the design objective is unattainable with the given array, one finds that the algorithm settles

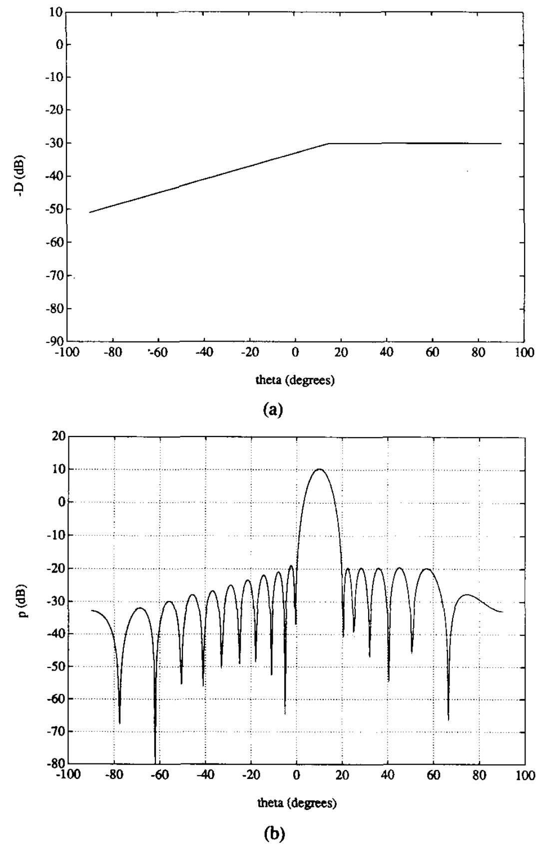


Fig. 12. A two-segment nonuniform sidelobe problem. (a) Desired envelope function. (b) Shaped pattern.

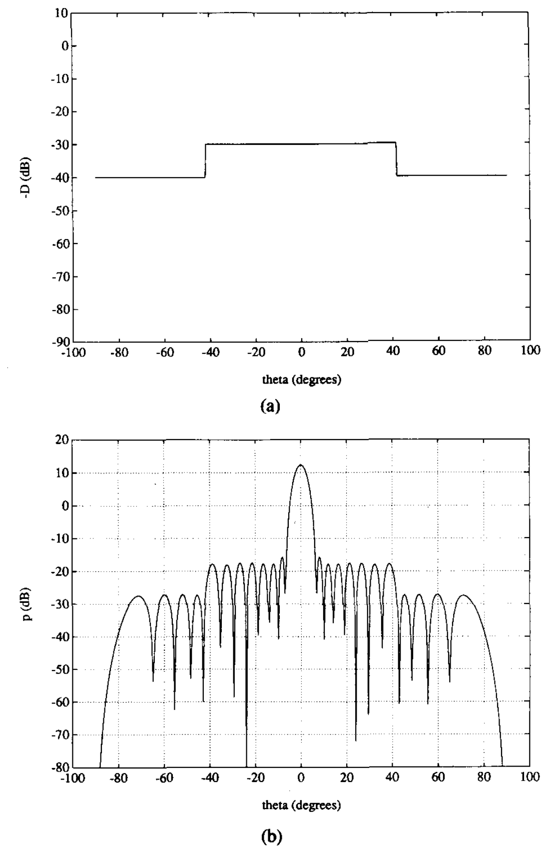


Fig. 13. A three-segment nonuniform sidelobe problem. (a) Desired envelope function. (b) Shaped pattern.

into a steady-state pattern that, while not meeting the objectives, is essentially the best that can be achieved. When the objective is attainable, the algorithm finds a suitable solution quickly.

C. Arbitrary Sidelobe Shaping

The algorithm can also be used to produce patterns in which the desired sidelobe level $D(\theta)$ varies as a function of θ.

For example, suppose we want to obtain a pattern whose sidelobes meet the design envelope shown in Fig. 12(a). The segment of $D(\theta)$ to the left of the beam slopes at 0.20 dB/degree, meeting the $\theta_d = 10°$ axis at 30 dB below the beam peak. The segment to the right of the beam is flat at 30 dB below the peak. If the algorithm is applied to an array of 17 half-wavelength spaced, isotropic elements, the final pattern is shown in Fig. 12(b). 25 iterations of the algorithm were performed. The iteration gain K was set to $K = 2.0$ for iterations 1–9 and reduced to $K = 1.2$ for iterations 10–25. The final pattern is clearly suitable.

Next, suppose the design objective is to achieve the sidelobe behavior shown in Fig. 13(a). This $D(\theta)$ has three flat segments: 40 dB below the peak for $-90° < \theta \leq -42°$, 30 dB below the peak for $-42° < \theta \leq 42°$, and 40 dB below the peak for $42° < \theta < 90°$. The beam is at $\theta_d = 0°$. For this example, we use an array of 24 center-excited quarter-wavelength dipoles. The dipoles have their axes aligned with the array axis and have a uniform half wavelength center-to-center spacing. Fig. 13(b) shows the result of using the algorithm on this array with the $d(\theta)$ in Fig. 13(a). The iteration gain was $K = 1.5$ for iterations 1–14, $K = 1.0$ for iterations 15–22, and $K = 0.7$ for iterations 23–29.

V. Conclusion

An algorithm has been described for choosing weights for a given set of array elements to meet a specified sidelobe criterion. The algorithm uses the array elements as elements of an adaptive array. The pattern is steered to the desired angle and interference is injected into the array at angles where low sidelobes are desired. The interference powers are then adjusted iteratively until the desired sidelobe behavior is achieved or until the best attainable pattern has been found.

The algorithm can be used on a large class of design problems. Unlike the classical Dolph–Chebyshev method, the algorithm can be used on arrays with nonuniformly spaced elements and with nonisotropic and unequal element patterns. It can also handle problems where the desired sidelobe level varies with angle.

References

[1] T. T. Taylor, "Design of line source antennas for narrow beamwidth

and low sidelobes," *IRE Trans. Antennas Propagat.*, vol. AP-3, pp. 16-28, 1955.

[2] R. T. Hyneman and R. M. Johnson, "A technique for the synthesis of shaped-beam radiation patterns with approximately equal-percentage ripple," *IEEE Trans. Antennas Propagat.*, vol. AP-15, pp. 736-743, 1967.

[3] R. F. Hynemen, "A technique for the synthesis of line-source antenna patterns having specified sidelobe behavior," *IEEE Trans. Antennas Propagat.*, vol. AP-16, pp. 430-435, 1968.

[4] R. S. Elliott, "Design of line source antennas for narrow beamwidth and asymmetric low sidelobes," *IEEE Trans. Antennas Propagat.*, vol. AP-23, pp. 100-107, 1975.

[5] ——, "Design of line source antennas for sum patterns with sidelobes of individually arbitrary heights," *IEEE Trans. Antennas Propagat.*, vol. AP-24, pp. 76-83, 1976.

[6] ——, "Design of line source antennas for difference patterns with sidelobes of individually arbitrary heights," *IEEE Trans. Antennas Propagat.*, vol. AP-24, pp. 310-316, 1976.

[7] W. D. White, "A flexible synthesis procedure for line source antennas," *IEEE Trans. Antennas Propagat.*, vol. AP-24, pp. 857-859, 1976.

[8] C. L. Dolph, "A current distribution for broadside arrays which optimizes the relationship between beamwidth and sidelobe level," *Proc. IRE*, vol. 34, p. 335-348, 1946.

[9] A. T. Villeneuve, "Taylor patterns for discrete arrays," *IEEE Trans. Antennas Propagat.*, vol. AP-32, pp. 1089-1093, 1984.

[10] R. S. Elliott and G. J. Stern, "A new technique for shaped beam synthesis of equispaced arrays," *IEEE Trans. Antennas Propagat.*, vol. AP-32, pp. 1129-1133, 1984.

[11] S. P. Applebaum, "Adaptive arrays," *IEEE Trans. Antennas Propagat.*, vol. AP-24, pp. 585-598, 1976.

[12] R. T. Compton Jr., *Adaptive Arrays—Concepts and Performance.* Englewood Cliffs, NJ: Prentice-Hall, 1988.

[13] J. C. Sureau and K. J. Keeping, "Sidelobe control in cylindrical arrays," *IEEE Trans. Antennas Propagat.*, vol. AP-30, pp. 1027-1031, 1982.

[14] E. C. Dufort, "Pattern synthesis based on adaptive array theory," *IEEE Trans. Antennas Propagat.*, vol. 37, pp. 1011-1018, 1989.

[15] C. F. Gerald and P. O. Wheatley, *Applied Numerical Analysis.* Menlo Park, CA: Addison-Wesley, 1984.

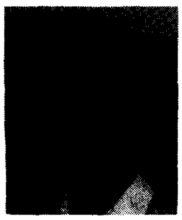
[16] K. Ogata, *Discrete-Time Control Systems.* Englewood Cliffs, NJ: Prentice-Hall, 1987.

[17] D. Barbiere, "A method for calculating the current distribution of Tschebyscheff arrays," *Proc. IRE*, vol. 40, pp. 78-82, Jan. 1952.

[18] C. J. van der Maas, "A simplified calculation for Dolph-Tschebyscheff arrays," *J. Appl. Phys.*, vol. 25, no. 1, pp. 121-124, 1954.

[19] A. N. Nuttal, "Generation of Dolph-Chebyshev weights via a fast Fourier transform," *Proc. IEEE*, vol. 62, p. 1396, 1974.

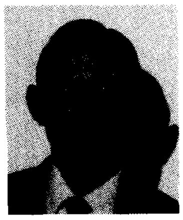
[20] A. D. Bresler, "A new algorithm for calculating the current distributions of Dolph-Chebyshev arrays," *IEEE Trans. Antennas Propagat.*, vol. AP-28, pp. 951-952, 1980.



Carl A. Olen (S'85-M'87-S'88-M'89) was born in East Cleveland, OH, on August 25, 1958. He received the B.E.E. degree from Cleveland State University, Cleveland, OH, in 1987, and the M.S. degree in electrical engineering from The Ohio State University, Columbus, in 1989.

From 1986 to 1987, he was a Research Assistant at the NASA Lewis Research Center Electro-Physics Office, Cleveland, OH, where he worked with non-propulsive applications of ion thrusters and low earth orbit environment simulation for materials testing. During 1987 he also helped build a solar cell testing facility for Cleveland State University. In 1988, as an Ohio State University Fellow, he studied communications, electromagnetics, and adaptive arrays. He was also an Ohio Board of Regents Fellow from 1987 to 1989. Currently a Research Associate with The Ohio State University ElectroScience Laboratory, he is working with international satellite communications systems.

Mr. Olen is a member of Tau Beta Pi and Eta Kappa Nu. He received the Cleveland State University President's Award for outstanding graduate of 1987.



R. T. Compton, Jr. (S'58-M'59-M'81-SM'82-F'84) was born in St. Louis, MO, on July 26, 1935. He received the S.B. degree from the Massachusetts Institute of Technology, Cambridge, in 1958, and the M.Sc. and Ph.D. degrees from The Ohio State University, Columbus, in 1961 and 1964, respectively, all in electrical engineering.

He is a Professor of Electrical Engineering at The Ohio State University. From 1965 to 1967, he was an Assistant Professor of Engineering at Case Institute of Technology, Cleveland, OH, and from 1967 to 1968 he was an NSF Postdoctoral Fellow at the Technische Hochschule, Munich, Germany. He spent the 1983-1984 academic year at the Naval Research Laboratory, Washington, DC.

Dr. Compton is a member of Sigma Xi and Pi Mu Epsilon. He received the M. Barry Carlton Award for best paper from the IEEE Aerospace and Electronic Systems Society in 1983.