PA 48, 119

UNIVERSITY OF CALIFORNIA, SAN DIEGO
MARINE PHYSICAL LABORATORY
SCRIPPS INSTITUTION OF OCEANOGRAPHY
SAN DIEGO, CALIFORNIA 92152

EFFICIENT COMPUTATION OF ARRAY PATTERNS

Victor C. Anderson

Sponsored by
Office of Naval Research
N00014-75-C-0749
and
Advanced Research Projects Agency
N00014-76-C-1080

MPL-U-23/76

Reprinted from the JOURNAL OF THE ACOUSTICAL SOCIETY OF AMERICA, Vol. 61, No. 3, pp. 744-755, March 1977.

Rec'd 7/5/77

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER MPL-U-23/76	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* EFFICIENT COMPUTATION OF ARRAY PATTERNS		5. TYPE OF REPORT & PERIOD COVERED Summary 6. PERFORMING ORG. REPORT NUMBER MPL-U-23/76
7. AUTHOR(s) Victor C. Anderson		8. CONTRACT OR GRANT NUMBER(s) ONR N00014-75-C-0749 and ARPA N00014-76-C-1080
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of California, San Diego, Marine Physical Laboratory of the Scripps Institution of Oceanography, San Diego, California 92152		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research, Code 222, Department of the Navy, Arlington, Virginia 22217		12. REPORT DATE March 1977 13. NUMBER OF PAGES 11
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Document cleared for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Array patterns, [Sonar] computation, quantized stored cosine function, digital Fourier transform algorithms, trigonometric interpolation, hypothetical array.

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

The impact of a symmetrical array geometry, the use of a quantized stored cosine function, the exploitation of digital Fourier transform algorithms, and the application of trigonometric interpolation in the computation of array patterns is discussed. Careful selection of parameters permits sampling the array pattern only 6% above the theoretical Nyquist limit. Reconstruction of array patterns showing -20, -30, and -40-dB relative interpolation errors are →

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

presented. A saving of 8000:1 in computation time over direct "brute-force" array-pattern computation is illustrated for a hypothetical array.

Efficient computation of array patterns

V. C. Anderson

University of California, San Diego, Marine Physical Laboratory of the Scripps Institution of Oceanography, San Diego, California 92132
(Received 30 August 1976; revised 22 October 1976)

The impact of a symmetrical array geometry, the use of a quantized stored cosine function, the exploitation of digital Fourier transform algorithms, and the application of trigonometric interpolation in the computation of array patterns is discussed. Careful selection of parameters permits sampling the array pattern only 6% above the theoretical Nyquist limit. Reconstruction of array patterns showing −20, −30, and −40-dB relative interpolation errors are presented. A saving of 8000:1 in computation time over direct "brute-force" array-pattern computation is illustrated for a hypothetical array.

PACS numbers: 43.60.Qv, 43.60.Cg, 43.30.Vh

INTRODUCTION

In today's world of readily available high-speed digital-computer facilities, the computation of the directional response for arrays composed of many arbitrarily distributed elements is commonplace. However, when the number of elements is very large and the aperture extends over many wavelengths in two or three dimensions, the cost of such computation can be significant. When the degree of array complexity is reached where computing cost becomes a concern, a careful selection of algorithms and sampling intervals for numerical processing can lead to substantial savings in computer time.

The rationale for the selection of sampling intervals begins with the basic transform relationship embodied in the farfield directional-response function of a coherently excited aperture;

$$\psi(\vec{k}) = \int_{-\infty}^{\infty} A(\vec{\rho}) \exp[i2\pi \vec{k} \cdot \vec{\rho}]\, d\vec{\rho} \ . \tag{1}$$

When the aperture is made up of N discrete phased array elements, $A(\vec{\rho})$ consists of a set of delta functions with complex amplitude coefficients. For this case, which is the one of interest here,

$$\psi(\vec{k}) = \sum_{j=1}^{N} A_j \exp[i2\pi \vec{k} \cdot \vec{\rho}_j] \ . \tag{2}$$

Furthermore, if the array is phased for plane waves, the complex phase of the A_j coefficients can be incorporated into $\vec{k}$ so that, in Cartesian coordinates, the array factor[1] is defined as

$$\psi(\vec{k}) = \sum_{j=1}^{N} A_j \exp[i2\pi(k_x x_j + k_y y_j + k_z z_j)] \ , \tag{3}$$

where $k_x = (\alpha - \alpha_0)/\lambda$; $k_y = (\beta - \beta_0)/\lambda$; $k_z = (\gamma - \gamma_0)/\lambda$; λ is the wavelength of the plane wave; α, β, γ are the direction cosines of the normal to the wavefront; and α_0, β_0, γ_0 are the direction cosines of the steering vector. Note that, in accord with conventional antenna terminology, a distinction is made between the generally complex array factor $\psi(\vec{k})$ and its power, the array pattern $|\psi(\vec{k})|^2$.

The array factor provides a convenient way of describing the performance of a distributed set of transmitting or receiving elements. However, it, of course, does not include the directional characteristics of the elements themselves. On the one hand, if all of the elements are identical and oriented with their response axes parallel, the overall response factor at a particular frequency is a simple product of the array factor and the directional factor of the individual elements. On the other hand, if they are not identical or not oriented parallel to each other, the element weights are not independent of angle. In this case the computation becomes much more laborious in that the response is no longer a function of the universal parameters k_x, k_y, k_z but must be considered a function of frequency, response direction, and steering direction.

The array factor $\psi(\vec{k})$ is a continuous function of $\vec{k}$ but, in practice, it is always approximated by a discrete set of M samples over the "visible" range of $\vec{k}$ that is of interest. (The visible range is that range of values corresponding to sums of the direction cosines for the response and steering vectors lying between $+2$ and -2 at the frequency of interest.[2]) Thus, the straightforward computation of Eq. (3) requires $M \times N$ operations of the form

$$\begin{aligned} &\mathrm{ARG} = \mathrm{KX(J)} * \mathrm{X(I)} + \mathrm{KY(J)} * \mathrm{Y(I} + \mathrm{KZ(J)} * \mathrm{Z(I)} \ , \\ &\mathrm{CSUM(J)} = \mathrm{CSUM(J)} + \mathrm{A(I)} * \mathrm{COS(ARG)} \ , \\ &\mathrm{SSUM(J)} = \mathrm{SSUM(J)} + \mathrm{A(I)} * \mathrm{SIN(ARG)} \ . \end{aligned} \tag{4}$$

I. EFFECT OF SYMMETRY

For the sake of completeness, what is perhaps the most common technique for reducing the computation effort will be mentioned—that of imposing a condition of symmetry on the array. Consider an array distribution which is symmetrical about all three axes. In this case there will be eight elements described by one set of coordinate values $(\pm|x|, \pm|y|, \pm|z|)$. Suitably grouping the terms, the eight-element sum becomes

$$\begin{aligned} &\sum_{8} \exp(\pm ik_x|x| \pm ik_y|y| \pm ik_z|z|) \\ &\quad = 8\cos(k_x|x|)\cos(k_y|y|)\cos(k_z|z|) \ . \end{aligned} \tag{5}$$

The eight complex terms of the sum thus combine into a single symmetrical term such that the imaginary part of the sum equals zero and need not be computed. Furthermore, because ψ is now symmetrical, i.e.,

$$\psi(k_x,\ k_y,\ k_z)=\psi(|k_x|,\ |k_y|,\ |k_z|)\ ,$$

just its real part, and that for only positive values of k_x, k_y, k_z need be computed. If the array is not symmetric, only the redundancy $\psi(\vec{k})=\psi^*(-\vec{k})$ holds so that the complex $\psi(\vec{k})$ must be computed over one-half of the appropriate $\vec{k}$ space. Thus, for the symmetrical array, the $2\times M/2\times N$ operations for a three-dimensional array and a three-coordinate beam set of the form of Eq. (4) have been replaced by $(M/8)\times(N/8)$ operations of the form

$$\mathrm{SUM(J)=SUM(J)+A(I)*COS[KX(J)*X(I)]}$$
$$\mathrm{*COS[KY(J)*Y(I)]*COS[KZ(J)*Z(I)]}\ . \qquad (6)$$

Recognizing that the COS-function routine dominates the computational time, the overall reduction in time for a three-dimensional array is about $\frac{1}{20}$. For two- and one-dimensional arrays the saving is not quite as great, about $\frac{1}{8}$ and $\frac{1}{4}$, respectively. Without question, this order-of-magnitude savings in time is a strong argument for symmetry in array design as far as computation of array patterns is concerned.

II. USE OF A STORED COSINE TABLE

Another very effective way of reducing the computation time is to replace the COS-function routine with a look-up table. This inevitably will introduce an error in the result because of the quantization introduced. The statistical effect of the quantization error can be estimated by using the expression for the effect of random phase errors in an array, [3]

$$\psi^2(k)=(1-\sigma^2)\psi_0^2(k)(\sigma_A^2+\sigma_\phi^2)\sum_m A_m^2\ , \qquad (7)$$

where $\psi_0^2(k)$ is the array pattern without errors, σ_ϕ^2 is the variance of the phase errors, σ_A^2 is the variance of the amplitude errors, and the A_m are the amplitude coefficients of the array elements. An insight into the magnitude of the effect of phase errors can be gained by considering an array with elements having unit weights. The difference between the mean pattern and the pattern unperturbed by errors will be given by

$$\psi^2(k)-\psi_0^2(k)=\sigma_\phi^2[N-\psi_0^2(k)]+\sigma_A^2 N\ . \qquad (8)$$

On axis ($k=0$), $\psi_0^2(k)=N^2$ so that σ_ϕ^2 represents the fractional loss in power in the main lobe.

If the elements of the array were randomly distributed, the mean side-lobe level $[\psi_0^2(k)]$ would be equal to N. The average effect of phase errors which further randomize the phase of the element summations is essentially zero for this case. In regions of side lobes where $\psi_0^2(k)$ is larger than N, the effect of phase errors will be to reduce the side-lobe level; where $\psi_0^2(k)$ is less than N, phase errors will increase the side-lobe level. A useful measure of the quantization error is obtained by placing $\psi_0(k)=0$, then the residual mean error pattern is $\sigma^2 N$ and σ^2 represents the fractional quantization noise power in the side-lobe region as referenced to the mean side-lobe power level. If $\Delta\phi$ is the step size in the argument quantization, the error can be treated as being uniformly distributed over the interval $(-\frac{1}{2}\Delta\theta - +\frac{1}{2}\Delta\theta)$. For this distribution, $\sigma_\theta^2=\frac{1}{12}\Delta\theta^2$.

Thus, for a quantization noise that is 20 dB below the mean side-lobe level, $\sigma_\theta^2=0.01$, $\Delta\theta\simeq 0.35$ rad. For a level 30 dB below the mean side-lobe level, $\Delta\theta\simeq 0.1$ rad. It is apparent that, for most purposes, a relatively crude quantization of the look-up table may be used—intervals between 0.1 and 0.3 rad. Depending upon the architecture of the specific computer involved, it may be more advantageous to use either a short look-up table that covers the range of 2π in 2^n intervals and use the lowest n bits of the argument for the address, or else use an extended table covering the entire range of arguments, i.e., $2\pi D/\lambda$, where D is the maximum dimension of the array aperture. In either case, the savings in time will result from replacing the operations: three cosine routines plus six multiplications plus one addition, by the operations: six multiplications plus one addition. If the cosine routine takes typically ten times as long as the multiplication routine, a reduction in processing time of about $\frac{1}{5}$ will be achieved by the use of a look-up table.

III. UNIFORM SAMPLING OF THE ARRAY PATTERN

From Eq. (1) it is clear that $\psi(\vec{k})$ and $A(\vec{\rho})$ are three-dimensional Fourier-transform pairs. In order to treat the sampling of the variables separately, the volume integration of Eq. (1) can be rewritten

$$\psi(k_x,\ k_y,\ k_z)=\int_{-\infty}^{\infty} a_x\exp(i2\pi k_x x)\,dx\ , \qquad (9)$$

where

$$a_x=a_x(x,\ k_y,\ k_z)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} A(x,\ y,\ z)$$
$$\times\exp[i2\pi(k_y y+k_z z)]\,dy\,dz$$

and a_y and a_z can be similarly defined. The resulting one-dimensional transform pairs are of the form

$$\psi(k)=\int_{-\infty}^{\infty} a_x\exp(i2\pi k_x x)\,dx\ ,$$
$$a_x=\int_{-\infty}^{\infty}\psi(k)\exp(-i2\pi k_x x)\,dx\ . \qquad (10)$$

It is apparent that the physical-aperture function a_x is the Fourier spectrum of the array factor $\psi(k)$. If $\psi(k)$ is to be sampled at uniform intervals of Δk, the resulting sampled function can be expressed as

$$\psi_s(k)=\psi(k)\sum_{n=-\infty}^{n=\infty}\delta(k-n\,\Delta k)\ . \qquad (11)$$

Here the sampled function is obtained by multiplying the continuous function by a Dirac comb. [4] The properties of the Dirac comb are such that the spectrum of the sampled function will be given by

$$a_{sx}(x)=\sum_{m=-\infty}^{m=+\infty} a_x\left(x-\frac{m}{\Delta k}\right)\ . \qquad (12)$$

The result is the familiar aliasing or replicating of the original spectrum at frequency intervals in x of $(\Delta k)^{-1}$. Because the physical aperture is clearly restricted to lie within its maximum dimensions, the maximum extent of the spectrum of $\psi(k)$ is $(-\frac{1}{2}D\leq x\leq\frac{1}{2}D)$. Thus, if $\Delta k\leq 1/D$ (Nyquist period) the aliased spectra will not overlap.

It is customary to reconstruct $\psi(k)$ from $\psi_s(k)$ by selecting the baseband ($m=0$) portion of the aliased spectrum by multiplying by a filter window $G(x)$. Then the reconstructed $\psi(k)$,

$$\psi_R(k)=\int_{-\infty}^{\infty} G(x)\, a_{sx}(x) \exp(-i2\pi kx)\, dx \tag{13}$$

or in its equivalent convolution expressions,

$$\psi_R(k)=\int_{-\infty}^{\infty} \psi_s(\zeta)\, H(k-\zeta)\, d\zeta$$

$$=\int_{-\infty}^{\infty} \psi(\zeta) \sum_{-\infty}^{\infty} \delta(\zeta - n\Delta k)\, H(k-\zeta)\, d(\zeta)\ ,$$

$$\psi_R(k)=\sum_{-\infty}^{\infty} \psi(n\Delta k)\, H(k-n\Delta k)\ , \tag{14}$$

where

$$H(k)=\int_{-\infty}^{\infty} G(x) \exp(-i2\pi kx)\, dx\ , \tag{15}$$

$$G(x)=\int_{-\infty}^{\infty} H(k) \exp(i2\pi kx)\, dk\ . \tag{16}$$

As shown by Eq. (14), the reconstruction operation is one of interpolation; that is, the value of ψ for a value of k lying between $n\Delta k$ and $(n+1)\Delta k$ is obtained as a weighted sum of the samples of ψ around that value.

The most desirable shape for $G(x)$ would, of course, be a rectangular window

$$G(x)=1 \quad (\tfrac{1}{2}D \leq x \leq \tfrac{1}{2}D)\ ,$$
$$=0 \quad (\tfrac{1}{2}D > x > \tfrac{1}{2}D)\ .$$

Unfortunately, as is well known, the $H(k)$ function for such a window $(\sin\pi kD)/\pi kD$ extends over a range of $\pm\infty$. In order to carry out the reconstruction in a practical way, it is necessary to restrict the interpolation sum of Eq. (14) to a limited set of samples around the interpolation point. This will give rise to an imperfect reconstruction because the aliased spectrum cannot be totally rejected. As a measure of the imperfection, the rms deviation of the filtered, aliased spectrum from the original spectrum will be computed as follows: Assuming a uniformly weighted aperture, $a_x(x)=1$ and a $G(x)$ normalized to a mean value of 1.0 at $x=0$,

$$\mathrm{ERR}=\left[\left(\int_0^{D/2}[G(x)-1]^2\, dx+\int_{D/2}^{\infty} G^2(x)\, dx\right) \Big/ \frac{D}{2}\right]^{1/2}\ . \tag{17}$$

This error serves as a criterion for the selection of both a suitable sampling period Δk and an appropriate interpolation algorithm.

IV. SOME INTERPOLATION ALGORITHMS

Several methods of interpolation are of interest for the reconstruction process. These, along with their associated $G(x)$ spectral windows and their interpolation errors are given below.

A. Boxcar smoothing

Whenever the continuous function $\psi(k)$ is represented by a set of discrete values, this form of interpolation is imposed. That is, the function is considered to have the value ψ_j in the interval $(k_j-\frac{1}{2}\Delta k<k<k_j+\frac{1}{2}\Delta k)$. The algorithm is defined as

$$H_A(k)=1,\ 0\leq|k|\leq\tfrac{1}{2}\Delta k\ ,$$
$$H_A(k)=0\ ,\ \tfrac{1}{2}\Delta k<|k|\ . \tag{18}$$

Its spectral window is

$$G_A(x)=\sin(\pi x\Delta k)/\pi x\Delta k\ .$$

The magnitude of this window function, shown in Fig. 1, features an infinite set of nulls, each centered on a multiple of the sampling frequency. Thus for a sufficiently high sampling rate, the relative width of the aliased spectra of ψ, which are centered around multiples of the sampling frequency, can be made small enough to achieve any desired degree of rejection. This is the ultimate interpolation imposed on the sampled function after a sufficiently high effective sampling frequency has been achieved by more efficient interpolation algorithms. The errors inherent in the gradual roll off within the desired pass band are correspondingly reduced by an increase in effective sampling frequency.

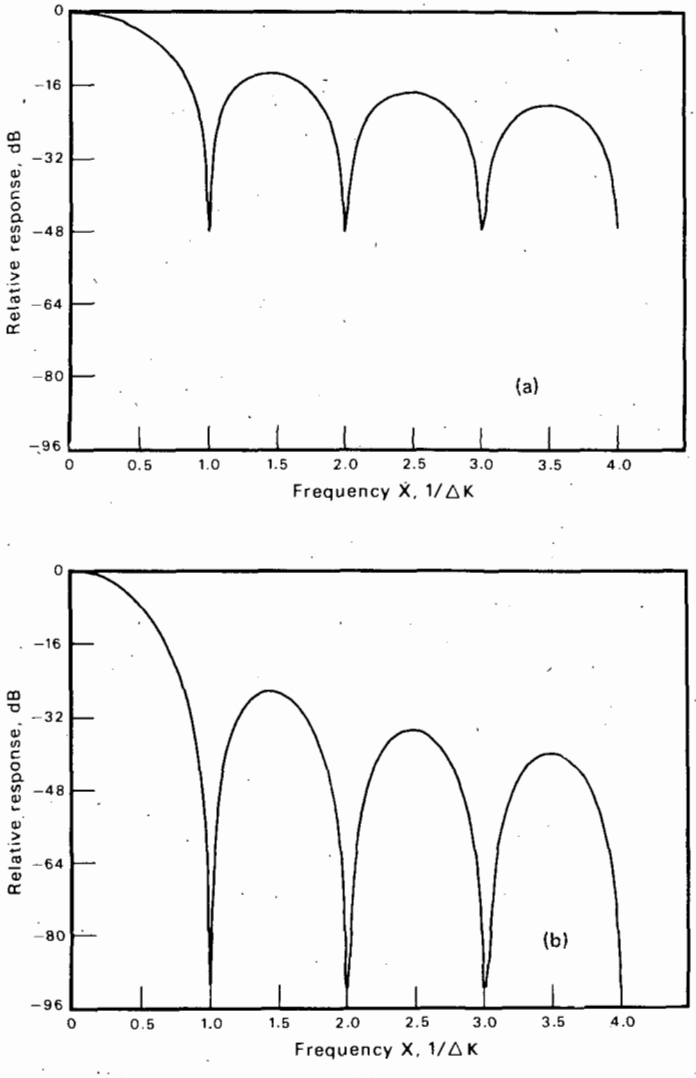

FIG. 1. (a) Frequency window $G(x)$ for boxcar interpolation and (b) frequency window $G(x)$ for linear interpolation.

B. Linear interpolation

In this method the values of the function at each end of a sampling interval are combined with proportional weights to obtain intermediate values. The common application of this method is the practice of connecting points on a graph by straight-line segments. The window functions are

$$H_B(k) = 1 - |k/\Delta k| \quad (0 < |k| \leq \Delta k),$$
$$= 0 \quad (\Delta k < |k| \leq \infty), \tag{19}$$
$$G_B(x) = [\sin(\pi x \Delta k)/\pi x \Delta k]^2 .$$

This window, also plotted in Fig. 1, exhibits deeper nulls, at the same multiples of the sampling frequency, than the previous one. It suffers, however, from a greater distortion within the pass band inasmuch as the roll off in that region is also greater than that for the boxcar smoothing.

C. Trigonometric interpolation

The trigonometric-interpolation method is ideally suited to uniformly spaced data samples and is closely related to the Fourier-series expansion. The method is described by Lanczos[5] in some detail. His formulation can be used to derive an expression for the weighting coefficients corresponding to the window function $H(k)$.

In brief: Given a set of $n+1$ uniformly spaced samples f_j of an arbitrary function $f(x)$, first scale x such that the interval $0 \leq x \leq n$ corresponds to the sampling interval $0 \leq j \leq n$. Now, generate a new function

$$h(x) = f(x) - (1 - x/n) f_0 - (x/n) f_n \quad (0 \leq x \leq n), \tag{20}$$
$$h(-x) = -h(x) \quad (-n \leq x \leq 0).$$

$h(x)$ is now an odd function of x. It is identically zero at $-n$, 0, n and the first derivative is continuous across these nodes. Thus, it can be readily expanded in the pure sine series

$$h(x) = \sum_{p=1}^{n-1} b_p \sin\left(\frac{p\pi x}{n}\right), \tag{21}$$

where

$$b_p = \frac{2}{n} \sum_{\alpha=1}^{n-1} h_\alpha \sin\left(\frac{p\alpha\pi}{n}\right). \tag{22}$$

TABLE I. Trigonometric weights for midinterval interpolation.[a]

	$(n+1)$			
m	2	4	8	16
1	0.500	0.5773	0.6259	0.6343
2		−0.0773	−0.1791	−0.2052
3			0.0688	0.1155
4			−0.0156	−0.0740
5				0.0484
6				−0.0297
7				0.0142
8				−0.0035

[a]In this table $(n+1)$ is the number of samples used for interpolation and m is the index of symmetrical displacement from the central interval.

Combining (20)–(22)

$$f(x) = \left\{ \sum_{p=1}^{n-1} \sum_{\alpha=1}^{n-1} \frac{2}{n} \left[f_\alpha - f_0\left(1 - \frac{\alpha}{n}\right) - f_n \frac{\alpha}{n} \right] \sin\left(\frac{\pi p \alpha}{n}\right) \times \sin\left(\frac{\pi p x}{n}\right) \right\} + f_0\left(1 - \frac{x}{n}\right) + \frac{x}{n} f_n . \tag{23}$$

We are interested in a set of interpolation weights such that

$$f(x) = \sum_{q=0}^{n} \varphi_q(x) f_q \tag{24}$$

collecting coefficients in Eq. (23) the set of $n+1$ weights can be written as

$$\varphi_0(x) = \left(1 - \frac{x}{n}\right) + \frac{2}{n} \sum_{p=1}^{n-1} \sum_{\alpha=1}^{n-1} \left(\frac{\alpha}{n} - 1\right) \sin\left(\frac{\pi p \alpha}{n}\right) \sin\left(\frac{\pi p x}{n}\right),$$
$$\varphi_q(x) = \frac{2}{n} \sum_{p=1}^{n-1} \sin\left(\frac{\pi p q}{n}\right) \sin\left(\frac{\pi p x}{n}\right) \quad (1 \leq q \leq n-1),$$
$$\varphi_n(x) = \frac{x}{n} - \frac{2}{n} \sum_{p=1}^{n-1} \sum_{\alpha=1}^{n-1} \frac{\alpha}{n} \sin\left(\frac{\pi p \alpha}{n}\right) \sin\left(\frac{\pi p x}{n}\right). \tag{25}$$

It is interesting to note that, if only two samples are used, $n=1$ and the summations have zero terms so that the interpolation reverts to $f(x) = (1-x)f_0 + xf_1$, which is just the linear interpolation mentioned above. Further, by virtue of the orthogonality of the functions, if x is an integer corresponding to one of the sample points, the weight for that sample point equals one and all the other weights are equal to zero, confirming that, indeed, the expansion reproduces the sample points exactly.

For any value of x within a sampling interval, a set of φ's may be calculated. In our application of the trigonometric interpolation the weighting coefficients for the middle of the sampling interval will be used. These weights are given in Table I for interpolation windows utilizing 2, 4, 8, and 16 samples.

The addition of a set of midinterval interpolated points doubles the effective sampling frequency. This in effect aliases the continuous $G(x)$ window associated with the continuously distributed weights of Eq. (25) around multiples of $2/\Delta k$, twice the original sampling frequency. The aliased windows can be computed empirically from an FFT transform of the delta-function sequence made up of the set of midinterval weights superimposed on the integer interval weights, e.g., for $(n+1)=4$ the set of uniformly spaced delta functions to be transformed have amplitudes of −0.0773, 0, 0.5773, 1.0, 0.5773, 0, −0.0773. These aliased windows are shown in Fig. 2 for the set of interpolation weights of Table I. In each case the aliased pass bands centered at $2n/\Delta k$ are apparent. As one would expect, increasing the number of samples improves (flattens) the acceptance band and also improves (steepens) the rejection band.

For reasons stated later, the complete interpolation will be performed as a sequence of midinterval interpolations. The first step of the sequence effectively doubles the sampling frequency, rejecting the odd aliased aperture bands but retaining the even ones in accord with the window functions of Fig. 2. The set of

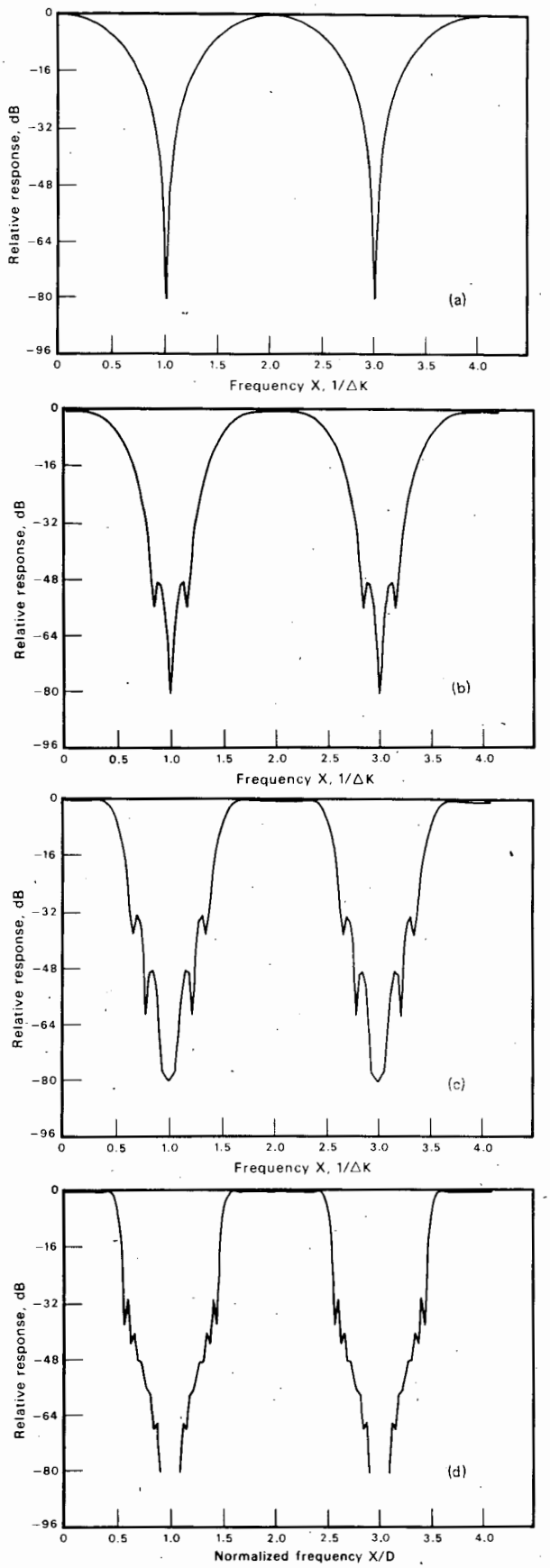

FIG. 2. (a) Aliased frequency window for 2-weight trigonometric interpolation, (b) aliased frequency window for 4-weight trigonometric interpolation, (c) aliased frequency window for 8-weight trigonometric interpolation, and (d) aliased frequency window for 16-weight trigonometric interpolation.

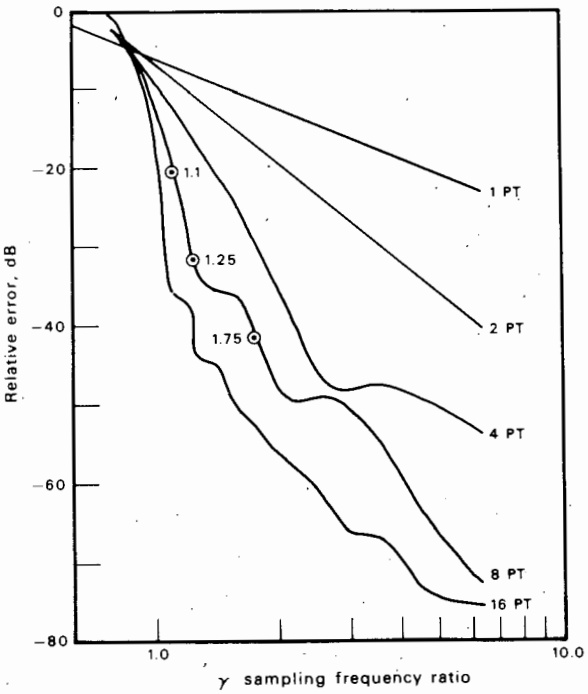

FIG. 3. Relative rms interpolation errors versus the ratio of the sampling frequency to Nyquist (D). Errors observed for the reconstructions of Figs. 4, 5, and 6 are plotted as circles for comparison.

samples now represent aperture bands aliased at $2n/\Delta k$, but the second interpolation step will be operating at an effective sampling frequency of $4/\Delta k$ for m odd and reject the bands for m even. Thus, the stop bands of successive midinterval interpolation windows are precisely aligned with the pass bands of the previous step. Also, of course, the relative width of the aperture band with respect to the sampling frequency is reduced by a factor of 2 with each iteration step so that lower-order windows may be used. The final iteration step in every case will use one of the windows of Fig. 1, i.e., either the boxcar window which corresponds to the representation of the array factor or of the array pattern by a set of numbers, or the linear window which corresponds to the representation of the pattern by a continuous plot with straight-line segments connecting the adjacent interpolated points. Each of these final windows has a rejection null centered on all of the residual aliased aperture bands resulting from the prior succession of interpolation steps.

V. INTERPOLATION ERRORS

The precision of the interpolation restoration can be estimated by numerically integrating Eq. (17) over the response curves of Figs. 1 and 2. These errors are presented in Fig. 3 as a function of γ, the ratio of the effective sampling frequency to Nyquist; i.e., units of D. Here one sees the dramatic results of the trigonometric interpolation.

As an aid to formulating a complete interpolation algorithm, the intersections of the error curves of Fig. 3 with various error levels are tabulated in Table II. As the number of interpolation points is increased there

TABLE II. Minimum sampling frequency ratio γ for interpolation errors.

Root-mean-square error (dB)	Number of points used for interpolation 1	2	4	8	16
−20	4.62	2.08	1.35	1.11	1.01
−30	13.80	3.63	1.78	1.23	1.06
−40	39.80	6.31	2.19	1.74	1.26

must be a corresponding addition to the number of data points computed. Thus, if the number of samples of k which are to be computed is small enough, the advantage gained in the reduction of sampling frequency by using more points in interpolation is offset by the number of additional points required for the interpolation filter. The minimum ranges of k in Nyquist intervals $(1/D)$ at which a given interpolation filter will save computation time are given in Table III for the conditions of Table II. These are limiting values obtained by assuming that the number of elements in the array is very large compared to the number of points used for interpolation. Except for very crude and broad patterns (−20 dB error, $-5 < kD < 5$) interpolation with 8 or even 16 weights is worthwhile. A 16-weight window will allow sampling at 1.06 Nyquist with an error of −30 dB.

VI. INTERPOLATION SEQUENCES

As an example of the selection of a complete interpolation algorithm, consider a requirement for an error of −40 dB and a range for kD of 20. From Table III we select a 16-point filter for the first step, from Table II the frequency ratio $\gamma = 1.26$. The first step increases γ to 2.52. Again from Table II the next step can use a 4-point filter to increase γ to 5.04. However, this frequency ratio is below that required for the 2-point interpolation so the 4-point filter has to be repeated to give a γ of 10.08. At this point the array-factor data can be plotted since the 2-point filter is inherent in the plotting process. If, on the other hand, one wishes to represent the array factor as a numerical data array, γ has to be greater than 39.8 to retain the −40-dB error band. Thus the sequences are as follows:

−40 dB (plotting): 16, 4, 4,

−40 dB (data array): 16, 4, 4, 2, 2.

This example brings out two reasons why the method of sequential midinterval interpolation steps was chosen rather than some other combination of fractional interval interpolation. First, a single, midinterval interpolation step brings the process well within range for a lower-order simpler window. Second, the midinterval 2 sample window for a final iteration step is much simpler to implement than any other fractional interval in that, with integer arithmetic, it consists of a simple binary sum and right shift. This can be quite important because this last step consumes a significant portion of the interpolation computing time as will be evident in the example later.

Before illustrating the interpolation process, a point should be raised regarding the spectrum of the array pattern. Since the array pattern is derived from the square of the array factor it has a spectral width twice as great as the aperture D. Thus, the array pattern requires a sampling frequency twice that for the array factor in order to avoid aliasing. In order to accommodate the broader spectrum an additional step of midinterval interpolation should be imposed before converting the array factor to an array pattern.

VII. PLOTTING EXAMPLE

An interpolation sequence of 8, 4, 2 has been chosen for the purpose of illustration. Figures 4, 5, and 6 are $[\sin(\pi k)/\pi k]^2$ array patterns interpolated by the use of the above sequence on the array factor, $\sin(\pi k)/\pi k$. The squares (plotted with a +10-dB offset) represent the data points used for interpolation and the line connecting these points is the uninterpolated plot. Sampling frequency ratios, referred to D, the Nyquist frequency, of 1.1 (Fig. 4), 1.25 (Fig 5), and 1.75 (Fig. 6) were used. The rms deviations of the interpolated data points from the true values are superimposed on the 8-point curve of Fig. 3. These show good agreement between the predicted errors and those actually realized. The three cases correspond to, roughly, −20-, −30-, and −40-dB errors.

Although the central lobe of the interpolated array pattern of Fig. 4 (1.1×Nyquist) matches the $[\sin(kx)/kx]^2$ pattern quite closely, the deviations of the side-lobe peaks from the theoretical values are quite large. In the region of the dip in the side-lobe pattern an error of 6–7 dB occurs. This is not inconsistent with the −20-dB error level because the side-lobe level itself is down to −30 dB and only small absolute errors are required to change the relative levels of these low side lobes significantly.

At the slightly higher sampling frequency of Fig. 5 (1.25 Nyquist) the side-lobe behavior is much better, with errors on the order of 1 dB. The side-lobe levels of Fig. 6 (1.75 Nyquist) are correct to within about 0.2 dB.

VIII. UNIFORM SAMPLING OF THE ARRAY APERTURE

At this point it is evident that both the aperture and array-factor fields are represented by sets of discrete data samples. The array-factor set is generally uniformly sampled but the aperture set is not. If both sets were to be composed of the same number of uniformly spaced samples, the transform of Eq. (1) would

TABLE III. Minimum $k/\Delta k$ range (Nyquist intervals) at which a given interpolation filter saves computation time.

Root-mean-square Error (dB)	Number of points used for interpolation 1	2	4	8	16
−20		0.8	3	15	80
−30		0.2	1.1	7.3	47
−40		0.06	0.5	9	17

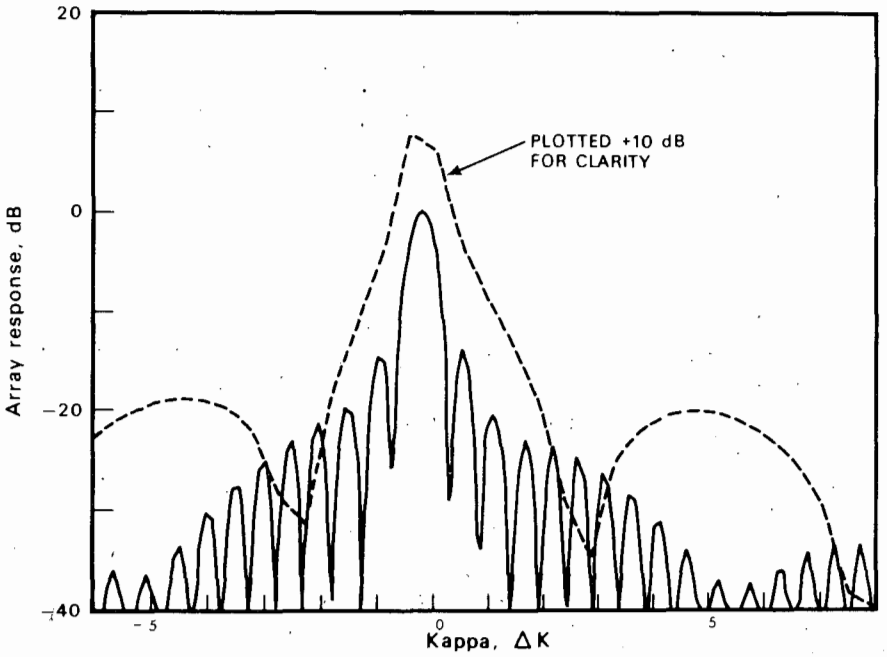

FIG. 4. Array pattern generated at a sampling frequency of 1.1×Nyquist. Squares are the computed pattern points and the associated uninterpolated plots are offset +10 dB. The solid curve is derived by an 8, 4, 2 trigonometric-interpolation sequence from the same set of computed data points (squares). The relative rms error is −20.6 dB.

take on the form of a discrete Fourier transform (DFT) and the powerful fast-Fourier-transform (FFT) algorithms could be exploited in the computation. As the aperture size and the number of elements are increased the computational efficiency of the FFT algorithms becomes more important.

One approach to obtaining a uniformly sampled array aperture is to place the elements, by design, on the interstices of a coordinate lattice corresponding to an appropriate uniform sample set. This approach, although simple and effective, is unduly restrictive in the selection of element positions. As an alternative, for any arbitrary array, each delta function of the array-element distribution must be expanded into a set of uniformly spaced spatial samples in the vicinity of the element. These individual sets are then summed to obtain the total array-aperture distribution. This expansion is, in essence, the inverse of the interpolation of the previous section where the value at an arbitrary point was obtained from the values of neighboring elements of a uniformly sampled data set.

Because the value of the delta function is zero at both ends of the expansion interval, a trigonometric sine-series expression is appropriate. The approach is to transform the delta function into a finite set of sine-series coefficients and then, through an inverse dis-

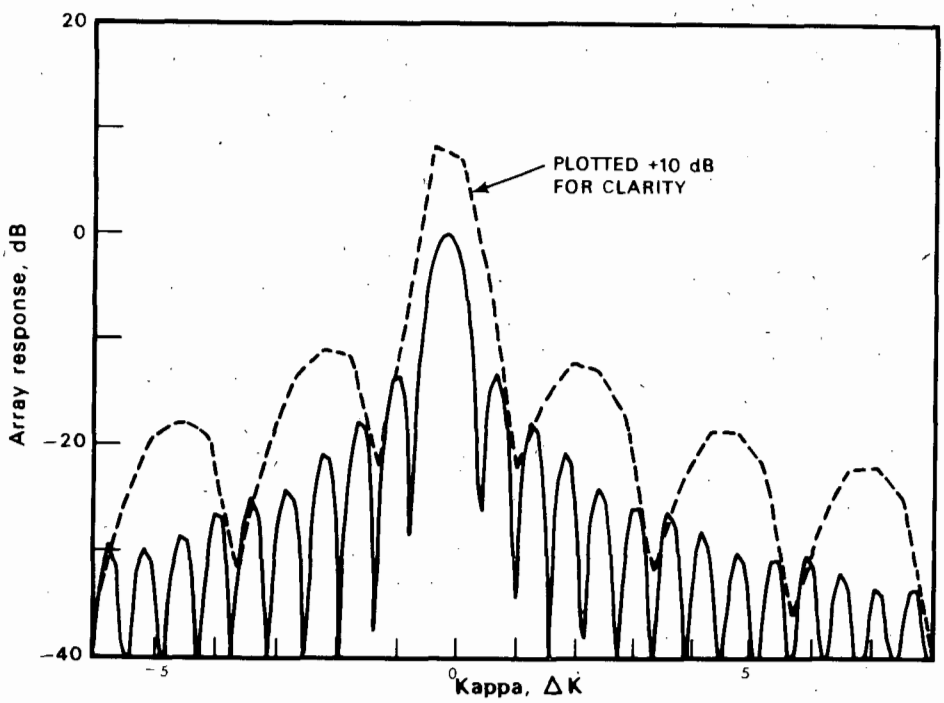

FIG. 5. The same presentation as Fig. 4 at a sampling frequency of 1.25×Nyquist. The relative rms error is −31.6 dB.

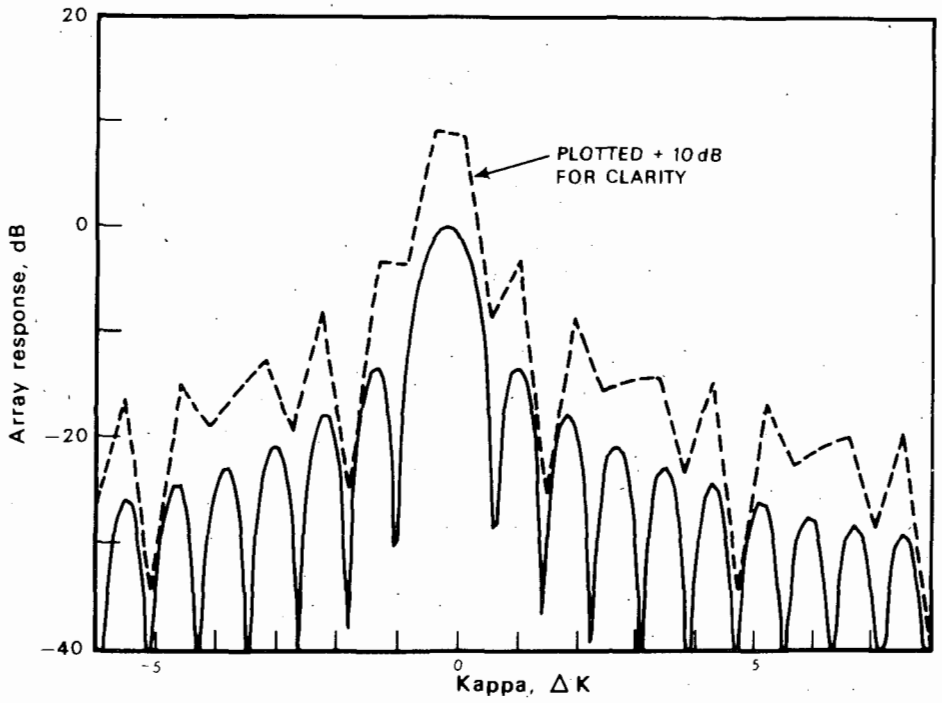

FIG. 6. The same presentation as Fig. 4 at a sampling frequency of 1.75× Nyquist. The relative rms error is −41.5 dB.

crete Fourier Transform obtain a corresponding set of uniform spatial samples which have spectral coefficients that approximate the uniform spectrum of the delta function.

The procedure in one dimension is as follows: Let

$$\delta(x-x_0)=\sum_{p=1}^{m-1} b_p \sin px \quad \text{with } 0\le x\le \pi, \tag{26}$$

which is readily solved for the coefficients

$$b_p=(2/\pi)\sin(px_0)\ . \tag{27}$$

An inverse discrete Fourier transform then gives a set of sample coefficients

$$g_\alpha=\frac{2}{m}\sum_{p=1}^{m-1}\sin px_0 \sin\left(p\alpha\frac{\pi}{m}\right)\ , \quad \alpha=1,2,\ldots,m-1\ . \tag{28}$$

This set of coefficients will not be exactly normalized for all values of x_0 because of truncation errors, so a better approximation to the delta function is achieved by the set of coefficients

$$h_\alpha=g_\alpha\Big/\sum_{\alpha=1}^{m-1} g_\alpha\ . \tag{29}$$

More explicity, if we arbitrarily select the number of nonzero coefficients in the sine series $(m-1)$ to be even, an expression for the sample coefficients as a function of the fraction of the center interval at which x_0 is located is

$$g_\alpha(\epsilon)=\frac{2}{m}\sum_{p=1}^{m-1}\sin\left[p\frac{\pi}{m}\left(\frac{m-1}{2}+\epsilon\right)\right]\sin\left(p\alpha\frac{\pi}{m}\right)\ . \tag{30}$$

This direct expansion of the delta function exhibits the troublesome Gibbs oscillations in its power spectrum as a consequence of the truncation of the infinite series. Lanczos[6] recommends what he terms "σ smoothing" which applies a $\sin(x)/x$ weighting function to the coefficients of the truncated trigonometric series. These smoothed coefficients are given by

$$g'_\alpha(\epsilon)=g_\alpha(\epsilon)\,\frac{\sin[\pi(\alpha-\frac{1}{2}m)/\frac{1}{2}m]}{\pi(\alpha-\frac{1}{2}m)/\frac{1}{2}m} \tag{31}$$

TABLE IV. Unsmoothed coefficients for an 8-point delta-function expansion.

	α								
e	1	2	3	4	5	6	7	8	C. G.
0	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	4.0000
0.1	−0.0148	0.0354	−0.0829	0.9867	0.1087	−0.0468	0.0245	−0.0109	4.1013
0.2	−0.0271	0.0642	−0.1447	0.9412	0.2339	−0.0946	0.0488	−0.0216	4.2018
0.3	−0.0359	0.0843	−0.1838	0.8655	0.3694	−0.1388	0.0702	−0.0309	4.3017
0.4	−0.0407	0.0946	−0.2004	0.7639	0.5082	−0.1743	0.0864	−0.0378	4.4010
0.5	−0.0412	0.0950	−0.1962	0.6424	0.6424	−0.1962	0.0950	−0.0412	4.5000
0.6	−0.0378	0.0864	−0.1742	0.5082	0.7639	−0.2004	0.0946	−0.0407	4.5990
0.7	−0.0309	0.0702	−0.1388	0.3694	0.8655	−0.1838	0.0843	−0.0359	4.6983
0.8	−0.0216	0.0488	−0.0946	0.2339	0.9412	−0.1447	0.0642	−0.0271	4.7982
0.9	−0.0109	0.0245	−0.0468	0.1087	0.9867	−0.0829	0.0354	−0.0148	4.8987
1.0	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	5.0000

TABLE V. Smoothed coefficients for an 8-point delta-function expansion.

	α								
e	1	2	3	4	5	6	7	8	C. G.
0	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	4.0000
0.1	−0.0039	0.0201	−0.0690	0.9735	0.1073	−0.0390	0.0139	−0.0029	4.1001
0.2	−0.0071	0.0361	−0.1191	0.9182	0.2281	−0.0779	0.0274	−0.0057	4.2001
0.3	−0.0093	0.0470	−0.1501	0.8373	0.3574	−0.1133	0.0391	−0.0080	4.3001
0.4	−0.0105	0.0524	−0.1628	0.7352	0.4891	−0.1415	0.0479	−0.0098	4.4000
0.5	−0.0106	0.0526	−0.1591	0.6171	0.6171	−0.1591	0.0526	−0.0106	4.5000
0.6	−0.0098	0.0479	−0.1415	0.4891	0.7352	−0.1628	0.0524	−0.0105	4.6000
0.7	−0.0080	0.0391	−0.1133	0.3574	0.8373	−0.1501	0.0470	−0.0093	4.6999
0.8	−0.0057	0.0274	−0.0779	0.2281	0.9182	−0.1191	0.0361	−0.0071	4.7999
0.9	−0.0029	0.0139	−0.0390	0.1073	0.9735	−0.0690	0.0201	−0.0039	4.8999
1.0	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	5.0000

with the corresponding normalized coefficients,

$$h'_\alpha(\epsilon) = g'_\alpha(\epsilon) \Big/ \sum_{\alpha=1}^{m-1} g'_\alpha(\epsilon) \,. \tag{32}$$

Sets of unsmoothed and smoothed coefficients for an 8-sample expansion are tabulated in Tables IV and V. An indication of the accuracy with which these expansions represent the delta function is given by the center of gravity of the set, also listed in Tables IV and V, which is a measure of the phase error in the low-frequency region of the transformed spectrum. The accuracy with which the center of gravity tracks the actual location of the delta function is considerably better in the smoothed sets (±0.0001 of an interval), than in the unsmoothed sets (±0.0018 of an interval). However, the maximum error for either has a negligible effect on the computed pattern as given by Eq. (8).

A look at the spectrum of the sample coefficients gives an even better understanding of the fidelity of representation. The maximum deviation of the power spectrum from the uniform spectrum of the delta function occurs for the midinterval set. Relative power spectra for several unsmoothed midinterval sample sets are plotted in Fig. 7; corresponding spectra for smoothed sets are shown in Fig. 8. The expansion, in effect, replaces the element with a directional subarray whose directivity pattern is represented by the power spectrum of the sample set. This varies from a uniform spectrum when the location of the element coincides with a sample position to the maximum deviation shown in Figs. 7 and 8 when the location of the element falls in the middle of a sampling interval.

Qualitatively, the errors indicated by the extreme deviations of the power spectrum can be related to the array pattern error of Eq. (8). For example, the unsmoothed 4-point expansion shows a +1.2-dB excursion at the midpoint of the transform spectrum. If this change in amplitude from 1.0 to 1.14 were considered to be distributed uniformly in a random manner over the element set, it would give rise to a $\sigma_a^2 = \frac{1}{12}(0.14)^2 = 0.0016$. Thus, even this crude 4-point expansion produces an error nearly 30 dB below the mean sidelobe level. Because the representation is so good, the dominant critierion for selection of the size of the set for expansion of the delta function is that of the usable fraction of the transform field. Figure 9 summarizes the pass bands of the spectra of Figs. 7 and 8. The unsmoothed expansions provide a significantly larger usable band, and the smoothed versions would only be required for very precise computations or for super gain arrays with large positive and negative weights, where the array sum becomes very sensitive to errors in either phase or amplitude.

Extending the expansion to two or three dimensions is straightforward. For two dimensions

$$\delta(x - x_f,\ y - y_f) = \sum_{p=1}^{m-1} \sum_{q=1}^{m-1} b_{pq} \sin(px) \sin(qy) \,,$$

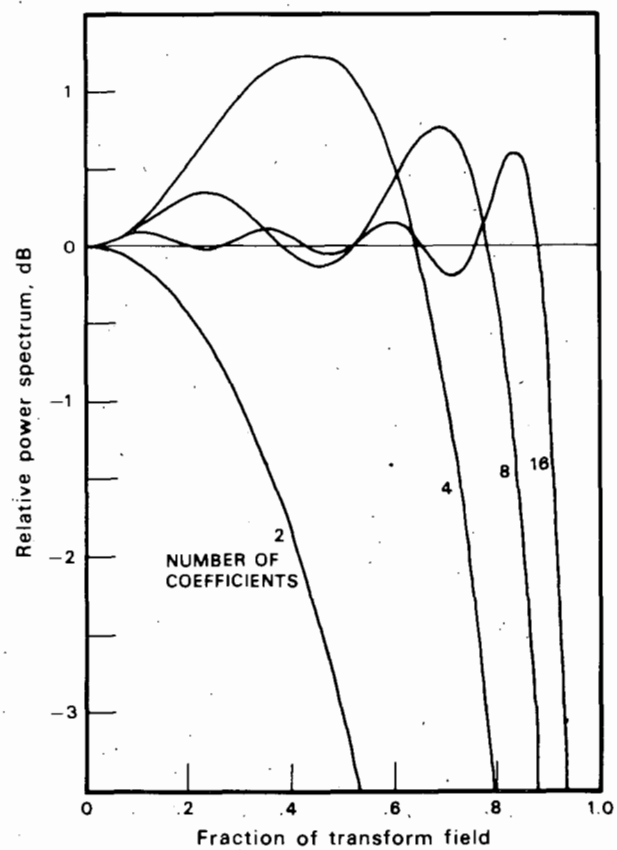

FIG. 7. Transform spectra for unsmoothed midinterval delta-function expansion.

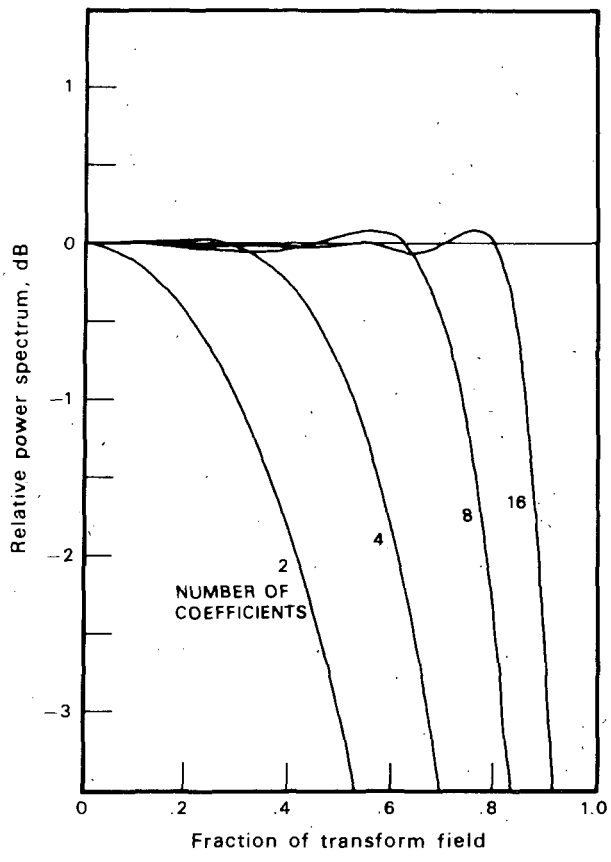

FIG. 8. Transform spectra for τ smoothed midinterval delta-function expansion.

with $0<x\leq\pi$, $0<y\geq\pi$,

$$g_{\alpha\beta}(\epsilon_x,\ \epsilon_y)=\frac{4}{m^2}\sum_{p=1}^{m-1}\sum_{q=1}^{m-1}\sin\left[p\frac{\pi}{m}\left(\frac{m-1}{2}+\epsilon_x\right)\right]\sin\left(p\alpha\frac{\pi}{m}\right)$$

$$\times\sin\left[q\frac{\pi}{m}\left(\frac{m-1}{2}+\epsilon_y\right)\right]\sin\left(q\beta\frac{\pi}{m}\right),$$

$$h_{\alpha\beta}(\epsilon_x,\ \epsilon_y)=g_{\alpha\beta}(\epsilon_x,\ \epsilon_y)\Big/\sum_{\alpha=1}^{m-1}\sum_{\beta=1}^{m-1}g_{\alpha\beta}(\epsilon_x,\ \epsilon_y).$$

If all of the elements of the array have uniform amplitude weights, there may be a computational advantage to precomputing and storing this four-dimensional matrix in a look-up table, providing the required memory is available. Otherwise, the sample coefficients can be readily generated from a much smaller two-dimensional table corresponding to Table IV or V.

Computing time for this expansion is proportional to $N(m-1)^2$ for a two-dimensional array so the selection of m depends on the trade off between the increase in the computing time for generating the expansion coefficients as m is increased and the reduction in DFT computing time achieved by the increase in the fraction of the transformed field which is usable.

IX. APPLYING THE DIGITAL FOURIER TRANSFORM

Now that criteria have been established for uniformly sampling both the $\vec{k}$ domain and the array aperture,

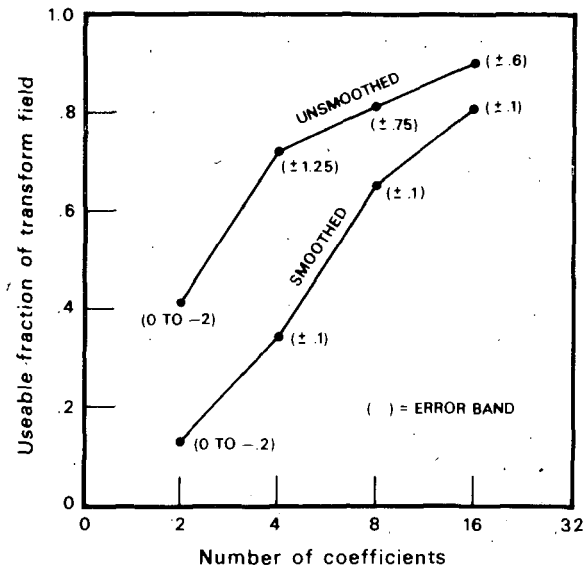

FIG. 9. Usable fraction of the transform field versus the number of coefficients in the delta-function expansion.

the first step in applying the DFT to the problem is to select the number of samples for the transform field. Figure 10 illustrates the relationship between the transform parameters. In general, the maximum range of k, k_{max}, will be larger than the visible range, $4/\lambda$, by a factor $\nu\geq 1$ and the maximum range in the spatial coordinate, x_{max}, will be larger than the aperture dimension D by a factor $\gamma\geq 1$. The sampling interval in x is $\Delta x=k_{max}^{-1}$ and the sampling interval in k is $\Delta k=x_{max}^{-1}$. Combining these relationships, the minimum number of samples in the DFT field is given by $L_{min}=4\gamma\nu\ D/\lambda$.

The selection of γ is predicted on two criteria:

(1) γ is greater than or equal to the sampling frequency ratio selected for interpolation of $\psi_s(k)$;

(2) $\gamma\geq(D+m\Delta x)/D$, where m is the number of coef-

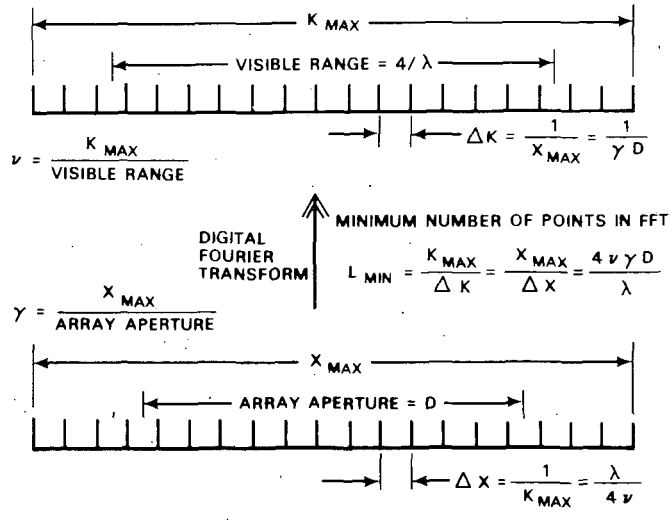

FIG. 10. Parameter relationships in the discrete Fourier transform.

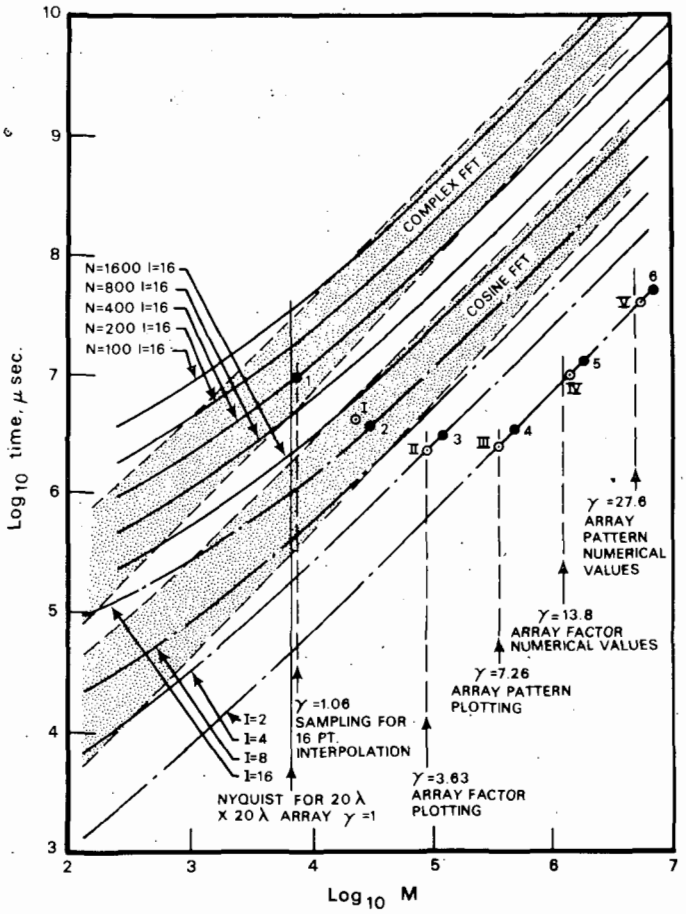

FIG. 11. Computation times for various algorithms versus the number of samples in the output field.

ficients used for the $\delta(x_j)$ expansion.

Of these two criteria, usually (1) is the controlling one.

A single criterion is used for the selection of ν:

$$\nu \geq \frac{4/\lambda + n\Delta k}{\text{usable fraction of } k_{\max} \text{ for selected } m} ,$$

where n is the number of samples used for $\psi(k)$ interpolation. These above criteria apply separately to the separable transformations. That is, $L_x\min$, ν_x, γ_x, D_x are related to each other but independent of $L_y\min$, ν_y, γ_y, D_y or $L_z\min$, ν_z, γ_z, D_z.

In order to exploit the use of packaged FFT algorithms of radix 2, it is necessary to select an integer power of 2 field size; i.e., $L = 2^p \geq L\min$. The scale factor Δx is used to relate array-element locations to sample indices and a fractional interval for delta function expansion. Note that the scale factors for x, y, z will usually be different except for the special case when the aperture size is the same in two or more coordinates.

The advantages of using a symmetrical array distribution appear in the application of the FFT algorithm also. By imposing the condition of a real symmetrical distribution on the array aperture, a cosine transform which utilizes a complex FFT working on $\frac{1}{4}$ as many sample points with a corresponding savings in time of about 4:1 may be used. An algorithm for the cosine transform is described by Cooley *et al.*[7] This savings of 4:1 becomes 16:1 for a two-dimensional array and 64:1 for a three-dimensional array and thus is well worth consideration. If the cosine transform is used, only one quadrant of a two-dimensional array aperture would be used and thus in the expansion of elements into the uniform sample set, coefficients of elements in the (+, +) quadrant would overlap into adjacent quadrants. These coefficient values are merely folded back into the (+, +) quadrant to complete the array-distribution matrix.

X. COMPUTATION-TIME GUIDELINES

A simple computer model having a memory cycle time of 2.5 μsec, addition/subtraction of 6 μsec, multiplication/division of 7 μsec, and a cosine routine of 135 μsec will be assumed for the purpose of making a relative comparison of computation times for the techniques which have been discussed above.

Figure 11 displays approximate computation times for a two-dimensional array for the interpolation algorithm, FFT, and direct computations as a function of $\log_{10} M$, the number of samples in the $\psi_s(k)$ field. One set of lines is for the direct computation with N, the number of elements, as a parameter assuming $I = 16$, a 16-point interpolation. The use of a stored cosine table along with array symmetry is assumed for these times. The other set of lines represent interpolation times for 2–16 points of interpolation filtering. The complex and cosine FFT times are represented as bands signifying that the FFT size changes in discrete steps of a power of two in both dimensions and times are bounded by the bands shown. The FFT times do not include the increased computation required by the factor ν^2 in comparison with the direct computation.

Not displayed in Fig. 11 are the times for the delta-function expansions required for the FFT computations. For the two-dimensional case the time will be given approximately by $N(m-1)^2 \times 25$ μsec where $(m-1)$ is the number of coefficients in the expansion. For filled arrays (average spacing $\frac{1}{2}\lambda$) this time is comparable to the cosine FFT time at the Nyquist sampling rate for $m - 1 = 8$. As such it represents a minor but not negligible part of the overall computation time.

The sequential use of the interpolation algorithms will be illustrated in a specific example.

XI. COMPUTATION-TIME EXAMPLE

To summarize the reduction in computing time which can be effected by the techniques that have been discussed, let us choose a hypothetical array for which computing time would be significant.

Take as an example a set of $N = 400$ uniformly weighted elements randomly distributed over a plane aperture of $20\lambda \times 20\lambda$. Our goal will be to represent the array factor by a data array covering the two-dimensional visible region (a fully steered hemisphere) to an accuracy of 3% (– 30 dB). As a basis for estimating computing time we will use the cycle times assumed in the previous section.

The total visible range of k is $4/\lambda$. Therefore, the total number of Nyquist intervals in one dimension of our 20λ array is 80.

From Table II we find that, for a data array (single-point interpolation) the sampling frequency must be $\geq 13.8\times$Nyquist to obtain an error < -30 dB. Thus, the number of elements in one row of our data matrix is $80\times 13.8 = 1104$ for a total set of $M = (1104)^2 = 1.22\times 10^6$.

A. Direct

The direct computation of Eq. (3) then requires $2MN \times$ (computation time per operation) $= 2\times 400\times 1.22\times 10^6 \times(2\times 135 + 7\times 2.5 + 2\times 7 + 4\times 6)\times 10^{-6} = 1.58\times 10^5$ sec $= 44$ h. One might hesitate before embarking on such a computing exercise!

B. Symmetry

The first step is to introduce symmetry into the array distribution. The time now becomes $(M/4)\times(M/4)\times(2 \times 135 + 5\times 2.5 + 4\times 7 + 6) = \frac{1}{4}(1.22\times 10^6)\times\frac{1}{4}(400)\times(316) = 9.64\times 10^3$ sec $= 2.7$ h. This is a good start at improving the computer time.

C. Look-up table

Now use a stored table instead of the cosine routine. For an error of -30 dB a quantization step of 0.1 rad is required. The range of the argument of the function for the symmetrical case is $2\pi D/\lambda$. Thus, a table of 1250 elements will be required at an initial set-up time of $1250\times 135 + 2.5 + 6\times 10^{-6} = 0.18$ sec—negligible.

The computation time may be obtained directly from Fig. 11. $N = 400$, $I = 1.22\times 10^6$ gives a time of $10^{9.2} = 1.12\times 10^9$ μsec $= 0.31$ h.

D. Direct with interpolation

Finally, introduce interpolation using a 16, 4, 2, 2 sequence. Sampling the final field at Nyquist rate will require $M = 80\times 80 = 6400$ samples, 1.06 Nyquist would be 6784. From Fig. 11, the time required for 400 elements will be $t_1 = 9.55$ sec indicated by point 1 on the figure. The first interpolation step, a 16-point weighting window, at point 2, required $10^{6.56}$ μsec, $t_2 = 3.63$ sec. The second step, a 4-point window at point 3 takes $t_3 = 3.02$ sec. Next, a 2-point window at 4 requires $t_4 = 3.39$ sec. Finally, for the last step, also a 2-point window, at 5 $t_5 = 12.9$ sec. The total time $t_1 + t_2 + t_3 + t_4 + t_5 = 32.5$ sec.

E. Cosine FFT with interpolation

An alternate route is the FFT *approach.* We select an 8-coefficient unsmoothed expansion for the element locations for which, from Fig. 9, a ν of 1.25 is appropriate.

After some trial and error with the minimum γ ratios of Table II and the relative computing times involved in the FFT and interpolation steps, a sequence of cosine FFT, and a 4-point interpolation followed by two 2-point interpolations is selected with the controlling $\gamma = 3.63$ being the required sampling rate at the output of the 4-point interpolation. The sequence is shown as steps I, II, III, and IV in Fig. 11. This selection gives precisely the required γ at steps II and III and provides the minimum value achievable at step IV, slightly in excess of that required but less than that resulting from the above direct method. The value of γ for the cosine FFT is $3.63/2 = 1.81$. From these values of ν and γ, $L\min = 4\times 1.25\times 1.81\times 20 = 181$ therefore, $L = 256$. The cosine transform is carried out with $\frac{1}{4}$ as many samples, so the cosine transform involves a 64-point FFT for one dimension. For the two-dimensional transform the total number of points is $64^2 = 4096$. Processing time will be $4096\times\log_2(4096)\times 84$ μsec $= 4.13$ sec. The time required for the 8-coefficient expansion for 400 elements is 0.64 sec. The 4-point interpolation of step II takes 2.2 sec, step III, 2.5 sec, and step IV, 10 sec, for a total of 19.5 sec.

This represents a savings of about 40% over the direct/interpolation method but at the cost of somewhat greater programming complexity. For arrays with more elements the savings would, of course, be greater. If the complex form of the FFT were used directly instead of the more efficient cosine FFT algorithm the computing time would be essentially the same as for the direct/interpolation method in this example.

It is interesting to note from Fig. 11 that the use of interpolation offers more than an order-of-magnitude decrease in computing time for the γ required for array pattern plotting or for numerical representation over even the very efficient cosine FFT process.

It is also apparent that either the 32.5- or 19.5-sec times are much more tolerable figures than any of the other computing times, particularly more so than the 44-h "brute-force" approach, a ratio of 8000:1.

XII. CONCLUSION

In the face of a savings of 8000:1 what else can one conclude but that efficient transform algorithms combined with interpolation algorithms really pay their way in array-pattern computation.

[1]R. E. Collin and X. Zucker, in *Antenna Theory Part* 1, edited by R. E. Collin (McGraw–Hill, New York, 1969), p. 141.
[2]R. E. Collin, Ref. 1. p. 147.
[3]R. E. Collin, Ref. 1. 231, Eq. (6.40).
[4]R. B. Blackman and J. W. Tukey, "The Measurement of Power Spectra," Bell Syst. Tech. J. 37, 76 (1958).
[5]C. Lanczos, *Applied Analysis* (Prentice–Hall, Englewood Cliffs, 1956), pp. 229–241.
[6]C. Lanczos, Ref. 5, p. 225.
[7]J. W. Cooley, P. A. Lewis, and P. D. Welch, "The Fast Fourier Transform Algorithm: Programming Considerations in the Calculation of Sine, Cosine and Laplace Transforms," in *Digital Signal Processing*, edited by L. R. Rabiner and C. M. Rader (IEEE, New York, 1972).

DISTRIBUTION LIST

Chief of Naval Research
Department of the Navy
Arlington, Virginia 22217
- Code 200 (1)
- Code 222 (2)
- Code 102IP (1)
- Code 1020SC (1)
- Code 460 (1)
- Code 480 (1)
- Code 481 (1)
- Code 486 (1)

Director
Office of Naval Research
Branch Office
1030 East Green Street
Pasadena, Calfornia 91101 (1)

Commander
Naval Sea Systems Command
Washington, D.C. 20362
- Code 03E (1)
- Code 034 (1)
- Code 0342 (1)
- Code 036 (1)
- Code 06H1 (1)
- Code 06H2 (1)
- Code 09G3 (1)
- Code 92 (1)
- Code 662C14 (1)
- PMS 395-4 (1)
- PMS 402 B (1)
- Code 03423 (Mr. Francis J. Romano) (1)

Commander
Naval Air Systems Command
Washington, D.C. 20361
- Code 370 (1)
- Code 264 (1)

Commander
Naval Electronics
Systems Command
Washington, D.C. 20360
- Code PME-124 (1)

Chief of Naval Material
Department of the Navy
Washington, D.C. 20360
- Code PM-4 (1)
- Code 034 (1)
- Code ASW121 (1)

Chief of Naval Operations
Department of the Navy
Washington, D.C. 20350
- Code Op 03 (1)
- Code Op 32 (1)
- Code Op 098 (1)
- Code Op 02 (1)
- Code Op 095 (1)
- Code Op 23 (1)
- Code Op 967 (1)

Commanding Officer
Naval Ocean Research &
Development Activity (NORDA)
National Space Technology Laboratories
Bay St. Louis, Mississippi 39529
- Code 100 (1)
- Code 460 (1)
- Code 500 (1)

Oceanographer of the Navy
The Hoffman Building
200 Stoval Street
Alexandria, Virginia 22332 (1)

Director Strategic Systems
Projects Office (PM-1)
Department of the Navy
Washington, D.C. 20360
- Code NSP-20 (1)

Director
Defense Research & Engineering
The Pentagon
Washington, D.C. 20301
- Assistant Director (Sea
Warfare Systems) (1)

Assistant Secretary of the Navy
(Research & Development)
Department of the Navy
Washington, D.C. 20350 (1)

U.S. Naval Oceanographic Office
Washington, D.C. 20373
- Code 1640 (1)
- Code 3440 (1)

Commander Operational Test &
Evaluation Force
U.S. Naval Base
Norfolk, Virginia 23511 (1)

Commander, Submarine Force
U.S. Pacific Fleet
Fleet Post Office
San Francisco, California 96601 (1)

Commander
Submarine Group FIVE
Fleet Station Post Office
San Diego, California 92132 (1)

Commander
Third Fleet
U.S. Pacific Fleet, FPO
San Francisco, California 96610 (1)

Commander
Submarine Development Group ONE
Fleet Post Office
San Diego, California 92132 (1)

Commander
Submarine Development Group TWO
Naval Submarine Base-New London
Groton, Connecticut 06340 (1)

Commander, Surface Force
U.S. Atlantic Fleet
Norfolk, Virginia 23511 (1)

Commander, Surface Force
U.S. Pacific Fleet
San Diego, California 92155 (1)

Reprint Custodian
Department of Nautical Science
U.S. Merchant Marine Academy
Kings Point, New York 11024 (1)

Deputy Commander
Operational Test &
Evaluation Force, Pacific
U.S. Naval Air Station
San Diego, California 92135 (1)

Commander
Naval Ship Reserach &
Development Center
Bethesda, Maryland 20084 (1)

Naval Civil Engineering Laboratory
Port Hueneme, California 93041
Code L40 (1)
Code L42 (1)

Naval Facilities
Engineering Command
Washington, D.C. 20390
Code 03 (1)
Code 032C (1)

Commanding Officer
U.S. Naval Air
Development Center
Warminster, Pennsylvania 18974 (1)

Commander
Naval Ocean Systems Center
San Diego, California 92152
Code 6700 (2)

Officer In Charge
Naval Ship Research &
Development Center
Annapolis, Maryland 21402 (1)

Commanding Officer, Naval
Coastal Systems Laboratory
Panama City, Florida 32401 (1)

Commander
Naval Surface
Combat Systems Center
White Oak
Silver Spring, Maryland 20910 (1)

Director
U.S. Naval Research Laboratory
Washington, D.C. 20375
Code 2620 (1)
Code 2627 (1)
Code 8000 (1)
Code 8100 (1)

Commanding Officer
Naval Underwater Systems Center
Newport, Rhode Island 20844 (1)
Attn: John D'Albora

Commanding Officer
Naval Underwater Systems Center
New London, Connecticut 06320
Code 900 (1)
Code 905 (1)
Code 910 (1)
Code 930 (1)
Code 960 (1)

Commanding Officer
Naval Training Equipment Center
Orlando, Florida 32813
Tech Library (1)

Chief Scientist
Navy Underwater
Sound Reference Division
U.S. Naval Research Laboratory
P.O. Box 8337
Orlando, Florida 32806

Superintendent
U.S. Naval Postgraduate School
Monterey, California 93940 (1)

Director
Defense Documentation Center
(TIMA), Cameron Station
5010 Duke Street
Alexandria, Virginia 22314 (12)

Executive Secretary
National Academy of Sciences
2101 Constitution Avenue, N.W.
Washington, D.C. 20418 (1)

Supreme Allied Commander
U.S. Atlantic Fleet
ASW Research Center
APO
New York, New York 09019
Via: ONR 210
CNO OPO92D1
Secretariat of Military
Information Control
Committee (1)

National Oceanic & Atmospheric
Administration
6001 Executive Boulevard
Rockville, Maryland 20852 (1)

Director of Naval Warfare Analysis
Institute of Naval Studies
1401 Wilson Boulevard
Arlington, Virginia 22209 (1)

Institute for Defense Analyses
400 Army-Navy Drive
Arlington, Virginia 22202 (1)

Director
Woods Hole
Oceanographic Institution
Woods Hole, Massachusetts 02543 (1)

Meterological &
Astrophysical Abstracts
310 E. Capitol Street
Washington, D.C. 20003 (1)

Director
Applied Physics Laboratory
University of Washington
1013 East 40th Street
Seattle, Washington 98105 (1)

National Science Foundation
Washington, D.C. 20550 (1)

Director
Lamont-Doherty
Geological Observatory
Torrey Cliff
Palisades, New York 10964 (1)

Director
College of Engineering
Department of Ocean Engineering
Florida Atlantic University
Boca Raton, Florida 33431 (1)

Director
Applied Research Laboratory
Pennsylvania State University
P.O. Box 30
State College, Pennsylvania 16802 (1)

Director
University of Texas
Applied Research Laboratory
P.O. Box 8029
Austin, Texas 78712 (1)

TACTEC
Battelle Columbus Laboratories
505 King Avenue
Columbus, Ohio 43201 (1)

Director
Institute of
Ocean Science & Engineering
Catholic University of America
Washington, D.C. 20017 (1)

Director
Marine Research Laboratories
c/o Marine Studies Center
University of Wisconsin
Madison, Wisconsin 53706 (1)

Office of Naval Research
Resident Representative
c/o University of California,
San Diego
La Jolla, California 92093 (1)

University of California,
San Diego
La Jolla, California 92093
MPL Branch Office (5)

Marine Physical Laboratory
MPL-U-23/76

EFFICIENT COMPUTATION OF ARRAY PATTERNS by Victor C. Anderson, University of California, San Diego, Marine Physical Laboratory of the Scripps Institution of Oceanography, San Diego, California 92152. Journal of the Acoustical Society of America, Vol. 61, No. 3, March 1977.

The impact of a symmetrical array geometry, the use of a quantized stored cosine function, the exploitation of digital Fourier transform algorithms, and the application of trigonometric interpolation in the computation of array patterns is discussed. Careful selection of parameters permits sampling the array pattern only 6% above the theoretical Nyquist limit. Reconstruction of array patterns showing -20, -30, and -40-dB relative interpolation errors are presented. A saving of 8000:1 in computation time over direct "brute-force" array-pattern computation is illustrated for a hypothetical array.

II. Signal Processing
a. Instrumentation

Victor C. Anderson

Sponsored by
Office of Naval Research
N00014-75-C-0749
and
Advanced Research Projects Agency
N00014-76-C-1080

UNCLASSIFIED

Marine Physical Laboratory
MPL-U-23/76

EFFICIENT COMPUTATION OF ARRAY PATTERNS by Victor C. Anderson, University of California, San Diego, Marine Physical Laboratory of the Scripps Institution of Oceanography, San Diego, California 92152. Journal of the Acoustical Society of America, Vol. 61, No. 3, March 1977.

The impact of a symmetrical array geometry, the use of a quantized stored cosine function, the exploitation of digital Fourier transform algorithms, and the application of trigonometric interpolation in the computation of array patterns is discussed. Careful selection of parameters permits sampling the array pattern only 6% above the theoretical Nyquist limit. Reconstruction of array patterns showing -20, -30, and -40-dB relative interpolation errors are presented. A saving of 8000:1 in computation time over direct "brute-force" array-pattern computation is illustrated for a hypothetical array.

II. Signal Processing
a. Instrumentation

Victor C. Anderson

Sponsored by
Office of Naval Research
N00014-75-C-0749
and
Advanced Research Projects Agency
N00014-76-C-1080

UNCLASSIFIED

Marine Physical Laboratory
MPL-U-23/76

EFFICIENT COMPUTATION OF ARRAY PATTERNS by Victor C. Anderson, University of California, San Diego, Marine Physical Laboratory of the Scripps Institution of Oceanography, San Diego, California 92152. Journal of the Acoustical Society of America, Vol. 61, No. 3, March 1977.

The impact of a symmetrical array geometry, the use of a quantized stored cosine function, the exploitation of digital Fourier transform algorithms, and the application of trigonometric interpolation in the computation of array patterns is discussed. Careful selection of parameters permits sampling the array pattern only 6% above the theoretical Nyquist limit. Reconstruction of array patterns showing -20, -30, and -40-dB relative interpolation errors are presented. A saving of 8000:1 in computation time over direct "brute-force" array-pattern computation is illustrated for a hypothetical array.

II. Signal Processing
a. Instrumentation

Victor C. Anderson

Sponsored by
Office of Naval Research
N00014-75-C-0749
and
Advanced Research Projects Agency
N00014-76-C-1080

UNCLASSIFIED

Marine Physical Laboratory
MPL-U-23/76

EFFICIENT COMPUTATION OF ARRAY PATTERNS by Victor C. Anderson, University of California, San Diego, Marine Physical Laboratory of the Scripps Institution of Oceanography, San Diego, California 92152. Journal of the Acoustical Society of America, Vol. 61, No. 3, March 1977.

The impact of a symmetrical array geometry, the use of a quantized stored cosine function, the exploitation of digital Fourier transform algorithms, and the application of trigonometric interpolation in the computation of array patterns is discussed. Careful selection of parameters permits sampling the array pattern only 6% above the theoretical Nyquist limit. Reconstruction of array patterns showing -20, -30, and -40-dB relative interpolation errors are presented. A saving of 8000:1 in computation time over direct "brute-force" array-pattern computation is illustrated for a hypothetical array.

II. Signal Processing
a. Instrumentation

Victor C. Anderson

Sponsored by
Office of Naval Research
N00014-75-C-0749
and
Advanced Research Projects Agency
N00014-76-C-1080

UNCLASSIFIED