Proceedings of the Institute of Acoustics

HOW TO FIND SHADING COEFFICIENTS THAT PRODUCE AN ARBITRARY BEAMPATTERN FROM AN ARBITRARY ARRAY

Peter F. Dobbins

British Aerospace, FPC 400, PO Box 5, Filton, Bristol, BS12 7QW

INTRODUCTION

The story is often told of the tourist asking the country dweller how to get to some place or other. "If that's where you want to go," comes the reply, "I wouldn't start from here." This answer might be wiser than it looks. In this paper, the point in question is a beampattern, described in one or two dimensions and at a specific frequency or, perhaps, over a band of frequencies. But how to get to that point depends very much on the starting position.

If the beampattern is specified simply by beamwidth and sidelobe level, and complete freedom is allowed in the design of the array then it is a simple matter of selecting a uniform linear or planar array of the appropriate dimensions and applying one of the many shading schemes in common use. Real life, however, is more complicated. Perhaps the aperture is restricted, or a portion of it is already occupied by something, or the array has to conform to a three dimensional surface. Perhaps an otherwise standard beampattern must be modified - by the introduction of nulls, say - or perhaps the array already exists, but its measured directivity is not as predicted and the excitation that produced the experimental result must be found.

In practice, problems such as these are usually tackled by trial and error or by computer optimisation techniques (which is still trial and error really) - methods which are time consuming, or costly, or both. This paper describes a method which uses least-squares approximation to find a set of complex weights that will produce a given beampattern from a given array. Success is not guaranteed - the desired directivity function may not be realisable with the specified array or the complex weighting scheme may not be compatible with a practical beamformer - but the method requires little effort to implement on a small computer and the results are easily checked.

THE METHOD

Consider the beamformer architecture shown in Figure 1. There are a number of transducers, arbitrarily located in space and the signal is split into a separate channel for each transducer. Each of these channels is further split into two channels, one of which is phase shifted by 90° so that the co-phase and quadrature components of the shading coefficients may be applied separately by the variable attenuators. The outputs from each co-phase/quadrature pair are then recombined and passed to the power amplifiers and transducers.

This structure allows the real and imaginary parts of the shading coefficients to be separated which is convenient for the mathematical formalism, but in practice the results may be expressed in terms of amplitude and phase, or equivalent time delay, and used

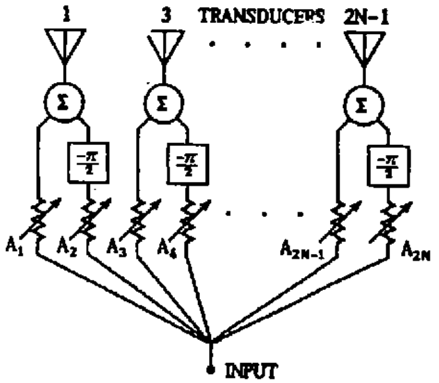


Fig.1 Beamformer architecture.

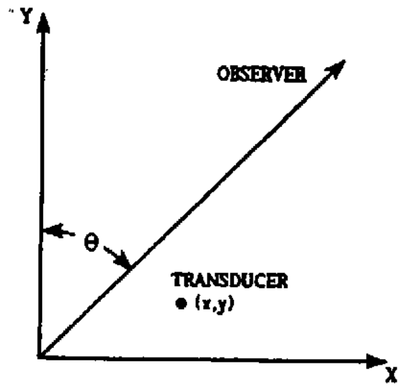


Fig.2 Coordinate system for computing array directional response.

with a more conventional beamformer. Note also that although a transmitting array is described here, the method may equally be applied to receiving arrays.

Figure 2 shows a transducer, with directivity function $G(\theta,\omega)$, located at an arbitrary point (x,y) in Cartesian coordinates. Although, for simplicity of notation, only the XY plane is considered here, the method is generally applicable to 3-dimensional space. The sound pressure $P(\theta)$ due to a signal of frequency ω emanating from this transducer, observed in the far field in direction θ and normalised relative to an imaginary reference element at the origin is

$$P(\theta) = G(\theta,\omega)\exp\{i(\omega t + u)\} \tag{1}$$

which may be expanded to give

$$P(\theta) = G(\theta,\omega)\exp(i\omega t)(\cos u + i\sin u) \tag{2}$$

and

$$u = \omega(x\sin\theta + y\cos\theta)/c \tag{3}$$

The $\exp(i\omega t)$ time dependence may be dropped, and the output from the array of N transducers determined by representing each transducer as a pair of identical elements, located at the same point in space, one driven by the co-phase channel and the other by the quadrature channel. If the weighting coefficients are $A_1, A_2, \ldots, A_{2N}$, the total output is

$$P_{tot} = \sum_{n=1}^{2N} G_n(\theta,\omega)A_n(\cos u + i\sin u) \tag{4}$$

Transducer position coordinates and directivity functions are duplicated for each pair of odd and even n's to correspond with the beamformer architecture, so for even n

$$G_n(\theta,\omega) = G_{n-1}(\theta,\omega) \tag{5}$$

and

$$u_n(\theta,\omega) = \begin{cases} k(x_n\sin\theta + y_n\cos\theta) & ;\ n \text{ odd} \\ u_{n-1} - \pi/2 & ;\ n \text{ even} \end{cases} \tag{6}$$

where k is the wavenumber, ω/c.

The array output may be separated into its real and imaginary parts and each equated to the real and imaginary parts of the desired beampattern, which must be known for M combinations of θ and ω. A system of linear simultaneous equations results which may be solved for the shading coefficients A_n if $M = N$ precisely, but generally, of course, M and N are different. The set of linear equations is

$$\begin{aligned} A_1 v_{1,1} + A_2 v_{1,2} + \cdots + A_{2N} v_{1,2N} &= w_1 \\ A_1 v_{2,1} + A_2 v_{2,2} + \cdots + A_{2N} v_{2,2N} &= w_1 \\ &\vdots \\ A_1 v_{2M,1} + A_2 v_{2M,2} + \cdots + A_{2N} v_{2M,2N} &= w_1 \end{aligned} \tag{7}$$

$$v_{m,n} = \begin{cases} G_n(\theta_m,\omega_m)\cos u_n(\theta_m,\omega_m) & ;\ m \text{ odd} \\ G_n(\theta_m,\omega_m)\sin u_n(\theta_m,\omega_m) & ;\ m \text{ even} \end{cases} \tag{8}$$

$$w_m = \begin{cases} Re\{D(\theta_m,\omega_m)\} & ;\ m \text{ odd} \\ Im\{D(\theta_m,\omega_m)\} & ;\ m \text{ even} \end{cases} \tag{9}$$

where, as with the G_n, the combinations of θ and ω have been duplicated, and for even m $\theta_m = \theta_{m-1}$ and $\omega_m = \omega_{m-1}$.

The best approximation to the shading coefficients A_n, in the least squares sense, may be found by defining a set of vectors, $\mathbf{V}_n$, formed from the simultaneous equation matrix [1]:

$$\begin{aligned} \mathbf{V}_1 &= (v_{1,1}\ ,\ v_{2,1}\ ,\ \cdots\ ,\ v_{2M,1}) \\ \mathbf{V}_2 &= (v_{1,2}\ ,\ v_{2,2}\ ,\ \cdots\ ,\ v_{2M,2}) \\ &\vdots \\ \mathbf{V}_{2N} &= (v_{1,2N},\ v_{2,2N},\ \cdots\ ,\ v_{2M,2N}) \end{aligned} \tag{10}$$

$$\mathbf{W} = (w_1\ ,\ w_2\ ,\ \cdots\ ,\ w_{2M})$$

the shading coefficients are determined from the orthogonality relationships [1]

$$\{(A_1\mathbf{V}_1 + A_2\mathbf{V}_2 + \cdots + A_{2N}\mathbf{V}_{2N}) - \mathbf{W}\}\cdot\mathbf{V}_n = 0 \qquad (11)$$

which yield a system of 2N equations

$$\begin{aligned}(\mathbf{V}_1 \cdot \mathbf{V}_1)A_1 + (\mathbf{V}_1 \cdot \mathbf{V}_2)A_2 + \cdots + (\mathbf{V}_1 \cdot \mathbf{V}_{2N})A_{2N} &= \mathbf{V}_1 \cdot \mathbf{W}\\ (\mathbf{V}_2 \cdot \mathbf{V}_1)A_1 + (\mathbf{V}_2 \cdot \mathbf{V}_2)A_2 + \cdots + (\mathbf{V}_2 \cdot \mathbf{V}_{2N})A_{2N} &= \mathbf{V}_2 \cdot \mathbf{W}\\ &\vdots\\ (\mathbf{V}_{2N} \cdot \mathbf{V}_1)A_1 + (\mathbf{V}_{2N} \cdot \mathbf{V}_2)A_2 + \cdots + (\mathbf{V}_{2N} \cdot \mathbf{V}_{2N})A_{2N} &= \mathbf{V}_{2N} \cdot \mathbf{W}\end{aligned} \qquad (12)$$

in the $2N$ unknowns. These are the normal equations for the approximation and may be solved, for example, by straightforward elimination or else by using the simultaneous equation solving/matrix inversion routines available with most computers.

SOME EXAMPLES

A simple 10 element line array with half-wave spacing is sketched in Figure 3A. Chebychev shading for -30dB (as listed in Table 1) results in the beampattern shown in Figure 4.

Given the array and the beampattern, but knowing nothing about Chebychev shading, suppose that it is necessary to reproduce the beampattern. If the array geometry and a random selection of 15 points from the beampattern are fed to the least-squares algorithm, a set of weighting coefficients (listed in Table 2, column A) are produced which result in the beampattern shown by the solid line in Figure 5A. The crosses represent the 15 random points..

The pattern obtained is a reasonable approximation to the requirements: it passes within a fraction of a dB of all the defining points, the beamwidth is right, and none of the sidelobes exceeds -30dB by more than 1dB. Furthermore, the magnitudes of all the estimated weighting coefficients are within 2% of the Chebychev weights from Table 1. This would all be quite acceptable in a practical system. This is a hypothetical case, however, not a practical system and it is noted, in particular, that the pattern lacks symmetry and that there are a lot of awkward little phase shifts in the required weighting coefficients.

These snags can be overcome simply by making the set of specifying points symmetrical about the 0° axis. The result of duplicating each of the random points, reflected about the 0° axis, is shown in Figure 5B, with the estimated weighting coefficients listed in

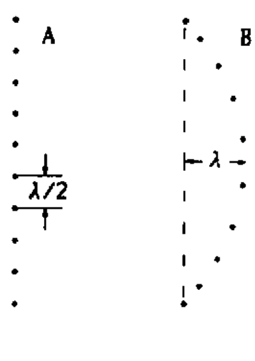


Fig. 3 _Sketch showing geometry of (A) line array and (B) curved array.

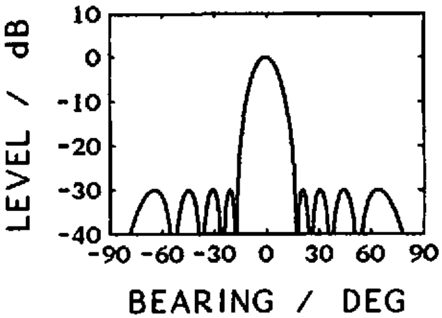
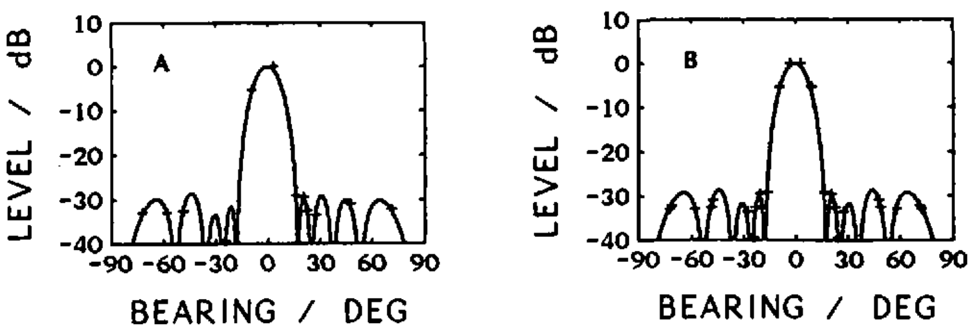


Fig. 4 Beampattern of 10 element line array with -30dB shading.

Table 1
-30dB Chebychev Shading Coefficients

Element Number	Weighting Coefficient
1	0.2575
2	0.4300
3	0.6692
4	0.8781
5	1.0000
6	1.0000
7	0.8781
8	0.6692
9	0.4300
10	0.2575



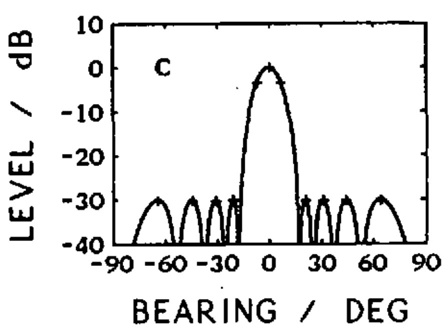


Fig 5

Beampatterns with estimated weighting coefficients: (A) random selection of data points from -30 dB Chebychev pattern, (B) symmetrical random data and (C) data defining main beam and sidelobes. Crosses show data.

Table 2.
Weighing coefficients to approximate the Chebychev -30dB beampattern.
A: 15 random points.
B: 30 random points, symmetrical about the 0° axis.
C: 11 points defining main beam and sidelobes.

Element Number	A		B		C	
	Magnitude	Phase/deg	Magnitude	Phase/deg	Magnitude	Phase/deg
1	0.2541	2.5549	0.2640	0.0	0.2575	0.0
2	0.4205	0.3970	0.4224	0.0	0.4301	0.0
3	0.6666	-0.8474	0.6749	0.0	0.6693	0.0
4	0.8846	0.1839	0.8935	0.0	0.8781	0.0
5	1.0000	0.3873	1.0000	0.0	1.0000	0.0
6	1.0000	-0.3873	1.0000	0.0	1.0000	0.0
7	0.8846	-0.1839	0.8935	0.0	0.8781	0.0
8	0.6666	0.8474	0.6749	0.0	0.6693	0.0
9	0.4205	-0.3970	0.4224	0.0	0.4301	0.0
10	0.2541	-2.5549	0.2640	0.0	0.2575	0.0

column B of Table 2. The pattern is now symmetrical, and the weighting coefficients are all real, but the desired uniform -30dB sidelobe level has not yet been achieved.

The final refinement is to forget the random selection of specifying points and, instead, to define the desired features in the beampatterns - in this case the peak and -3dB points on the main beam and the peaks of the sidelobes. Feeding these data to the least-squares algorithm produces the beampattern shown in Figure 5C and the weighting coefficients listed in Table 2 column C. Now, the beampattern exactly reproduces the original in figure 4, and the weights, to within 0.02%, are identical to the Chebychev coefficients.

The objective has been achieved, but some additional complications may now be introduced: The beampattern shown in Figure 4 has a null at -53°. Suppose it is required to move that null to -60°. The data used to generate Figure 5C were modified by replacing the point specifying the sidelobe at -64° with one defining a zero at -60°. The result is shown in Figure 6 (note that the vertical scale has

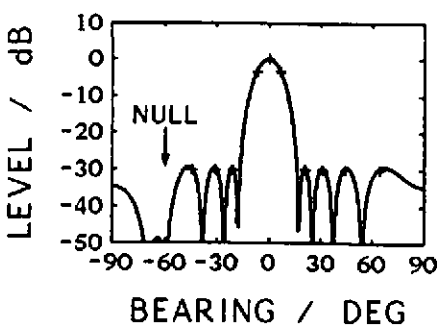


Fig 6 Beampattern using coefficients estimated from Chebychev -30dB pattern with null at 60°.

been expanded slightly), and the shading coefficients listed in Table 3. It cannot be denied that there is a deep null at -60°.

Table 3
Weighting for null at -60°

Element Number	Magnitude	Phase/deg
1	0.2403	0.7789
2	0.4440	-1.0376
3	0.6551	1.1171
4	0.8880	-1.0826
5	1.0000	1.0497
6	1.0000	-1.0497
7	0.8880	1.0826
8	0.6551	-1.1171
9	0.4440	1.0736
10	0.2403	-0.7789

Returning now to the beampattern data used to produce Figure 5C, consider a change of array shape as follows: each element of the line array shown in Figure 3A has been projected radially onto the circular arc that passes through the original end elements and deviates from the straight line by one wavelength at the mid-point. This is shown schematically in Figure 3B.

Anyone who has investigated curved arrays will know that this amount of curvature will completely wreck the beampattern, as demonstrated in Figure 7A. It is possible to introduce time delays to compensate for the deviation, but the resulting beampatterns are still inferior to those obtained from a uniform line array, and there are no standard shading schemes to control sidelobe levels. Feeding the data to the least-squares routine, however, gives the weights listed in Table 4 which generate the beampattern shown in Figure 7B. The result, once again, is to reproduce the original pattern from Figure 4.

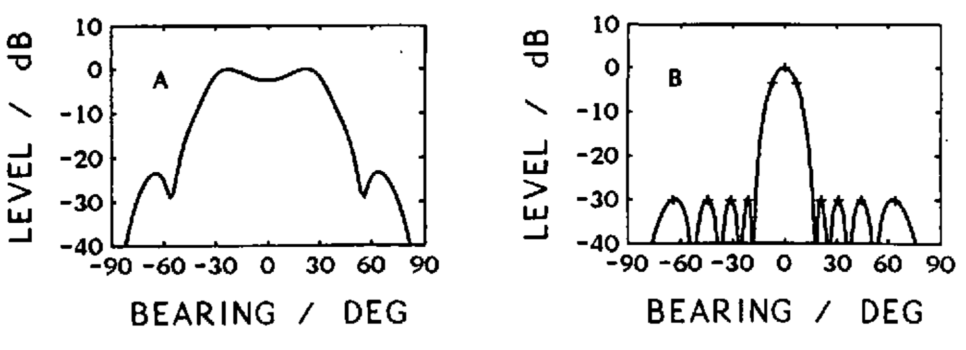


Fig 7 Beampatterns for curved array: (A) Chebychev -30dB coefficients for line array and (B) coefficients estimated from data defining main beam and sidelobes.

DISCUSSION AND CONCLUSIONS

The examples presented here show that it is generally possible, as the title suggests, to find shading coefficients to produce an arbitrary beampattern from a specified array. These examples, however, highlight a few points that should be born in mind when specifying the desired beampattern.

Table 4
Weighting for curved array

Element Number	Magnitude	Phase/deg
1	0.2673	-10.296
2	0.4272	77.818
3	0.6683	-20.958
4	0.8708	69.203
5	1.0000	13.576
6	1.0000	13.388
7	0.8708	69.258
8	0.6685	-20.945
9	0.4272	77.815
10	0.2673	-10.297

The first should be obvious. The algorithm does not know anything about your required beampattern but the points you supply. The curve that fits these points is not necessarily unique - the beampatterns shown in Figures 5A, 5B and 5C, for example, all fit the specifying data used to generate Figure 5A. To get the best results the data should identify the important features in the pattern such as the main beam, sidelobe levels, null positions and so on. And it should be remembered, when specifying sidelobes, that they are not necessarily in phase with the main beam.

The next point is that if symmetry is required in the beampattern, then the specifying data should also be symmetrical about the same point/axis.

Another matter related to the question of symmetry is the choice of the origin of coordinates used to define the array element locations. If there is an axis of symmetry the origin must lie on this axis or the results will be meaningless. If there is no symmetry the origin must lie at the acoustic centre of the array, but how that is found is another problem!

Finally, there has not been space in this short paper to delve into matters such as the directivity and frequency response of the transducers making up the array, how to deal with steered beams or how to handle beampatterns specified over a range of frequencies. A glance at Equations (1) to (12), however, will confirm that these points are all accounted for in the algorithm, or could be incorporated with a little ingenuity.

REFERENCES

1. D.C. Kreider, R.G. Kuller, D.R. Ostberg, and F.W. Perkins, 'An Introduction to Linear Analysis', Addison-wesley (1966).