AUTOMATED TRANSDUCER ARRAY OPTIMISATION

P F Dobbins (1) and G J Heald (2)

(1) BAeSEMA, Marine Division, PO Box 5, Filton, Bristol BS12 7QW
(2) DRA Bincleaves, Weymouth, Dorset DT4 8UR

1. INTRODUCTION

Optimising the parameters that control transducer array beampatterns grows in difficulty with the complexity of the array and the incompatibility of the directivity requirements. The easiest cases, such as the uniform line array require only simple formulae, but more perverse situations, conformal arrays for example, call for computer intensive optimisation routines. An alternative is to use some approximation technique to find parameters that best fit the beampattern equations to the requirements as defined at a number of discrete points in terms of angular bearing and array response. Such an approach, using a linear least-squares method, has previously been demonstrated by the present authors in a manual form for line arrays [1], plane arrays [2,3] and fully three-dimensional arrays [4].

This procedure is straightforward and for simple arrays can produce results at the first attempt, perhaps requiring a second pass to refine the solution. For large planar and 3D arrays with complex directivity requirements that include steered nulls and constraints on beamwidth or DI the process rapidly becomes tedious and time-consuming. The bulk of this effort, however, is consumed in examining the beampattern produced and comparing this with the requirements in order to define a new set of points that might achieve a better result. These are all tasks that are easily carried out by a computer, and in this paper an automated algorithm is presented that is capable of quickly and efficiently locating the best values for (complex) shading coefficients given a possibly conflicting specification including beamwidths, sidelobe levels and null directions.

The formulation includes element directivities, frequency responses, mutual coupling and other complicating factors encountered in practice, so direct application to the design of sonar arrays can be undertaken with confidence. Nevertheless, the examples presented here are restricted to the ubiquitous uniformly spaced line of omnidirectional elements. This is simply so that results may be presented in an easily assimilable form that clearly demonstrates the power and reliability of the method.

2. THE METHOD

The theory and mathematical formulation of the least-squares array shading algorithm were explained at length in [1] for line arrays and for completely general 3-dimensional geometries in [4]. The application of the method to particular array design problems, however, has not been

AUTOMATED TRANSDUCER ARRAY OPTIMISATION

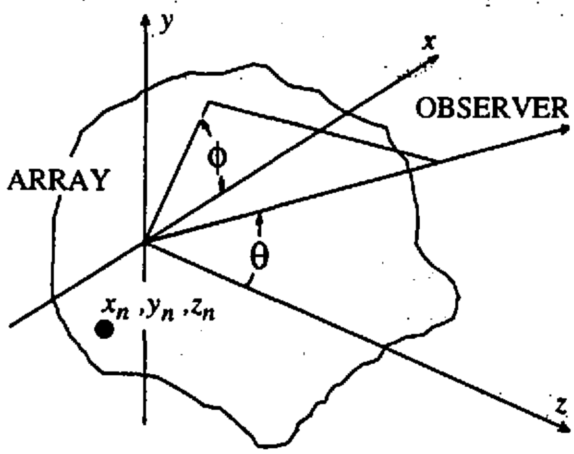

Fig.1 Geometry and coordinate system.

discussed in any detail. An explanation of the manual procedure will be given here and then it should be obvious how this sequence of events can be formalised and coded as a computer program.

The general coordinate system to be considered is sketched in Figure 1. Generally, the array is contained within the 3-dimensional 'blob', but for the line array examples, the transducer elements lie along the x axis, and the beampattern is plotted in the xz plane as function of bearing θ, with ϕ being set to zero.

The examples presented in earlier papers [1-4] demonstrate the application of the algorithm to finding coefficients that reproduce a given beampattern, and this may be considered a somewhat artificial situation. More common is the requirement to find shading that will produce a pattern defined in terms of a basic parameter like sidelobe level. The procedure adopted then is straightforward; the first step is to specify the required beampattern, and for this example this is simply a statement of desired sidelobe levels.

The second step is to identify a feasible solution as a point of departure; the algorithm cannot find coefficients for a beampattern that is not realisable with the given array geometry. If shading for a pattern close to the requirement is known then this can be used to initiate the process, otherwise uniform shading is as good a starting point as any. The bearings of the sidelobes in this pattern must be located, and their polarity noted, so that a required pattern definition can be prepared to apply the required level at these locations, along with unit amplitude in the main beam direction.

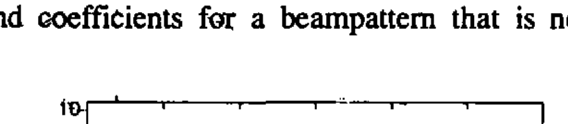

Fig.2 Line array beampatterns for unshaded array (dash-dot), first (dashes), second (dots) and final (solid line) manual algorithm iterations.

The least-squares approximation is next applied using this definition to obtain a new set of coefficients and a corresponding beampattern. This new pattern will be closer to the requirement than the original, but sidelobes may move as their level is

Table 1. Sidelobe locations and levels for successive algorithm iterations.

Uniform		1st Iteration		2nd Iteration		Final Shading	
θ / deg	Level / dB	θ / deg	Level / dB	θ / deg	Level / dB	θ / deg	Level / dB
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
±16.7	-13.0	±21.0	-22.3	±23.7	-33.3	±24.4	-40.0
±29.6	-16.9	±33.9	-30.9	±33.8	-40.0	±33.3	-40.0
±44.2	-19.0	±47.9	-36.5	±46.4	-39.6	±46.2	-40.0
±64.0	-19.9	±66.2	-39.6	±65.1	-40.0	±65.0	-40.0

Table 2. Shading coefficients for successive algorithm iterations and Chebychev -40dB shading.

Element	Uniform	1st Iteration	2nd Iteration	Final	Chebychev
1/10	1.0000	0.2236	0.1376	0.1253	0.1253
2/9	1.0000	0.5174	0.3444	0.3154	0.3154
3/8	1.0000	0.7569	0.6130	0.5802	0.5802
4/7	1.0000	0.9187	0.8577	0.8390	0.8390
5/6	1.0000	1.0000	1.0000	1.0000	1.0000

lowered, and the result may not be exact. In this case the new sidelobe positions should be located and the process repeated until the desired pattern is achieved.

Figure 2 demonstrates application of this procedure to a 10-element line array with $\lambda/2$ spacing and the objective of a uniform -40dB sidelobe level. This can be achieved to within ±0.1dB in three iterations. The initial unshaded pattern is shown as the dash-dot line, with the first and second iterations as dashed and dotted lines respectively and the final result as the solid line. The sidelobe locations and levels in dB for each stage of the procedure are listed in Table 1, along with the shading coefficients obtained at each stage in Table 2. For comparison, the coefficients for -40dB Chebychev shading are also given in the table, and it will be seen that these are identical to the values obtained with the least-squares algorithm to at least four decimal places.

This simple procedure may be formalised as the block diagram in Figure 3. The functions of most of the blocks are obvious but the three critical elements, those labelled 'Locate Sidelobes, Nulls, etc.', 'Meets Requirements?', and 'Determine Pattern Definition', require further comment. This is because the process as described above is somewhat brutal and liable to become

AUTOMATED TRANSDUCER ARRAY OPTIMISATION

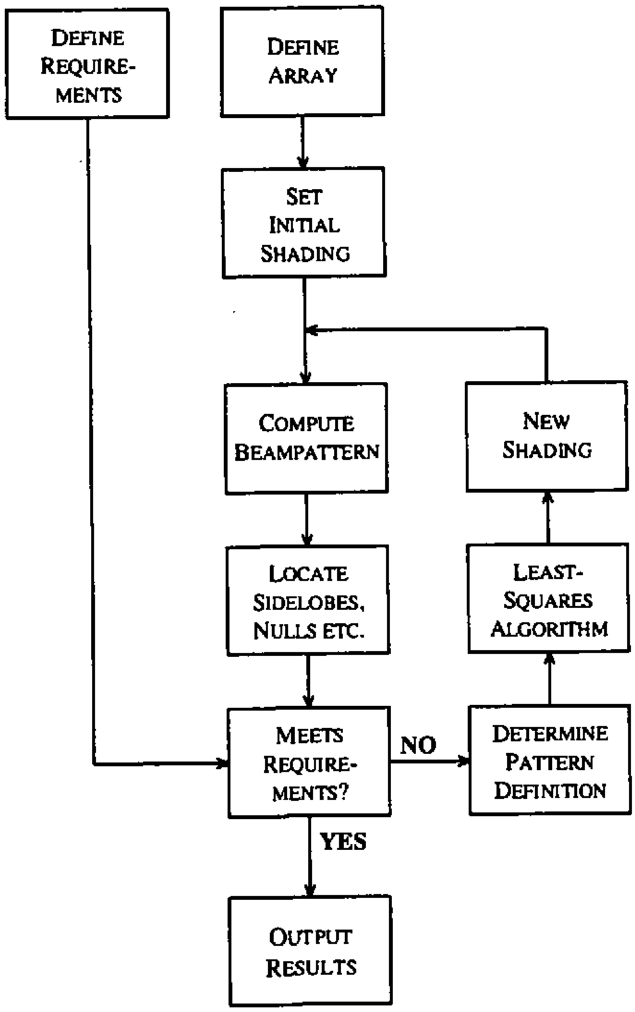

Fig.4 Formalised shading procedure block diagram.

unstable, leading either to completely meaningless results or a closed loop where the pattern found by the least-squares algorithm hops between two stable states.

A human operator, especially one with some understanding of the behaviour of arrays and beamformers, can intuitively influence each stage of the iteration by choosing new pattern definitions that are likely to produce sensible results. A computer program, of course, is not capable of such subtlety and the actions of each of the three critical blocks must be carefully defined to avoid unstable conditions. The most important problematic situations to be accounted for may be summarised as follows:

a) Sidelobes spread out in space as they are forced down in level. In particular, the iteration process may be 'confused' if the change is so great that the outer sidelobes move outside the visible angular region and the total number of sidelobes changes. This problem is easily overcome by taking small steps and reducing sidelobes by just a few dB at each stage rather than aiming at the target requirement immediately.

b) Diffraction secondaries and periodicity in the pattern due to regular element spacing cannot be controlled by shading and any attempt to do so will lead to instability. This is overcome by calculating the period along all lines of symmetry in angular space and restricting the pattern definition to locations within one half-period of the main beam direction.

c) Meaningless results will be produced if the pattern definition includes conflicting requirements. If a requirement for a null, or other specific level in a particular direction, is close to a sidelobe, even a sidelobe at the same bearing in θ along a different line of

AUTOMATED TRANSDUCER ARRAY OPTIMISATION

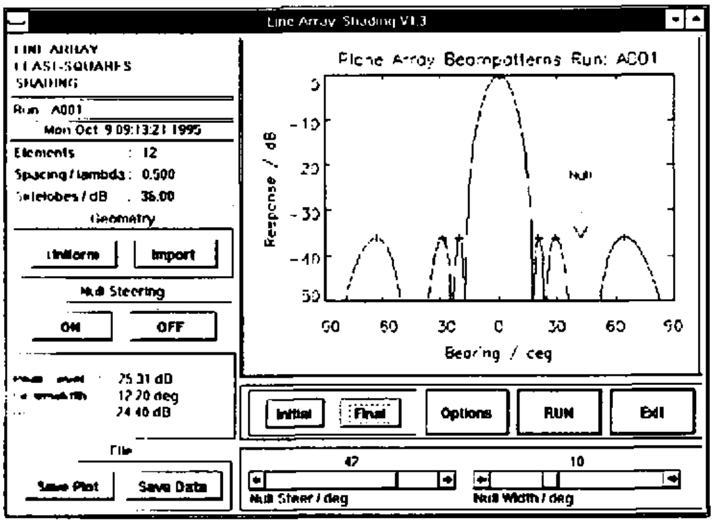

Fig.4 GUI implementation in IDL.

symmetry in ϕ, the sidelobe should be dropped from the pattern definition. This allows the null or specified level to take precedence.

d) Exact matching of the computed beampattern with the requirement is impossible due to rounding errors and other practical imperfections, and if a reasonable tolerance is not allowed in deciding that convergence is complete it never will be achieved.

e) The accuracy with which the angular locations of sidelobes, nulls and other points of interest are estimated must be small compared with the width of the narrowest sidelobe in the pattern and with the change occurring between stages in the iteration. If not, convergence will be impossible because the algorithm will not be able to see the variation from one step to the next and will wander aimlessly.

These points may be programmed as logical decisions, and once incorporated the algorithm is easily implemented in almost any programming language. Figure 4 shows a graphical user interface (GUI) developed in *IDL*, although such sophistication is not necessary for successful employment of the method.

3. SOME EXAMPLES

A few unpretentious examples will serve to demonstrate the method. Each of these uses as its basis a uniformly spaced line array of twelve omnidirectional elements. The objectives are firstly to find a set of shading coefficients that give a uniform sidelobe level for the uncomplicated case with $\lambda/2$ spacing and then for a spacing of λ, where diffraction secondaries and periodicity in the pattern could present difficulties. The required sidelobe level was arbitrarily chosen to be -36dB relative to the main beam. Having achieved this, nulls steered in specific directions will be introduced, firstly at a bearing coinciding with an existing sidelobe and then in a direction where there is already a null but aiming to broaden this null to a width comparable with the sidelobes.

AUTOMATED TRANSDUCER ARRAY OPTIMISATION

Table 3. Shading coefficients for example algorithm applications and Chebychev -36dB shading.

Element	Fig.5	Fig.6	Fig.7	Fig.8	Chebychev
1/12	0.1590	0.1590	0.1297	0.0968	0.1590
2/11	0.2990	0.2990	0.2885	0.2440	0.2990
3/10	0.5034	0.5034	0.4995	0.4821	0.5034
4/9	0.7179	0.7179	0.6946	0.6721	0.7179
5/8	0.8974	0.8975	0.9144	0.8835	0.8975
6/7	1.0000	1.0000	1.0000	1.0000	1.0000

The results obtained are shown in Figures 5-8. Each of these figures shows (A) the uniformly shaded beampattern, with crosses denoting the locations and levels of the sidelobes found at the first pass through the algorithm, (B) the sequence of beampatterns obtained with the sidelobe levels reduced by 2dB at each iteration and (C) the final beampattern with crosses denoting the locations and levels used to form a pattern definition at the last pass of the algorithm. The shading coefficients used to produce the final pattern in each example are listed in Table 3, along with the coefficients for -36dB Chebychev shading.

The unshaded pattern has a beamwidth of 8.5°, a DI of 10.8dB and a sensitivity of 21.6dB relative to a single transducer. In the first example, Figure 5, the sidelobes reduce smoothly in 2dB steps and converge in 13 iterations to the required pattern with -36dB sidelobes, a beamwidth of 11.6°, a DI of 9.8dB and a sensitivity of 17.1dB. No constraint was put on the symmetry of the beampattern, so it is symmetrical about the main beam and, as can be seen in Table 3, the shading coefficients are real and symmetrical about the array centre. Furthermore, these coefficients are within ±0.04% of the Chebychev coefficients for a -36dB sidelobe level.

Figure 6 demonstrates that by restricting the control range to one period of the beampattern, as indicated by the crosses, problems with periodicity and diffraction secondaries are avoided. In this case the beamwidth starts at 4.2° and finishes at 5.8°, the diffraction lobes are ignored, and this time the resulting coefficients are identical to Chebychev shading to 4 decimal places. The DI's and sensitivities are, of course, the same as in the previous example.

It will be seen in Figure 5 that there is a sidelobe in the final pattern at a bearing of 49°. Figure 7 shows the result of including a requirement for a null at this location. The sidelobes reduce slightly less smoothly than in the previous examples, but still converge in 13 iterations to a uniform -36dB level with a null in the required position. The final pattern has a beamwidth of 11.8°, a DI of 9.7dB and a sensitivity of 17.0dB, so is essentially the same as that in Figure 5C.

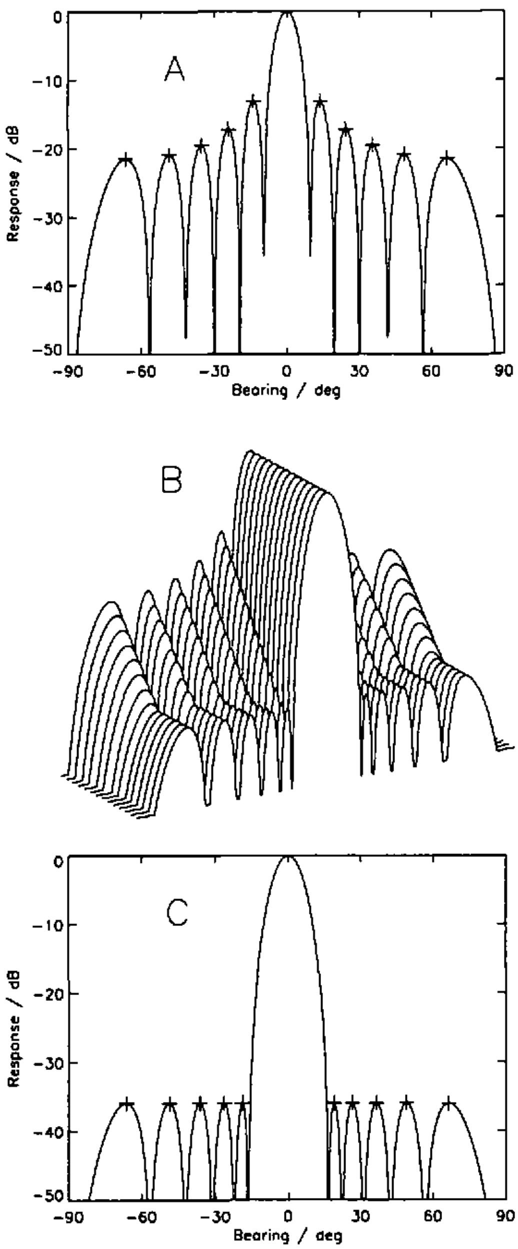

Fig.5 Initial pattern (A), iteration sequence (B) and final pattern (C) for 12 element array with λ/2 spacing and objective of uniform -36dB sidelobes.

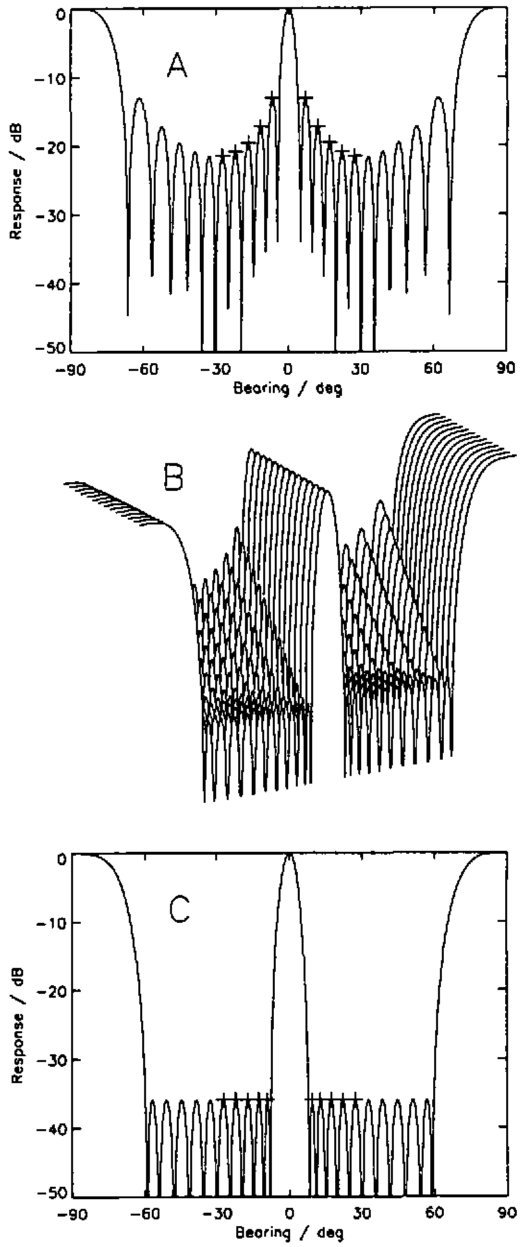

Fig.6 Initial pattern (A), iteration sequence (B) and final pattern (C) for 12 element array with λ spacing and objective of uniform -36dB sidelobes.

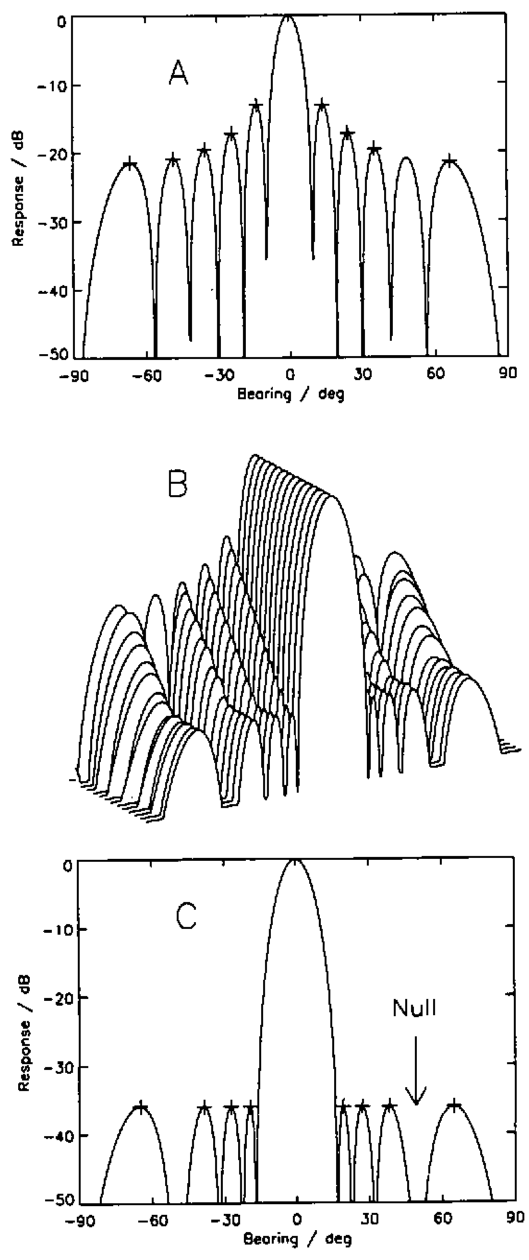

Fig.7 Initial pattern (A), iteration sequence (B) and final pattern (C) for 12 element array with λ/2 spacing and objective of -36dB sidelobes and null at 49°.

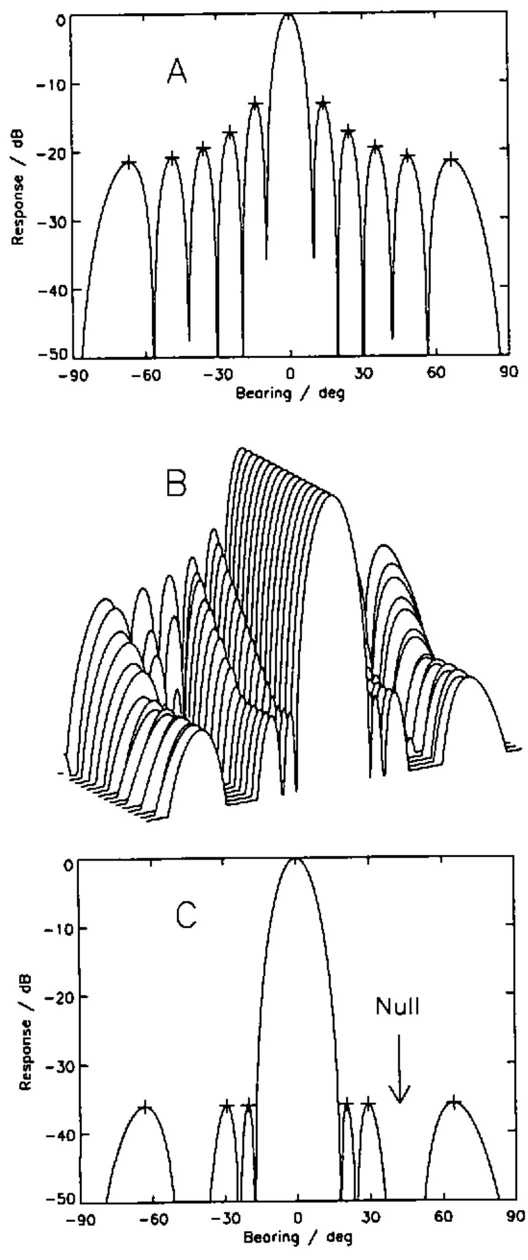

Fig.8 Initial pattern (A), iteration sequence (B) and final pattern (C) for 12 element array with λ/2 spacing and objective of -36dB sidelobes and broad null at 42°.

It is also noted that all the coefficients are within ±3.5% of Chebychev shading, except for the outer two, which differ by 18%, so errors and tolerances in a practical array and beamformer must be smaller than this to form such a null.

Finally, referring again to the pattern in Figure 5C, it will be seen that there is a null located at 42°. Obviously, a requirement for a zero at this bearing would have no effect on the final pattern. The null can be broadened, however, by specifying a group of nulls spaced apart by angles small compared with the width of a sidelobe in this region. For the example in Figure 8 a group of five nulls was specified at 42°, 42°±2° and 42°±4°. This time it takes 15 iterations to converge, but the result is again a pattern with a uniform -36dB sidelobe level and a broad null in the desired position. The final pattern has a beamwidth of 12.2°, a DI of 9.6dB and a sensitivity of 16.6dB so, once more, is not significantly degraded relative to the pattern in Figure 5C, and the remarks above relating to the accuracy of the shading coefficients are equally applicable.

4. In Conclusion

This paper has described a method for finding shading coefficients that can be applied to arrays of arbitrary geometry to achieve any physically realisable set of requirements for the resulting beampattern. The method is fully automatic in the sense that if the decisions and parameters discussed in Section 2 are properly programmed no intervention by the user is required. Nor is any skill or knowledge required of the user beyond defining a reasonable specification for the beampattern before running the program.

The algorithm is easily implemented as a computer program, either in conventional programming languages or the more recent higher level languages such as *Mathcad*, *Mathematica* and *IDL*. The examples were obtained with an *IDL* GUI based implementation, but the algorithm has previously been programmed in both BASIC and *Mathcad*. Compared with earlier numerical optimisation approaches (see [2] or [3] for a brief review) this method is not computer intensive and arrays of several hundreds of elements are easily handled by a standard PC (Figures 5-8 each took about 20 seconds to produce on a 66Mhz 486 DX2 machine, including saving the graphics as Postscript files).

The examples given here do not represent any specific beamforming application, and nor do they show the most complex or most difficult situations that can be handled by the method. They are simply intended to give an idea the sort of problems that can be tackled using the least-squares technique, and to demonstrate that reasonable solutions can be obtained quickly and easily. However, the procedure does not always find a useful answer, or indeed any answer, if the requirements are not physically realisable with the specified array geometry. The algorithm cannot, for example, suppress diffraction secondary lobes if the element spacing is too wide.

Nevertheless, like other CAD techniques, if it is used intelligently and with an understanding of the behaviour of arrays and beamforming, the method provides an efficient tool for finding the best possible shading for a particular array geometry given a set of beampattern requirements.

5. ACKNOWLEDGEMENTS

This work was carried out with the support of the Defence Research Agency.

6. REFERENCES

[1] P F Dobbins, 'How to Find Shading Coefficients that Produce an Arbitrary Beampattern from an Arbitrary Array', *Proc. Inst. Acoust.*, 13(2), 243-250 (1991).

[2] P F Dobbins and G J Heald, 'General Radiation Pattern Synthesis Technique for Sonar Transducer Arrays', *Proc. UDT '94*, Nexus Business Communications 206-210, (1994).

[3] P.F. Dobbins and G.J. Heald, 'A New Beampattern Design Method for Sonar Arrays', *Acoustics Bulletin*, 19(5), 13-17 (1994).

[4] P.F. Dobbins and G.J. Heald, 'General Shading Technique for Sonar Transducer Arrays', *Proc. Inst. Acoust.*, 17(3), 117-126 (1995).