Phase-only Beam Broadening in Large Transmit Arrays Using Complex-Weight Gradient Descent

Jeffrey O. Coleman W. Mark Dorsey
jeffc@alum.mit.edu wmdorsey@vt.edu

Naval Research Laboratory
Radar Division
Washington, DC

***Abstract*—Day recently obtained phase-only element weights for a large transmit array of arbitrary geometry by optimizing complex weights using an objective function that penalized amplitude variation. He used gradient descent modified with a classic SVD technique to create point and sector nulls as desired.**

Here Day's approach is extended to allow for main-beam broadening as might be needed to illuminate multiple receive beams. This is done by adding a second objective term, one penalizing variation in array-factor amplitude across a grid of beamspace points covering the desired broad main beam. The tension between the two objective terms turns out to be challenging to manage, and at this preliminary stage of algorithm development, parameter scheduling across a sequence of stages of the optimization is required to obtain good solutions.

I. Introduction

This paper takes Day's approach [1], [2] to phase-only array weights and expands on it to broaden the main beam. To see how much harder a problem this is, consider Fig. 1, which shows array factors for a circular transmit array of radius 10λ with its 1075 elements laid out on a hexagonal lattice—equilateral triangular grid—with a center-to-center spacing of $\lambda/\sqrt{3}$, the classic maximum for remaining free of grating lobes at all steering angles.

The simplest phase-only weights are uniform weights and, of course, yield a boresight pencil beam. The upper plot in Fig. 1 shows the associated array factor steered slightly off boresight for display purposes only, this to spread the uniform element weights out along a circle in the complex plane. For compactness of presentation in the plot, a complex-plane scatter plot of those steered weights is overlaid on the array factor, with that circle sized, purely for convenience, to overlay it on the edge of the visible region. That these are indeed phase-only weights is immediately apparent visually.

So far the design problem is trivial, but Day changed that by assuming a need to suppress the array factor in some region, for example in a band along the horizon. The middle plot of Fig. 1 illustrates this by replacing the original weight vector with its projection onto a subspace in which array-factor magnitude is everywhere quite small in a suppression band extending from 1° below the horizon to 3° above the horizon, assuming that the array face is tilted back to put boresight 25° above the horizontal. (A steering offset is still present

This work was supported by the Naval Research Laboratory base program.

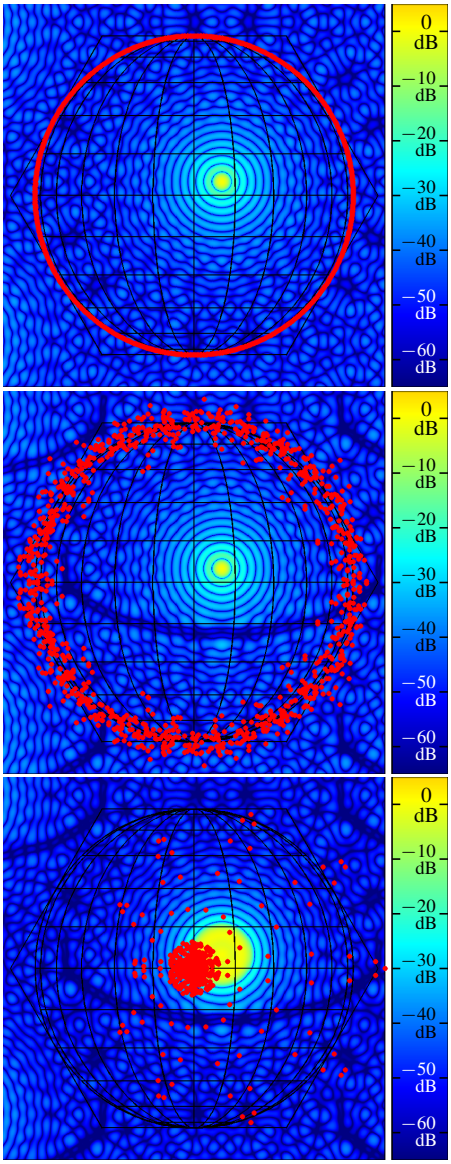


Fig. 1. Example array factors for uniform illumination (top), uniform illumination modified using an SVD-based projection to incorporate horizon nulling (center), and similarly modified lattice-sampled Airy illumination that creates a broad main beam (bottom). Here steering away from boresight is purely for display, to give the complex weights a variety of angles in their complex-plane scatterplots here superposed on the array factors.

in the plot to better display the weights.) A singular-value decomposition (SVD) approach is used to avoid restricting the dimensionality of the subspace further than necessary to obtain the null depth desired. The weights that were precisely on the circle in the top plot are now somewhat spread in magnitude.

Day optimized the weights from just such a starting point to push them back towards phase-only behavior. Specifically, he used a modified gradient descent in complex weight space to minimize the sample variance of the squared weight magnitudes and so gradually force them back onto the circle. The modification comprised stripping all weight-vector increments of components along the boresight direction and in the subspace zeroed to create the null region. This ensured that the pencil beam and the null region would be undisturbed by weight iteration. What was remarkable about this approach was that it could easily handle large arrays and do so without sophisticated optimization tools.

This paper draws inspiration from Day's approach to create a new algorithm for designing phase-only weights but now with a broadened main beam useful for simultaneously illumination of a cluster of narrower receive beams. Day's SVD-based projection approach is adopted, with minimal modification, to maintain a null region throughout the optimization. The major change is the addition of an objective-function term to discourage major variation in the magnitudes of samples of the main-beam array factor as well as among magnitudes of the weights themselves.

This algorithm presents two specific challenges. The first challenge is that a broadened main beam, even as a pre-iteration starting point, corresponds to weights not tending to cluster near a circle at all. This is illustrated in the bottom plot of Fig. 1. (The weight scatterplot is not visually much different if the null zone is removed.) The second challenge is that the relative weights given to the two objective-function terms turn out to be quite tricky to manage, and those weights, among other things, must be tweaked during the optimization using a hand-tuned schedule, at least for now.

Indeed, this need for hand tuning correctly suggests that this algorithm must at present be regarded as preliminary, as a base for further development. If this need for hand tuning can somehow be overcome, the approach will allow the creation of broad beams in large arrays using a relatively simple matlab iteration and without requiring any general-purpose optimization package.

II. The Algorithm

We assume each element-center position vector takes the form of column vector $\mathbf{p} = \mathbf{B}\boldsymbol{m}$, where the number of length-dimensioned basis-vector columns of matrix $\mathbf{B}$, one for a line array or two for a planar array, is also the length of the index column vector $\boldsymbol{m}$ of integers. We suppose the elements actually driven are those with $\boldsymbol{m} \in \mathcal{M}$, some finite set. Guard or dummy elements are presumably present as well so as to put the driven elements in an essentially periodic structure as far as electromagnetics are concerned.

A. Initialization

Just as Day did, we choose initial complex weights to meet all desired criteria except the phase-only requirement. The specifics are application dependent, but typically a least-squares or convex-programming solution can provide the initial weights. Here we instead develop closed-form initial weights for the special case of illuminating a circular region in beamspace while nulling a horizon zone.

1) Illuminating a Circular Beamspace Region: Use a Bessel function of the first kind of order one to define

$$\operatorname{jinc}(x) \triangleq \begin{cases} J_1(x)/x & \text{for } x \neq 0, \\ \lim_{x \downarrow 0} J_1(x)/x = \frac{1}{2} & \text{for } x = 0 \end{cases}$$

and given any set $\mathcal{A}$, let indicator function $1_{\mathcal{A}}(x)$ take value 1 when $x \in \mathcal{A}$ and 0 otherwise. A 2D Fourier pair from optics,

$$1_{[0,1]}(\|\boldsymbol{p}\|) \leftrightarrow 2\pi \operatorname{jinc}(2\pi\|\boldsymbol{k}\|)$$

relates the indicator on the closed unit disc in normalized (dimensionless) 2D position vector $\boldsymbol{p}$ on the left to the Airy distribution in normalized 2D spatial-frequency vector $\boldsymbol{k}$ on the right [3]. Simple Fourier properties then yield the inverse transform from physical spatial frequency $\mathbf{k} \triangleq k_r \boldsymbol{k}$ to physical position $\mathbf{p} \triangleq k_r^{-1} \boldsymbol{p}$ of the unit-height disc of radius k_r in $\mathbf{k}$:

$$2\pi k_r^2 \operatorname{jinc}(2\pi k_r \|\mathbf{p}\|) \leftrightarrow 1_{[0,k_r]}(\|\mathbf{k}\|).$$

If we take basis matrix $\mathbf{B}$ to be square for convenience (it could have been made taller—for 3D positions perhaps—than it is wide at some cost in mathematical convenience), lattice sampling onto element centers yields

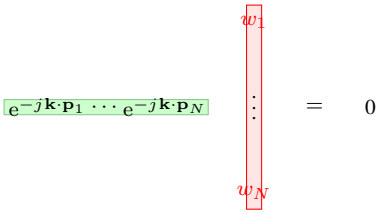
$$|\mathbf{B}^T\mathbf{B}|^{1/2} \sum_{\boldsymbol{m} \in \mathbb{Z}^2} 2\pi k_r^2 \operatorname{jinc}(2\pi k_r \|\mathbf{B}\boldsymbol{m}\|)\, \delta(\mathbf{p} - \mathbf{B}\boldsymbol{m}) \leftrightarrow \sum_{\boldsymbol{\ell} \in \mathbb{Z}^2} 1_{[0,k_r]}(\|\mathbf{k} - \mathbf{B}^{-1}\boldsymbol{\ell}\|). \quad (1)$$

Here $|\cdot|$ denotes the absolute value of the determinant, and $\mathbf{k}$ is taken to be a row vector. In the infinite distribution then, the array weight at position $\mathbf{B}\boldsymbol{m}$ is just $2\pi k_r^2 |\mathbf{B}^T\mathbf{B}|^{1/2} \operatorname{jinc}(2\pi k_r \|\mathbf{B}\boldsymbol{m}\|)$ for any $\boldsymbol{m} \in \mathbb{Z}^2$.

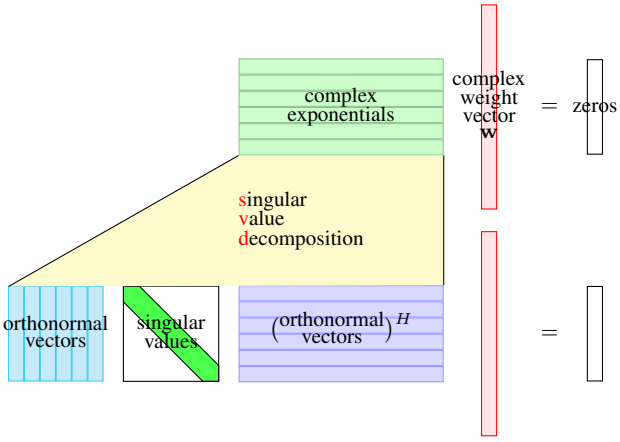
Given a realizable aperture defined by $\boldsymbol{m} \in \mathcal{M}$, the least-squares realizable approximation of the infinite distribution on $\boldsymbol{m}$ is the latter's orthogonal projection onto $\mathcal{M}$. Later we will want weight amplitude unity and therefore a total weight energy of N, so let us normalize this realizable approximation to that energy. The initial weight at $\mathbf{p} = \mathbf{B}\boldsymbol{m}$ then becomes

$$w_{\boldsymbol{m}} = \frac{\sqrt{N} \operatorname{jinc}(2\pi k_r \|\mathbf{B}\boldsymbol{m}\|)}{\operatorname{rms}\{ \operatorname{jinc}(2\pi k_r \|\mathbf{B}\boldsymbol{m}\|) : \boldsymbol{m} \in \mathcal{M} \}}.$$

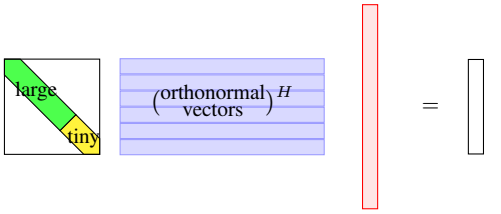
2) Forcing a Horizon Null Zone: We impose a sector null just as Day did, using a routine signal-processing approach sketched here only briefly. Fig. 2(a) shows that a beamspace null means weight vector $\mathbf{w} \triangleq [w_1, \ldots, w_N]^T$ is zero in a one-dimensional subspace. In the top of Fig. 2(b), a multiple-null requirement makes this subspace multi-dimensional. The rest of Fig. 2 uses singular-value decomposition to reduce the dimensionality of the nulled subspace at the cost of null



(a) An array-factor null at $\mathbf{k}$ in beamspace.



(b) Many array-factor nulls (top) re-expressed using SVD (bottom). The bottom equation is equivalent to (c) below.



(c) Tiny-singular-value rows assert $0 \approx 0$ and can be ignored. Other rows represent constraints enforced in (2)

Fig. 2. We could create an approximate sector null by fully nulling the array factor at a modest list of specific sector points, effectively zeroing the weight-vector component in a particular subspace. Instead we space null points much more closely and use SVD to express that full-nulling subspace as the span of the orthonormal vectors in (c). The initial weight vector (before nulling) has amplitudes along those orthonormal directions given by singular values in (b). Many of these are already small and can be ignored. The others flag directions along which the weight vector must be forcibly zeroed (see (2)) to effect the desired sector null.

depth. In particular, if $(\cdot)^H$ denotes the Hermitian transpose and if matrix $\mathbf{V}^H$ comprises those orthonormal row vectors in Fig. 2(c) that correspond to singular values large enough (empirically determined for now) to be of interest, nulling is forced by simply replacing $\mathbf{w}$ with

$$\mathbf{w}_{\text{nulled}} \triangleq \mathbf{w} - \mathbf{V}\mathbf{V}^H\mathbf{w} \tag{2}$$

to subtract from $\mathbf{w}$ any component it had in the subspace.

Note that writing the right side as $(\mathbf{I} - \mathbf{V}\mathbf{V}^H)\mathbf{w}$ and pre-computing projection matrix $\mathbf{I} - \mathbf{V}\mathbf{V}^H$ is inefficient for large arrays when $\mathbf{V}$ has far fewer columns than rows, as is the usual case. Instead the term $\mathbf{V}\mathbf{V}^H\mathbf{w}$ should be computed in a right-associative way as $\mathbf{V}(\mathbf{V}^H\mathbf{w})$ and subtracted from $\mathbf{w}$.

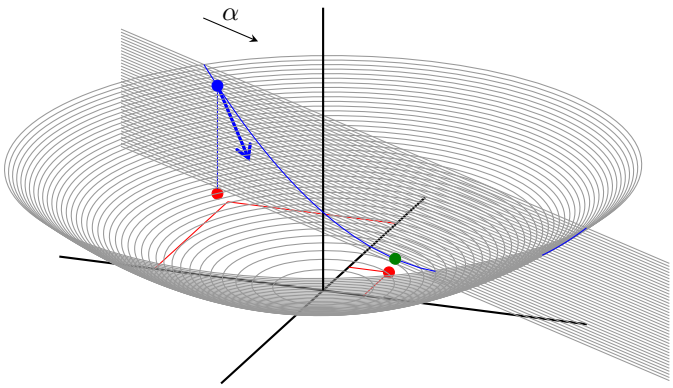


Fig. 3. The objective "bowl" is a 4th-degree multinomial in the real and imaginary components of the weights, here represented in two rather than its actual many dimensions by position in the horizontal plane. That bowl is sliced along the direction of the **negative gradient vector** at an initial **point** to create a 4th-degree polynomial in a position-in-slice parameter α. The one real root of the 3rd-degree derivative polynomial with respect to α locates a **minimum**, which becomes the next step's initial **point**. This simple scheme is modified, as discussed in the text, by replacing the gradient with a projected gradient, to maintain the array factor's null zone, and by incorporating a Polak-Ribiere conjugate-direction formulation to improve convergence speed.

B. Gradient Descent

Day's approach and ours both use the conventional gradient-descent approach sketched in Fig. 3 as modified, in a way more formulaic than interesting, to incorporate Polak-Ribiere conjugate gradients, strictly for convergence speedup. Day's approach and ours both work without that modification, though with slower convergence, so that modification is not discussed further even though it was incorporated into the computations of the Section III design example below.

1) Classic steepest descent: The classic steepest-descent approach is simple. To iterate weight vector $\mathbf{w}$ through values $\mathbf{w}_0, \mathbf{w}_1, \ldots$ that converge towards some $\mathbf{w} = \mathbf{w}^*$ minimizing an objective function, say $F(\mathbf{w})$, over all $\mathbf{w}$, first choose an initial weight vector $\mathbf{w}_0$ as discussed above. Then, since the gradient $\nabla F(\mathbf{w})$ of $F(\mathbf{w})$ with respect to vector $\mathbf{w}$ points "uphill," compute that gradient at point $\mathbf{w}_n$ and, using a free parameter $\alpha \geq 0$, set $\mathbf{w}_{n+1} = \mathbf{w}_n - \alpha\nabla F(\mathbf{w}_n)$ so that the next choice $\mathbf{w}_{n+1}$ of the vector is a function of α. Choosing the α corresponding to the $\mathbf{w}_{n+1}$ that minimizes $F(\mathbf{w}_{n+1})$ is now a one-dimensional minimization problem. This process is repeated until $\mathbf{w}_{n+1}$ no longer differs significantly from $\mathbf{w}_n$, at which point $\mathbf{w}^*$ is set to $\mathbf{w}_{n+1}$.

2) Modify to handle a complex vector: The process just discussed applies naturally to optimizing a real vector $\mathbf{w}$, but things are very little different if, as is the case here, $\mathbf{w}$ is complex so that $\mathbf{w} = \mathbf{x} + j\mathbf{y}$, where $\mathbf{x} \triangleq \left[x_1, \ldots, x_N\right]^T$ and $\mathbf{y} \triangleq \left[y_1, \ldots, y_N\right]^T$. One simply replaces $\mathbf{w}$ with

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}$$

everywhere, so that the iteration becomes

$$\begin{bmatrix}\mathbf{x}_{n+1}\\ \mathbf{y}_{n+1}\end{bmatrix} = \begin{bmatrix}\mathbf{x}_n\\ \mathbf{y}_n\end{bmatrix} - \alpha \begin{bmatrix}\nabla_{\mathbf{x}} F(\mathbf{w}_n)\\ \nabla_{\mathbf{y}} F(\mathbf{w}_n)\end{bmatrix},$$

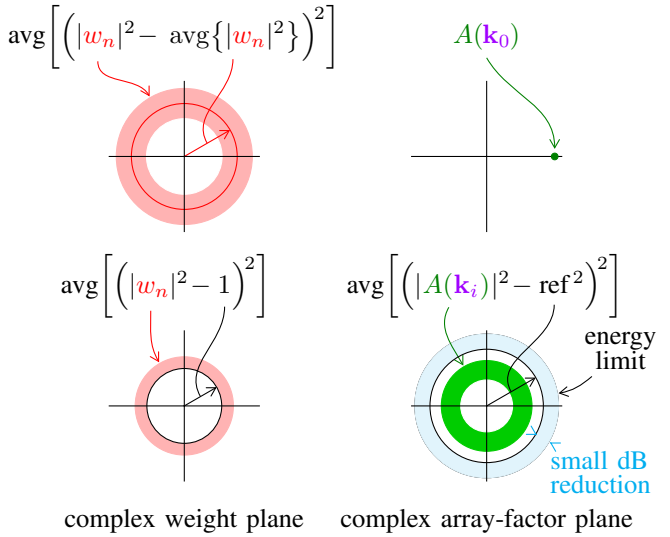


Fig. 4. Array weights w_n (left) and samples of mainbeam array factor $A(\mathbf{k})$ (right), sketched in the complex plane. Day (upper left) minimizes the sample variance of $|w_n|^2$ with $A(\mathbf{k}_0)$ fixed (upper right) for a single pointing-direction $\mathbf{k}_0$. Here we instead minimize a mixture of two terms, the mean over n of the squared error of $|w_n|^2$ relative to unity (lower left), and the mean over i of squared error of $|\mathbf{k}_i|$ relative to a squared reference level (lower right). That reference level is reduced a small number of dB from a value that would nominally put all of the energy into the main beam.

using gradients with respect to $\mathbf{x}$ and $\mathbf{y}$. It is equivalent to just define the gradient with respect to the complex vector $\mathbf{w}$ by

$$\nabla F(\mathbf{w}) \triangleq \nabla_{\mathbf{x}} F(\mathbf{w}) + j \nabla_{\mathbf{y}} F(\mathbf{w}) \tag{3}$$

and rewrite the iteration as

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \alpha \nabla F(\mathbf{w})$$

to put it in the same form used for a real vector.

3) Peculiarities of a 4th-degree nonnegative objective: Day's work and ours each minimize a nonnegative 4th-degree multinomial objective function in the real and imaginary components of complex weight vector $\mathbf{w}$. The function sketched (inaccurately) on the slice in Fig. 3 is a nonnegative quartic polynomial in α, and its derivative is a cubic polynomial in α, the roots of which are easily found. There are several cases.

a) A zero gradient: This implies $\alpha = 0$ and $\mathbf{w}_* = \mathbf{w}_n$. We assume we are at a local minimum. (However, a local maximum is actually possible with our objective.)

b) One real root: This is the desired α. The negative gradient points downhill, so $\alpha > 0$ is assured.

c) Three distinct real roots: This implies a double dip in the bowl slice. Small positive α values move downward along the slice curve, so choosing the smallest positive root guarantees a local minimum and is consistent with a philosophy of moving incrementally, without wild jumps. Alternatively, choose the root yielding the lowest objective value, i.e. take some jumps.

d) One simple real root and one double real root: This implies, given the nonnegative quartic objective, that the single root yields the lower objective value and is so the right choice.

4) A Chain Rule and a Gradient Derivation: A particular chain rule makes it relatively simple to obtain the key gradients below. The rule is, using $\text{len}(\mathbf{u})$ for the length of vector $\mathbf{u}$,

$$\nabla S(\mathbf{u}(\mathbf{w})) = \sum_{k=1}^{\text{len}(\mathbf{u})} \frac{\partial S(\mathbf{v})}{\partial v_k}\bigg|_{\mathbf{v}=\mathbf{u}(\mathbf{w})} \nabla u_k(\mathbf{w}) \tag{4}$$

where real vector-valued function $\mathbf{u}(\mathbf{w})$ is intermediate between complex vector argument $\mathbf{w}$ and real scalar $S(\mathbf{u}(\mathbf{w}))$.

The needed gradients will follow from specific choices of complex matrix parameter $\mathbf{A}$ in

$$S(\mathbf{v}) \triangleq \frac{1}{\text{len}(\mathbf{v})} \sum_{\ell=1}^{\text{len}(\mathbf{v})} (v_\ell - 1)^2 \tag{5}$$

$$u_k(\mathbf{w}) = \big[\mathbf{u}(\mathbf{w})\big]_k \triangleq \big|\big[\mathbf{A}\mathbf{w}\big]_k\big|^2. \tag{6}$$

Certainly

$$\frac{\partial S(\mathbf{v})}{\partial v_k}\bigg|_{\mathbf{v}=\mathbf{u}(\mathbf{w})} = \frac{2(u_k(\mathbf{w}) - 1)}{\text{len}(\mathbf{u}(\mathbf{w}))}. \tag{7}$$

Further, it's straightforward if perhaps tedious to show that $\nabla \mathbf{w}^H \mathbf{Q} \mathbf{w} = 2\mathbf{Q}\mathbf{w}$ for any choice of Hermitian matrix $\mathbf{Q}$. Using this fact and letting $\mathbf{a}_1, \ldots, \mathbf{a}_K$ denote the rows of $\mathbf{A}$,

$$\nabla u_k(\mathbf{w}) = \nabla \mathbf{w}^H \mathbf{a}_k^H \mathbf{a}_k \mathbf{w} = 2\mathbf{a}_k^H \mathbf{a}_k \mathbf{w}.$$

Substituting this and (7) into chain rule (4) yields

$$\nabla S(\mathbf{u}(\mathbf{w})) = \frac{4}{\text{len}(\mathbf{A}\mathbf{w})} \sum_{k=1}^{\text{len}(\mathbf{A}\mathbf{w})} \big(\big|\big[\mathbf{A}\mathbf{w}\big]_k\big|^2 - 1\big) \mathbf{a}_k^H \mathbf{a}_k \mathbf{w}$$

or, using $\text{dg}(\,)$ for the "diagonal" function that creates a square diagonal matrix from its vector argument,

$$\nabla S(\mathbf{u}(\mathbf{w})) = \frac{4}{\text{len}(\mathbf{A}\mathbf{w})} \mathbf{A}^H \big(\text{dg}(\mathbf{A}\mathbf{w})\, \text{dg}(\mathbf{A}\mathbf{w})^* - \mathbf{I}\big)\mathbf{A}\mathbf{w}. \tag{8}$$

5) A Weight-Amplitude Objective Term: Day's approach minimizes the sample variance of complex weight set $\{w_1, \ldots, w_N\}$, as shown on the upper left in Fig. 4. Sample squared magnitudes $|w_n|^2$ are pulled towards mean quantity $\text{avg}\{|w_n|^2\}$, the squared radius of the red circle in the middle of the lighter cloud that represents the locations of the weights themselves. Using the sample mean of $|w_n|^2$ as a reference allows the overall scaling of the weights to float, necessary because the overall scaling is set to fix array factor sample $A(\mathbf{k}_0)$, where $\mathbf{k}_0$ represents the desired beam direction.

Here, however, we do not fix the scaling in beamspace and must set it elsewhere. To do this, we replace reference mean $\text{avg}\{|w_n|^2\}$ with unity to obtain objective term

$$F(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} (|w_n|^2 - 1)^2. \tag{9}$$

(Also see the lower left in Fig. 4.) Objective term $F(\mathbf{w}) = S(\mathbf{u}(\mathbf{w}))$ with $d = 1$ and $\mathbf{H} = \mathbf{I}$, so (8) yields

$$\nabla F(\mathbf{w}) = \frac{4}{\text{len}(\mathbf{w})} \big(\text{dg}(\mathbf{w})\, \text{dg}(\mathbf{w})^* - \mathbf{I}\big)\mathbf{w} \tag{10}$$

or the equivalent, since $\text{len}(\mathbf{w}) = N$,

$$\big[\nabla F(\mathbf{w})\big]_n = \frac{4}{N} w_n (|w_n|^2 - 1).$$

6) A Passband-Amplitude Objective Term: Given complex weights w_n associated to element positions $\mathbf{B}\boldsymbol{m}_n$ for $n = 1,\ldots,N$, a column vector containing array-factor samples at spatial-frequency row vectors $\mathbf{k} = \mathbf{k}_1,\ldots,\mathbf{k}_K$ is given by

$$\begin{bmatrix} A(\mathbf{k}_1) \\ \vdots \\ A(\mathbf{k}_K) \end{bmatrix} \triangleq \mathbf{H}\mathbf{w}, \quad \mathbf{H} \triangleq \begin{bmatrix} \mathrm{e}^{-j2\pi\mathbf{k}_1\mathbf{B}\boldsymbol{m}_1} & \cdots & \mathrm{e}^{-j2\pi\mathbf{k}_1\mathbf{B}\boldsymbol{m}_N} \\ \vdots & \ddots & \vdots \\ \mathrm{e}^{-j2\pi\mathbf{k}_K\mathbf{B}\boldsymbol{m}_1} & \cdots & \mathrm{e}^{-j2\pi\mathbf{k}_K\mathbf{B}\boldsymbol{m}_N} \end{bmatrix}.$$

The passband-derived objective term is then given by

$$G(\mathbf{w}) \triangleq \frac{1}{K}\sum_{k=1}^{K}(|A(\mathbf{k}_k)/d|^2 - 1)^2$$

for some real constant target amplitude $d > 0$. Comparing to (5) and (6), objective term $G(\mathbf{w}) = S(\mathbf{u}(\mathbf{w}))$ with $\mathbf{A} = d^{-1}\mathbf{H}$ and so $\mathrm{len}(\mathbf{A}\mathbf{w}) = K$. It then follows from (8) that

$$\nabla G(\mathbf{w}) = \frac{4}{d^4 K}\,\mathbf{H}^H\big(\mathrm{dg}(\mathbf{H}\mathbf{w})\,\mathrm{dg}(\mathbf{H}\mathbf{w})^* - d^2\mathbf{I}\big)\mathbf{H}\mathbf{w}.$$

This can be computed as written or using (10) as

$$\nabla G(\mathbf{w}) = d^{-1}\mathbf{H}^H\,\nabla F(d^{-1}\mathbf{H}\mathbf{w}).$$

7) Hidden Offsets Spread Points in Complex Angle: The array factors in Fig. 1 offset the beam from the intended boresight aim point to spread the complex weights in angle around the circle, purely for display. Here we again introduce such a temporary offset, both for weight display and to improve algorithm behavior. The offset is removed in the final result, after algorithm convergence. The size of the offset is set so that the phase factor introduced into the weights covers the entire unit circle when all elements are considered.

Similarly, a temporary offset is introduced in position and removed after convergence. This centers the optimized array away from the origin and spreads complex passband samples around the circle in complex angle, the latter both to aid display of those samples and to benefit algorithm behavior. Because this offset is chosen from the element-center lattice, array-factor periodicity is not disturbed. The size of the offset is set so that the phase factor introduced into the array factor covers the entire unit circle (and typically a touch more) as the desired passband is sampled.

8) Mixing the Two Objective Terms: The nominal objective used in the gradient descent is $\gamma F(\mathbf{w}) + (1-\gamma)G(\mathbf{w})$, where $0 \le \gamma < 1$. Tuning parameter γ balances the "forces" that lower the two objective terms at each gradient-descent iteration.

C. Preserving the Sector Null

The nominal gradient above is modified to create the actual gradient used in steepest-descent iteration. That modification removes any gradient component that would change the null sector by applying the technique of Section A2. In subspace-removal operation (2), the nominal gradient vector simply replaces $\mathbf{w}$. The same subspace-definition matrix $\mathbf{V}$ is used.

TABLE I
THE PARAMETER SCHEDULE USED IN THE DESIGN EXAMPLE.

stage	*purpose*	*iterations*	γ	*relaxation*
0	initialization of weights			
1	scale passband	1	0%	5 dB
2	start work on weights	3	10%	5 dB
3	work more on weights	3	5%	5 dB
4	clean up passband	10	3%	5 dB
5	clear center	20	20%	5 dB
6	rotate passband	55	20%	4 dB
7	pull up passband	50	20%	3 dB
8	pull up again	30	20%	2 dB
9	pull up yet again	25	20%	1 dB
10	clean up weights	5	50%	1 dB
11	pull less to clean up	5	50%	1.25 dB
12	final tweak	2	90%	1.5 dB
13	hard limiting of weights			

III. A Single Example Optimization

Figs. 5 through 18 present the state of an example optimization at the end of each stage in Table I. The captions describe each stage and the hand-tuned parameters from the table.

On the left in each figure is a high-level view of the array factor using the color scale shown, which has 10 dB ticks and marked 0 dB and peak array-factor levels. A hexagon and a globe circumscribe an array-factor period and the visible region respectively, and the latter has azimuth and elevation lines at 15° intervals. In Fig. 5 the 857 pre-SVD sector-null constraint points are marked faintly—the broad smile. The sector nulling of Section A2 kept 109 singular values.

On the right in each figure is a zoomed-in view of the passband region with boundary as marked. Each figure's color scale is different but marked. In Fig. 5 the 1483 passband-constraint locations appear as a faint triangular grid.

Each figure's center shows two complex planes. On the bottom are complex weights and a unit-circle reference. On top are complex passband samples and a reference circle with the radius that gives an ideal passband (brick-wall sides, flat top) the entire targeted weight energy of N. From Fig. 6 on, a smaller circle of radius d marks the amplitude at which the passband-samples objective term is zero.

A matlab bug precluded larger fonts in the images (they were larger onscreen, but...), so the numbers are hard to read. However, those numbers are repeated in the captions and in Table I with the exception of the dB numbers on the color scales.

IV. Conclusion

Iterative creation of phase-only weights for broad beams in large planar arrays was presented. A quartic objective function keeps computation simple and quick, but its few degrees of freedom prohibit tuning to forcefully keep complex weights and passband samples away from the origin, which becomes a sort of metastable trap slowing convergence. At this preliminary stage, we dealt with those origin traps using hand-tuned parameter scheduling, but clearly this should be replaced with some sort of adaptive parameter tuning. Other metastable traps need work as well. Randomization of the Section B7 offsets may help.

REFERENCES

[1] D. Day, "Robust phase-only nulling for adaptive and non-adaptive phased arrays," in *IEEE Asilomar Conf. on Signals, Systems and Computers*, Nov. 2007, pp. 2173–2176.

[2] D. A. Day, "Fast phase-only pattern nulling for large phased array antennas," in *IEEE Radar Conf.*, May 2009, pp. 1–4.

[3] E. Stein and G. Weiss, *Introduction to Fourier Analysis on Euclidean Spaces*. Princeton University Press, 1971.

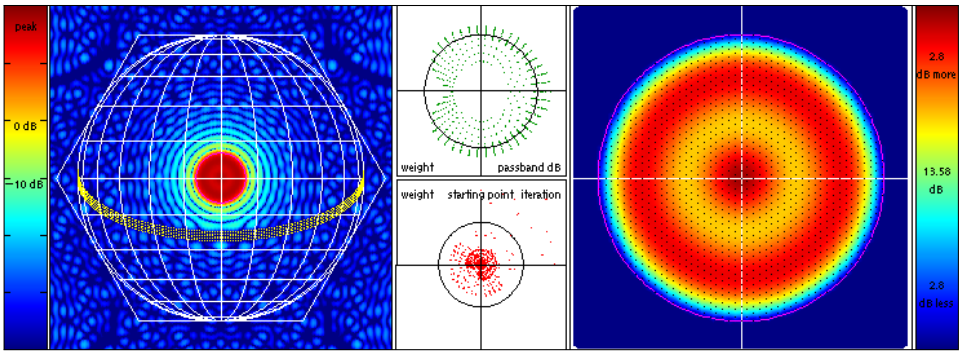


Fig. 5. Stage 0 of the example optimization initializes the weights for a radius-10λ circular aperture including $N = 1075$ elements on a equilateral-triangular grid spaced at $\lambda/\sqrt{3}$. Boresight is $25°$ above the horizon. Weights are obtained by (Section A) element-center sampling of an Airy distribution parameterized for a boresight beam $20°$ wide and scaled to a power gain of N. A sector null extends from $1°$ below the horizon to $3°$ above it.

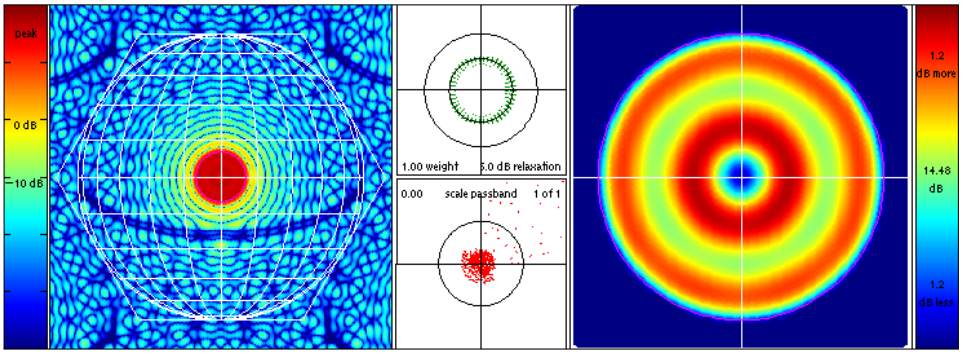


Fig. 6. Stage 1 of the example optimization scales the passband and compresses its magnitude fluctuations, in a single iteration, to move the passband sample points to a safer distance from the origin, where they could so easily be trapped. Here the weight term in the objective is simply dropped, and the passband amplitude target is relaxed 5 dB from what would put the entire energy of N unit-amplitude elements into an ideal passband of the design radius.

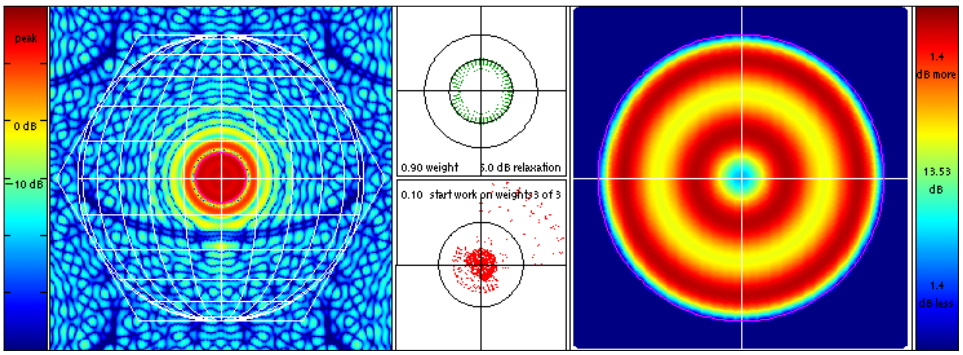


Fig. 7. Stage 2 of the example optimization uses three iterations to begin to pull the weights towards the unit circle. Objective terms governing weights and the passband are weighted at 10% and 90%, with the latter large to keep passband samples from slipping towards the origin.

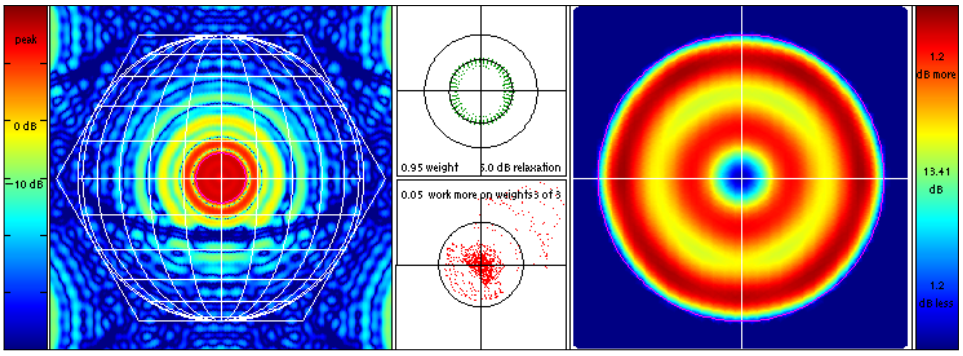


Fig. 8. Stage 3 of the example optimization continues with three more iterations, now with weight and passband objective terms weighted 5% and 95%. The more extreme term weighting is needed because here some passband samples are initially in a more precarious position nearer the origin.

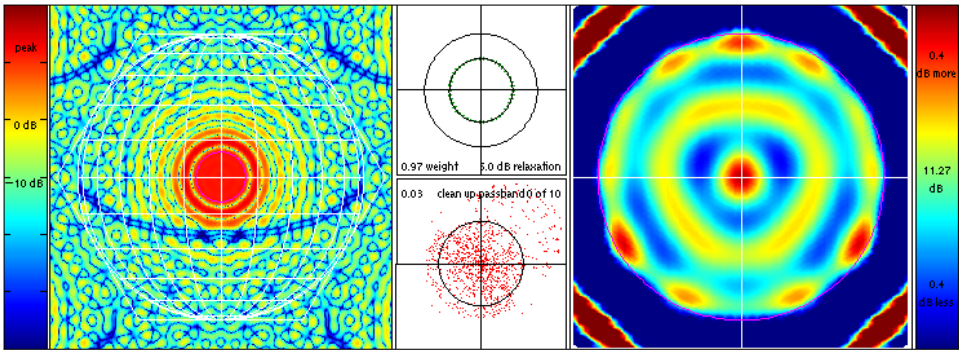


Fig. 9. Stage 4 of the example optimization uses 10 iterations to really shrink the passband variation, possible now because of a safer—away from the origin—starting point. Here objective weighting tilts 97% towards the passband.

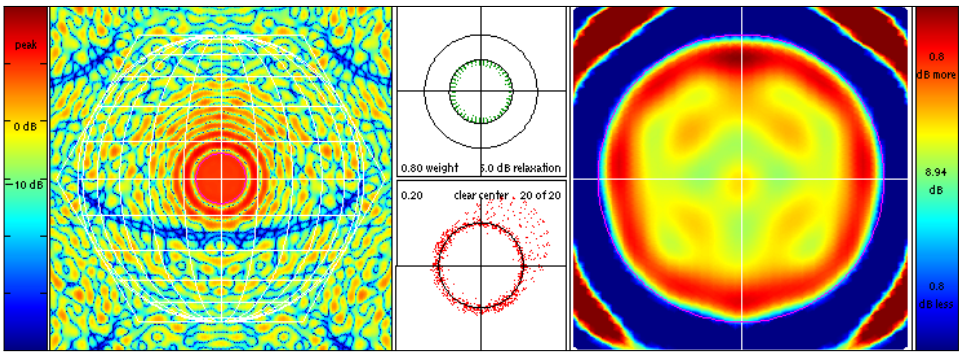


Fig. 10. Stage 5 of the example optimization dedicates 20 iterations to clearing the center of the weight scatterplot by giving the weight amplitudes a 20% weight in the objective.

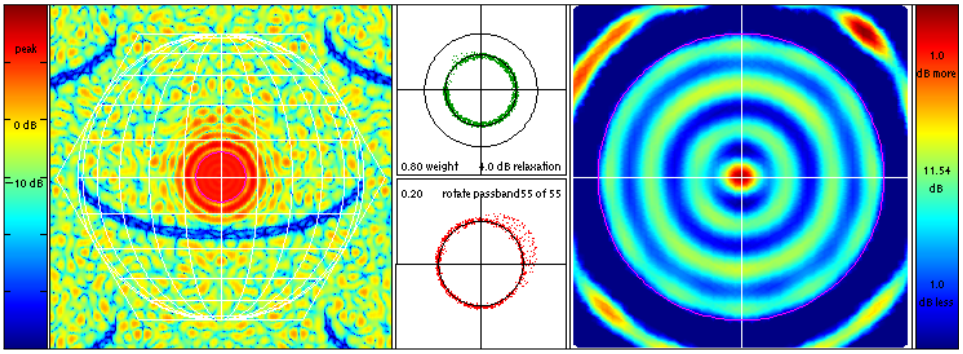


Fig. 11. Stage 6 of the example optimization runs a full 55 iterations to allow some peculiar phenomology to play out: both passband samples and weights gradually rotate in the complex plane, some more than others, until they are better distributed in angle. This rotation appears to allow the peak weight amplitudes to shrink more easily. Here the passband amplitude target is also raised by 1 dB to begin to shrink sidelobes by "pulling up" on the main beam while total energy is more or less fixed by the unit-amplitude goal of weight iteration.

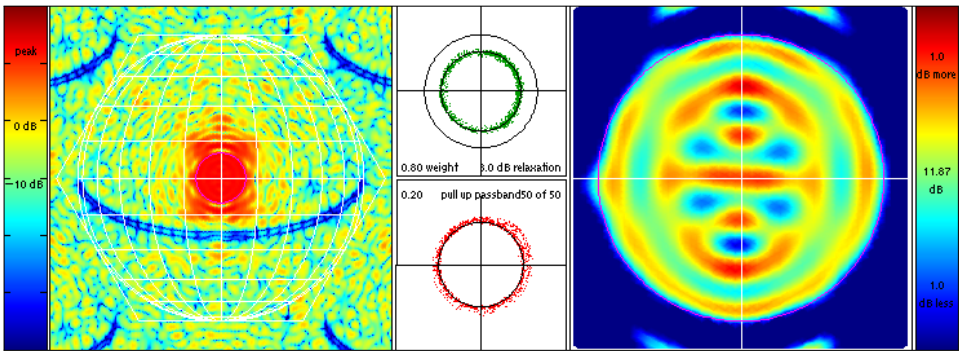


Fig. 12. Stage 7 of the example optimization runs 50 more iterations with no change other than yet another 1 dB increase in the passband amplitude goal. This more aggressive "pulling up" on the main beam begins to introduce significant asymmetry in the inner sidelobes, a phenomenon we are at a loss to explain. The peak weight amplitude has decreased significantly, presumably due to the many iterations now run with a nontrivial weight given to the relevant objective term.

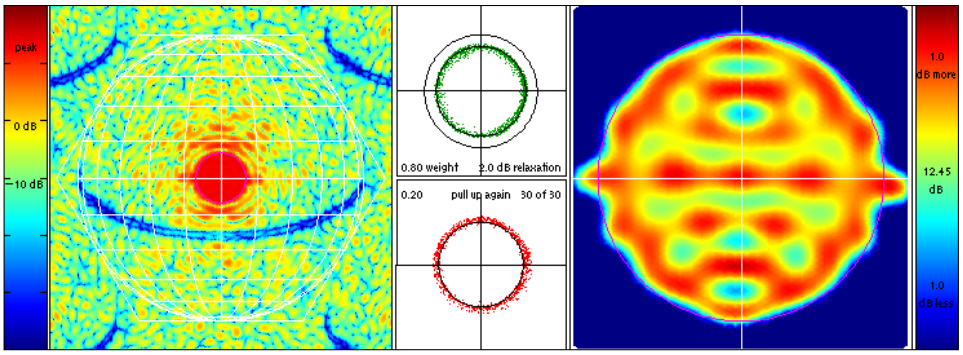


Fig. 13. Stage 8 of the example optimization runs 30 iterations with yet another 1 dB increase in the passband-amplitude goal. This is done simply to improve sidelobe performance. This stage ends with weight amplitudes a little more uniform in amplitude as well.

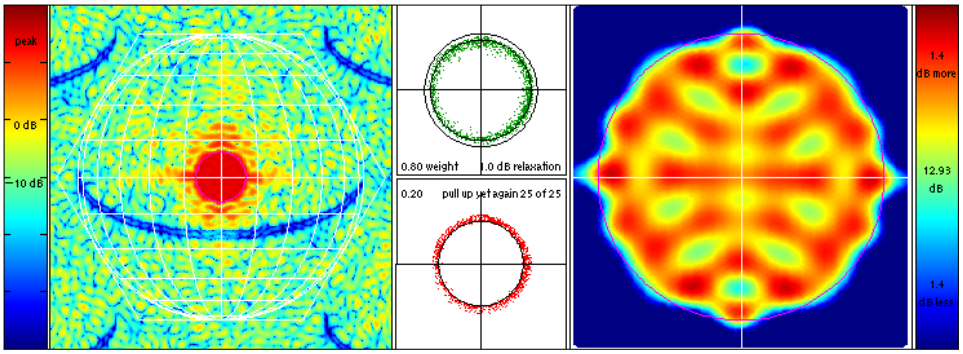


Fig. 14. Stage 9 of the example optimization uses 25 iterations to give a last firm tug upward on the main beam with one final 1 dB increase in the goal, leaving the latter now only 1 dB below what would put all the energy in an ideal "brick wall" beam of the same radius. While some benefit to sidelobe levels is visible, the greater variation in passband amplitude suggests we may have gone just a bit too far.

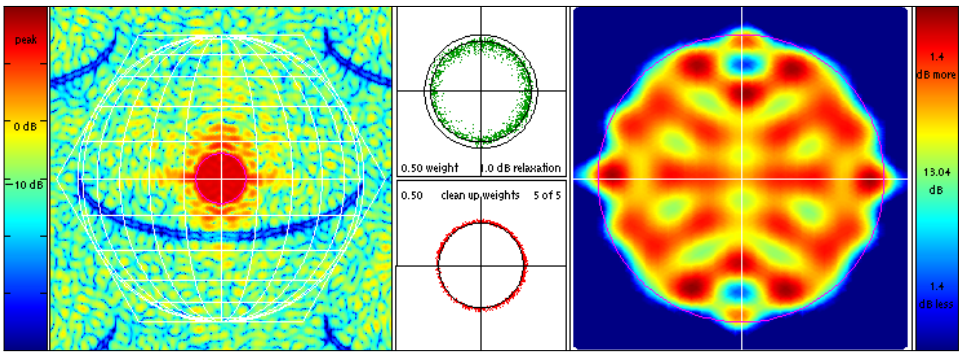


Fig. 15. Stage 10 of the example optimization runs five iterations with the two objective terms given equal weight. This is the most that weight-amplitude variation has been targeted in the objective, and the intent here is to try to aggressively move the weights towards the desired phase-only status without damaging passband performance.

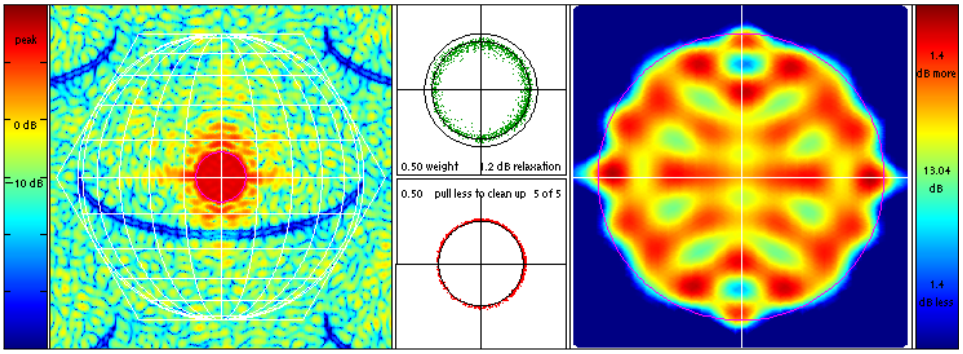


Fig. 16. Stage 11 of the example optimization runs five more iterations with the passband amplitude goal decreased by 0.25 dB in hopes that the amplitude variation within the passband might be reduced a little without hurting other aspects of performance.

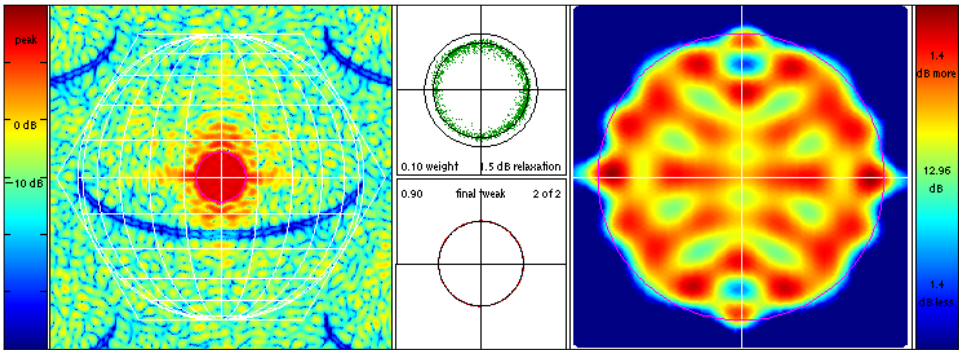


Fig. 17. Stage 12 of the example optimization comprises just two iterations. The goal for passband amplitudes is decreased by another 0.25 dB, now to 1.5 dB below the all-energy-in-the-passband level, this just to ease up on the optimization a bit to allow for what we are really after here: much lower variation in weight amplitudes. The objective term governing the latter is weighted at 90%.

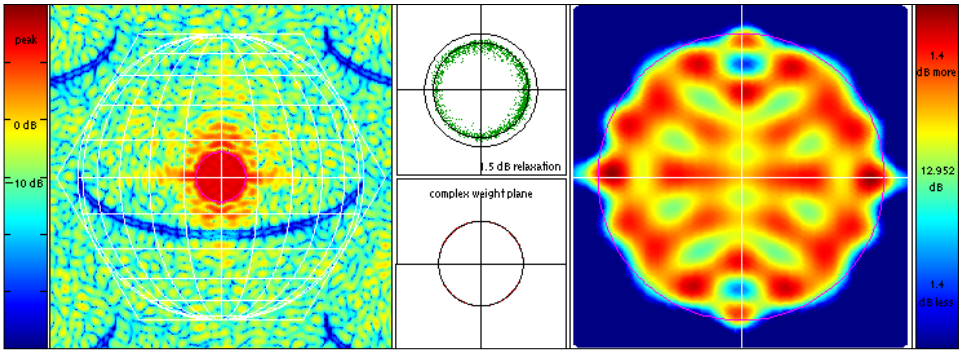


Fig. 18. Stage 13 of the example optimization does not iterate at all but simply hard limits the weight amplitudes to unity. At this point this is a very small change, so the array factor does not undergo marked changes.